BEYOND A SINGLE THOUGHT: LEARNING DIVERSE LATENT REASONING PATHS WITH SOFTPATHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Chain-of-Thought (CoT) reasoning improves the problem-solving capabilities of large language models by generating intermediate reasoning steps, but explicitly verbalizing these steps incurs substantial inference costs and often produces redundant content. Latent CoT alleviates this limitation by encoding intermediate reasoning in continuous representations. However, existing methods often rely on a single latent condition or generate multiple variants through simple resampling or perturbation, which may provide limited exploration of the solution space and fail to capture functionally distinct reasoning directions. We propose **SoftPaths**, a two-stage framework that learns multiple structured latent reasoning paths and adapts the generation policy to reason effectively under these paths. In the first stage, path-specific routers compose a shared set of projection experts to construct multiple input-dependent latent conditions. Answer-only supervision ensures that each path remains predictive of the final answer, while a contrastive diversity objective mitigates representation collapse across paths. In the second stage, the learned latent-path constructor is frozen, and the paths are activated in a round-robin manner to provide diverse latent conditions for GRPO-based policy optimization. A correctness-gated length reward prioritizes answer correctness while discouraging unnecessary verbalization among correct responses. At inference time, SoftPaths supports both efficient single-path reasoning and multi-path test-time scaling through answer aggregation. Experiments across multiple reasoning benchmarks demonstrate that SoftPaths consistently improves the balance between answer accuracy and efficiency over explicit and latent CoT methods.

1 INTRODUCTION

Chain-of-thought reasoning Wei et al. (2022); Kojima et al. (2022) improves the ability of large language models to solve complex problems by generating intermediate reasoning steps before producing a final answer. Recent reasoning models Guo et al. (2025); Team et al. (2025) further extend this paradigm through prolonged analysis, verification, and self-correction. Despite their effectiveness, conventional CoT methods perform intermediate reasoning through discrete natural-language tokens, requiring every reasoning step to be autoregressively verbalized. This process constrains intermediate computation to the vocabulary space and often produces long traces containing repeated calculations, unnecessary verification, or content that does not contribute to the final answer Ma et al. (2025); Qiao et al. (2025); Dumitru et al. (2025). Methods based on self-consistency Wang et al. (2022), structured search Yao et al. (2023), and repeated sampling Brown et al. (2024) can further multiply this cost by generating multiple complete rea-

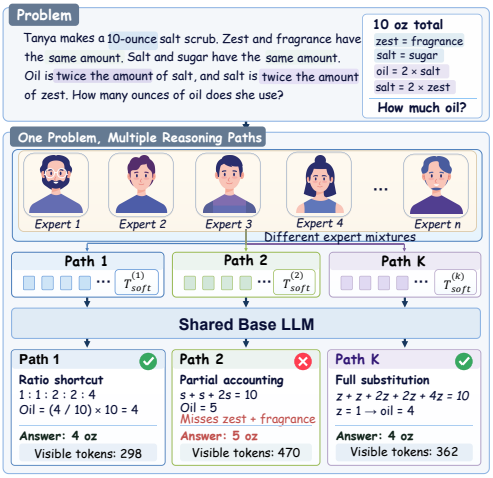


Figure 1: Illustration of how the same question can be answered from different perspectives.

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

soning trajectories. These limitations have motivated increasing interest in latent CoT reasoning, which performs part of the intermediate computation in continuous representation space rather than explicitly expressing every reasoning operation in language.

Recent studies have demonstrated the feasibility of latent CoT reasoning. Instead of treating natural-language rationales as the only form of intermediate computation, it represents reasoning states using dense continuous vectors that can encode task-relevant information compactly. Compressed CoT Cheng & Van Durme (2024) and Coconut Hao et al. (2024) demonstrate that continuous representations can replace or compress parts of explicit reasoning trajectories while preserving useful problem-solving information. SoftCoT Xu et al. (2025a) uses a frozen assistant model to produce contextual hidden states and trains an external projector to map them into the embedding space of a frozen base model. More recently, Self-SoftCoT Dong et al. (2026) introduces sequence-level reinforcement learning to optimize latent projection parameters while keeping the backbone decoder frozen. Despite this progress, existing latent CoT methods often either construct a single latent condition for each input or obtain multiple variants through simple resampling or perturbation. A single latent condition restricts the model to one implicit reasoning direction, while resampled variants may differ in representation space without inducing meaningfully different reasoning behaviors. Moreover, when multiple paths are trained under the same answer supervision, they may gradually converge to similar latent transformations and provide redundant guidance. This limitation is particularly important for complex problems, which often admit multiple valid solution strategies that differ substantially in their intermediate computations, reasoning difficulty, and required number of steps. Functionally diverse latent paths can broaden coverage of the solution space and increase the likelihood of discovering a more direct and reliable solution.

To address these limitations, we propose **SoftPaths**, a two-stage latent CoT framework that combines diverse latent-path learning with latent-conditioned generation-policy optimization. In the first stage, a frozen assistant model extracts contextual representations from the input, and a trainable projector transforms them into multiple latent reasoning paths. The projector contains a shared pool of projection experts and multiple path-specific routers, allowing different paths to construct distinct expert compositions from the same input representation. All paths are jointly trained with answer-only supervision to preserve task-relevant information, together with a contrastive diversity objective that discourages representational collapse. In the second stage, the latent-path constructor is frozen, and a LoRA policy is optimized using Group Relative Policy Optimization (GRPO). At each optimization step, the latent path is activated in a round-robin manner and used to condition a group of sampled responses. This exposes the policy to different latent reasoning conditions while keeping the learned paths fixed throughout reinforcement learning. The sampled responses are scored using a correctness-gated length reward that prioritizes answer correctness and favors shorter generations only among correct responses, preventing the policy from exploiting the length objective through incorrect but concise outputs. At inference time, SoftPaths supports both efficient single-path decoding and multi-path test-time scaling through answer aggregation. Experiments across multiple benchmarks demonstrate that SoftPaths achieves a stronger balance between answer accuracy and efficiency than competitive explicit and latent CoT methods.

2 Related Work

Explicit Thinking. Explicit Chain-of-Thought (CoT) reasoning improves complex problem solving by verbalizing intermediate computations before producing a final answer (Wei et al., 2022; Kojima et al., 2022). Subsequent approaches enhance reasoning through repeated sampling, structured search, process supervision, and reinforcement learning (Wang et al., 2022; Yao et al., 2023; Lightman et al., 2024; Shao et al., 2024). Recent reasoning models further scale test-time computation through longer analysis, reflection, and verification (Guo et al., 2025; Team et al., 2025). Although effective, these approaches still rely heavily on autoregressive generation of intermediate reasoning tokens, leading to substantial inference overhead. Recent studies therefore explore more efficient explicit reasoning through adaptive reasoning budgets, chain-of-thought compression, and more concise reasoning trajectories (Shen et al., 2025a; Ma et al., 2025; Qiao et al., 2025). These limitations motivate an alternative line of research that moves intermediate computation from discrete language tokens into continuous latent representations.

Latent Thinking. Latent reasoning performs intermediate computation in continuous representations rather than explicitly verbalizing every reasoning step, reducing the dependence on lengthy autoregressive trajectories. Early studies showed that hidden states can encode unverbalized intermediate information and explored distillation, progressive internalization, and additional latent computation (Deng et al., 2023; 2024; Goyal et al., 2024). Coconut (Hao et al., 2024) introduced Continuous Chain-of-Thought reasoning by recurrently feeding hidden states back into the model, while subsequent approaches compress or learn reasoning information through compact continuous representations or soft tokens (Cheng & Van Durme, 2024; Shen et al., 2025b; Xu et al., 2025a; He et al., 2026). More recent work further investigates training-free latent reasoning and test-time scaling in continuous space (Zhang et al., 2026; Xu et al., 2025b). Existing methods primarily focus on constructing, compressing, or scaling latent reasoning representations. In contrast, SoftPaths explicitly learns multiple structured latent alternatives through path-specific routing over shared experts, and further adapts a shared generation policy to effectively exploit these diverse latent paths.

3 METHOD

3.1 PROBLEM FORMULATION AND METHOD OVERVIEW

Let $x = [I; Q]$ denote the input consisting of a task instruction I and a question Q, and let A^* be the ground-truth answer. In explicit CoT reasoning, the model generates $Y^{\text{exp}} = [R_{\text{exp}}; A]$, where R_{exp} is the explicit reasoning trace and A is the predicted answer, both decoded autoregressively as visible tokens. Latent CoT instead introduces a continuous, input-dependent representation that conditions the model before visible response generation. Formally, a latent-path constructor maps the input x to a high-dimensional latent representation T_{soft}. The base model then generates a visible response conditioned on both the input and the latent representation:

$$p_\theta(Y \mid x, T_{\text{soft}}) = \prod_{t=1}^{|Y|} p_\theta\big(y_t \mid x, T_{\text{soft}}, y_{<t}\big). \tag{1}$$

Here, $Y = [R_{\text{vis}}; A]$, where R_{vis} denotes the visible reasoning content generated before the final answer. Unlike explicit CoT, part of the intermediate computation is carried by T_{soft} in continuous representation space rather than being verbalized as visible tokens.

The overview of SoftPaths consists of two stages. In Stage I, the shared projection experts and path-specific routers are learned while both the assistant and base models remain frozen. Each latent path is trained with answer supervision together with a contrastive objective that discourages representational collapse across paths. In Stage II, the assistant model and SoftPaths projector are frozen, while a LoRA adapter attached to the base model is optimized using groups of stochastic rollouts conditioned on one active latent path at a time. At inference, SoftPaths supports both single-path and multi-path decoding: a single latent path produces one answer, whereas multiple paths produce one answer each and are aggregated by majority voting.

3.2 LATENT PATH CONSTRUCTION

Assistant Representation Extraction. Given the task input $x = [I; Q]$, we form the assistant input $X_{\text{assist}} = [I_{\text{assist}}; Q; \texttt{<cot>}^m]$ where the original task instruction I is replaced with an assistant-specific instruction I_{assist}, and $\texttt{<cot>}^m$ denotes m latent placeholder positions. A frozen assistant model f_{assist} encodes X_{assist} and produces contextual hidden states $\mathbf{H}_{\text{assist}} = f_{assist}(X_{\text{assist}})$. We extract the hidden states at the placeholder positions as $\mathbf{H}_{\text{cot}} = \mathbf{H}_{\text{assist}}[\mathcal{P}_{\text{cot}}] \in \mathbb{R}^{m \times d_a}$, where $\mathcal{P}_{\text{cot}}$ denotes the corresponding position indices and d_a is the hidden dimension of the assistant model. The extracted representation provides the contextual features used to construct multiple latent reasoning paths.

Multi-Path MoE Projection and Injection. The representation space of the assistant model generally differs from the embedding space of the base model. To bridge this gap, we introduce an MoE projector consisting of E shared projection experts $\{\mathcal{E}_e\}_{e=1}^{E}$ and K path-specific routers $\{\mathbf{U}_k\}_{k=1}^{K}$. Each expert maps the extracted assistant representations from dimension d_a to the base-model embedding dimension d, while each router independently determines how the shared experts are composed for its corresponding latent path. For the k-th path, the router produces a sparse and normal-

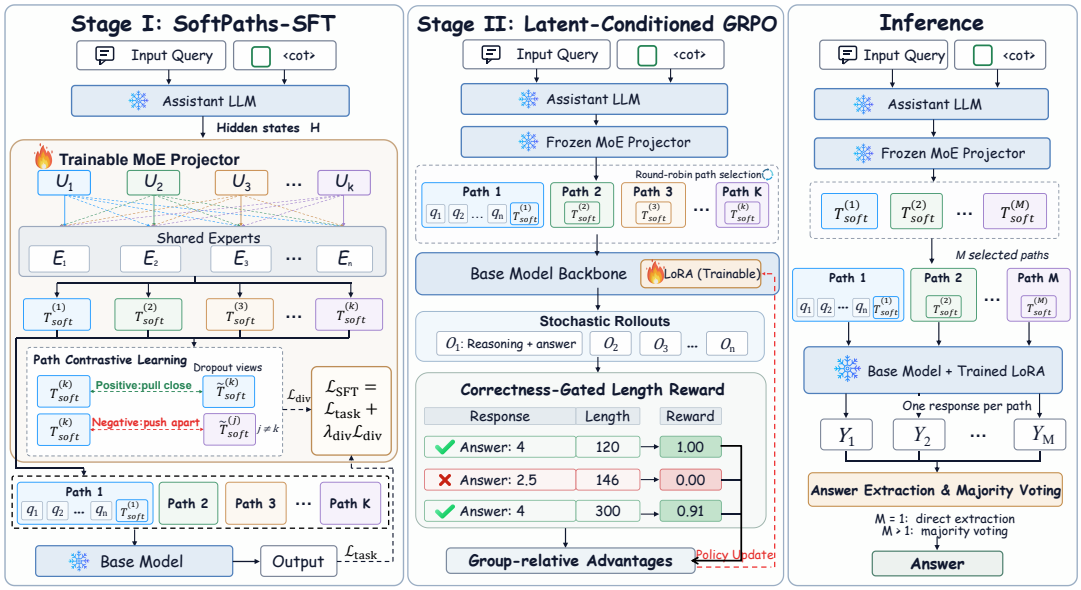


Figure 2: Overview of SoftPaths. Stage I learns diverse latent reasoning paths through answer supervision and path diversity regularization. Stage II freezes the learned paths and optimizes a fresh LoRA policy with GRPO under round-robin path conditioning. At inference, SoftPaths supports both single-path decoding and multi-path answer aggregation.

ized expert-weight vector

$$\boldsymbol{\pi}_k = \mathbf{U}_k(\mathbf{H}_{\text{cot}}) = [\pi_{k,1}, \ldots, \pi_{k,E}] \in \mathbb{R}^E, \qquad \sum_{e=1}^{E} \pi_{k,e} = 1, \tag{2}$$

where only a subset of experts receives non-zero routing weights. Here, $\mathbf{U}_k$ denotes the complete path-level routing function, including aggregation over the placeholder representations and sparse normalization. The corresponding latent thought path is constructed as

$$T_{\text{soft}}^{(k)} = \sum_{e=1}^{E} \pi_{k,e} \mathcal{E}_e\left(\mathbf{H}_{\text{cot}}\right) \in \mathbb{R}^{m \times d}, \qquad k = 1, \ldots, K. \tag{3}$$

Although all paths share the same projection experts, their routers have independent parameters and can therefore learn different expert-composition functions. This design transforms a common assistant representation into multiple input-dependent latent conditions, allowing different paths to capture alternative reasoning perspectives for the same question.

For each path k, we first map the task input x into the embedding space of the base model and then concatenate its embeddings with the projected latent sequence:

$$\mathbf{X}_b^{(k)} = \text{Emb}_b(x) \oplus T_{\text{soft}}^{(k)}, \tag{4}$$

where $\oplus$ denotes concatenation along the sequence dimension, and $\text{Emb}_b(x)$ denotes the base-model embeddings of x. The resulting embedding sequence $\mathbf{X}_b^{(k)}$ jointly preserves the textual information in x and introduces the latent condition associated with path k. The latent vectors are directly processed by the base model as continuous input embeddings and are never decoded into vocabulary tokens.

3.3 Stage I: Learning Diverse Latent Paths

Stage I aims to learn multiple latent paths that are individually informative for answer prediction while remaining distinct from one another. We first perform a short task warm-up in which only a temporary LoRA adapter attached to the base model is optimized on the full response $Y^{\text{full}} = [R; A]$,

while the assistant model, base model, and SoftPaths projector remain frozen. After warm-up, the temporary adapter is frozen and only the SoftPaths projector is optimized to construct answer-relevant latent paths for each input question.

All-Path Answer Supervision. For each training example $(I, Q, R, A) \in \mathcal{D}_{\mathrm{SFT}}$, projector optimization uses only the final answer A as the generation target, while the textual rationale R is excluded. Given the K latent-conditioned inputs $\{\mathbf{X}_b^{(k)}\}_{k=1}^{K}$ defined in Eq. 4, the answer-supervision objective is

$$\mathcal{L}_{\mathrm{task}} = \frac{1}{K} \sum_{k=1}^{K} \ell_{\mathrm{CE}} \left(A \mid \mathbf{X}_b^{(k)} \right), \tag{5}$$

where ℓ_{CE} denotes the token-averaged autoregressive cross-entropy loss. This all-path supervision encourages each latent condition to preserve the information required for final-answer prediction without explicitly reproducing the textual rationale.

Path Diversity Regularization. Answer supervision alone does not prevent different paths from converging to similar latent representations. We therefore introduce a contrastive diversity objective that encourages consistency within each path while separating different paths generated for the same input. For each path, its original representation and a dropout-perturbed view form a positive pair, while views from the other paths of the same input serve as negatives. Let $s_{k,j} = \mathrm{sim}(T_{\mathrm{soft}}^{(k)}, \widetilde{T}_{\mathrm{soft}}^{(j)})/\tau_c$, where $\mathrm{sim}(\cdot,\cdot)$ denotes cosine similarity and τ_c is the contrastive temperature. The diversity objective is

$$\mathcal{L}_{\mathrm{div}} = -\frac{1}{K} \sum_{k=1}^{K} \log \frac{\exp(s_{k,k})}{\sum_{j=1}^{K} \exp(s_{k,j})}. \tag{6}$$

Together, $\mathcal{L}_{\mathrm{task}}$ ensures that each path remains useful for answer prediction, while $\mathcal{L}_{\mathrm{div}}$ discourages representational collapse across paths. The overall Stage-I objective is

$$\mathcal{L}_{\mathrm{SFT}} = \mathcal{L}_{\mathrm{task}} + \lambda_{\mathrm{div}} \mathcal{L}_{\mathrm{div}}, \tag{7}$$

where λ_{div} controls the strength of the diversity regularization. Additional details on contrastive-pair construction are provided in Appendix A.1.

3.4 Stage II: Latent-Conditioned Policy Optimization

SFT learns answer-relevant latent paths through teacher-forced token prediction, but it does not directly optimize sequence-level rewards for responses sampled under these latent conditions. Latent-conditioned GRPO addresses this limitation by optimizing a fresh base-model LoRA adapter parameterized by ω, while the assistant model, MoE projector, and pretrained base model remain frozen. This update strategy adapts visible response generation without altering the learned latent paths.

Latent-Conditioned Group Sampling. For each training input x, the frozen assistant model and MoE projector construct K latent paths. At each optimization step, one path $k \in \{1, \ldots, K\}$ is selected, with the paths activated in a fixed round-robin order across consecutive steps. Following Eq. 4, the selected path forms the corresponding latent-conditioned input $\mathbf{X}_b^{(k)}$. Conditioned on $\mathbf{X}_b^{(k)}$, the policy then samples a group of G textual responses. All G responses share the same training input and active latent path, while their variation arises solely from stochastic decoding. Cycling the active path across optimization steps systematically exposes the LoRA policy to all learned latent conditions while keeping each rollout group conditioned on a single path.

Correctness-Gated Length Reward. For the g-th response Y_g, we define the correctness indicator as $c_g = \mathbb{I}[\mathrm{Extract}(Y_g) = A^*]$, where $\mathbb{I}[\cdot]$ is the indicator function, $\mathrm{Extract}(\cdot)$ extracts the predicted answer, and A^* is the ground-truth answer.

Let $f_g \in \{0, 1\}$ denote the task-specific format indicator,where $f_g = 1$ if the response satisfies the required answer format and $f_g = 0$ otherwise. The final reward is defined as

$$r_g = c_g \left[(1 - \lambda_{\mathrm{fmt}}) + \lambda_{\mathrm{fmt}} f_g \right] \exp(-\alpha \delta_g), \tag{8}$$

where λ_{fmt} controls the contribution of format compliance, α controls the strength of the length penalty, and δ_g denotes the normalized excess-length term defined below. Let $\ell_g = |Y_g|$ denote the

number of visible tokens in response Y_g, and let $\mathcal{C} = \{g \mid c_g = 1\}$ denote the indices of the correct responses in the rollout group. When $|\mathcal{C}| \geq 2$, we compute the mean $\mu_{\mathcal{C}}$ and standard deviation $\sigma_{\mathcal{C}}$ of their visible-token lengths. The normalized excess length is defined as

$$\delta_g = \begin{cases} \dfrac{\ell_g - \mu_{\mathcal{C}}}{\sigma_{\mathcal{C}} + \epsilon}, & g \in \mathcal{C},\ |\mathcal{C}| \geq 2,\ \ell_g > \mu_{\mathcal{C}}, \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

where $\epsilon > 0$ ensures numerical stability. Thus, correct responses below the mean correct-response length receive no length penalty, while only longer correct responses are penalized. When fewer than two correct responses are available in the rollout group, no length penalty is applied. Because the entire reward is multiplied by c_g, an incorrect response receives zero reward regardless of its format indicator or response length. Additional details of the GRPO optimization are provided in Appendix A.2.

3.5 Test-Time Scaling with Latent Paths

All model parameters remain fixed during inference. Given an input $x = (I, Q)$, the assistant model and MoE projector construct K latent paths $\{T_{\text{soft}}^{(k)}\}_{k=1}^{K}$. SoftPaths supports two inference modes by varying the number of latent paths used at test time.

Single-Path Inference. For efficient inference, SoftPaths selects one latent path $T_{\text{soft}}^{(k)}$ and concatenates it with the base-model embeddings of x following Eq. 4. The base model then performs a single decoding pass to generate one response $Y^{(k)}$, from which the final answer is directly extracted. This mode introduces only one base-model inference pass and therefore provides the most efficient deployment setting in our framework.

Multi-Path Inference. For test-time scaling, SoftPaths can use multiple latent paths for the same input. Each selected latent path is independently concatenated with the textual input embeddings and processed through a separate base-model decoding pass. Using M latent paths therefore produces M candidate responses, where $1 < M \leq K$. A task-specific extractor converts each response into a normalized answer, and the final prediction is determined by majority voting over all valid answers. Invalid outputs are excluded before the final majority vote is performed.

Single-path inference corresponds to $M = 1$ and requires no answer aggregation, whereas multi-path inference trades additional decoding computation for broader coverage of learned latent reasoning paths. Unlike conventional self-consistency, which repeatedly samples from the same input condition, SoftPaths generates candidate responses under multiple learned latent conditions.

4 Experiments

We evaluate SoftPaths on five benchmarks covering mathematical and commonsense reasoning through comparisons with text-space and latent-space methods. We investigate visible-token efficiency, candidate scaling, and path complementarity, and conduct ablations to assess the contributions of latent-path construction and reward design. Implementation details and evaluation protocols are provided in Appendix B, while qualitative behavior is illustrated in Appendix C.

4.1 Datasets and Baselines

Datasets. We evaluate SoftPaths on five reasoning benchmarks: GSM8K (Cobbe et al., 2021), ASDiv-Aug (Miao et al., 2020; Xu et al., 2025a), MATH (Hendrycks et al., 2021), CommonsenseQA (Talmor et al., 2019), and StrategyQA (Geva et al., 2021). GSM8K, ASDiv-Aug, and MATH evaluate mathematical reasoning at varying levels of difficulty, whereas CommonsenseQA and StrategyQA assess commonsense reasoning and implicit multi-step reasoning, respectively.

Baselines. The text-space baselines include CoT (Wei et al., 2022), Assist-CoT (Xu et al., 2025a;b), CoT-SC (Wang et al., 2022), and Assist-CoT-SC (Xu et al., 2025a;b). CoT generates a single explicit reasoning trajectory, while Assist-CoT conditions generation on an explicit reasoning prefix produced by the assistant model; their SC variants obtain multiple candidates through repeated sampling from the corresponding shared condition. The latent-space baselines include Coconut (Hao

Table 1: Answer accuracy (%) on five reasoning benchmarks. Light blue and light orange rows denote candidate budgets $M = 1$ and $M = 4$, respectively. The best and second-best results within each budget are shown in bold and underlined, respectively. CSQA denotes CommonsenseQA.

	Method	Training	M	GSM8K	ASDiv-Aug	MATH	CSQA	StrategyQA	Avg.
Text-space baselines	CoT	Training-free	1	79.00	88.25	34.40	64.46	66.81	66.58
	Assist-CoT	Training-free	1	80.76	88.63	33.40	66.50	66.55	67.17
	CoT-SC	Training-free	4	84.23	89.21	41.80	70.19	72.49	71.58
	Assist-CoT-SC	Training-free	4	87.64	90.17	43.00	71.91	70.74	72.69
Latent-space baselines	Coconut	SFT	1	76.12	86.80	—	—	—	—
	SoftCoT	SFT	1	81.03	87.19	33.13	65.03	69.04	67.08
	Self-SoftCoT	RL	1	82.91	**92.35**	34.20	**75.18**	69.30	70.79
	SoftCoT-SC	SFT	4	87.49	88.54	42.60	68.47	70.31	71.48
	SoftCoT++	SFT	4	86.96	88.63	35.00	70.11	69.87	70.11
Our methods	SoftPaths-SFT	SFT	1	84.08	86.99	36.00	74.69	72.27	70.81
	SoftPaths	SFT + GRPO	1	**84.31**	89.88	**38.00**	75.02	**73.80**	**72.20**
	SoftPaths-SFT	SFT	4	89.76	91.04	44.40	76.90	73.36	75.09
	SoftPaths	SFT + GRPO	4	**90.67**	**92.00**	**45.00**	**77.81**	**77.07**	**76.51**

et al., 2024), SoftCoT and SoftCoT-SC (Xu et al., 2025a), Self-SoftCoT (Dong et al., 2026), and SoftCoT++ (Xu et al., 2025b). These methods respectively represent recurrent latent reasoning, assistant-guided soft thoughts, reinforcement-learning-based latent reasoning, and multi-condition latent reasoning. We additionally report SoftPaths-SFT to isolate the contribution of latent-conditioned GRPO beyond supervised latent-path learning.

4.2 Main Results

Table 1 reports answer accuracy on five reasoning benchmarks under single-response ($M = 1$) and four-candidate ($M = 4$) inference.

Single-Response Performance. At $M = 1$, SoftPaths achieves the highest average accuracy and outperforms both CoT and SoftCoT across all five benchmarks. These gains across arithmetic and commonsense reasoning show that its benefits extend to individual response generation. Furthermore, SoftPaths consistently improves over SoftPaths-SFT, suggesting that learning useful latent conditions alone does not fully exploit their potential. Since this setting does not involve answer aggregation, the improvements suggest that latent-conditioned GRPO helps the shared decoder use the fixed latent conditions more effectively. Self-SoftCoT remains stronger on ASDiv-Aug and slightly stronger on CommonsenseQA. One possible explanation is that directly optimizing the latent projection favors a specialized single condition, whereas SoftPaths adapts a shared decoder to multiple fixed latent paths. Such specialization may benefit particular tasks, while SoftPaths achieves stronger performance on average across the evaluated benchmarks covering both arithmetic and commonsense reasoning.

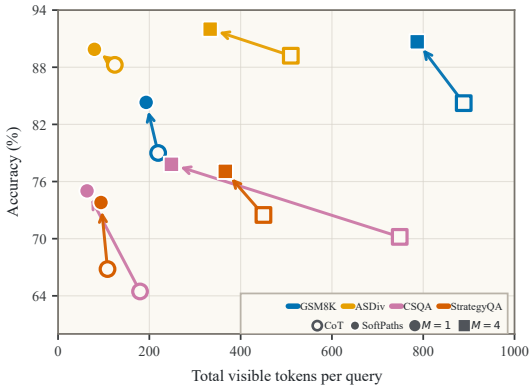


Figure 3: Accuracy–visible-token trade-off per query.Arrows connect each matched CoT baseline to SoftPaths,with upper-left movement indicating higher accuracy and lower visible-token cost.

Multi-Path Performance. First, increasing the candidate budget from $M = 1$ to $M = 4$ generally improves accuracy, showing the benefit of multi-candidate inference. Second, the strong performance of SoftPaths-SFT shows that path-specific routing already produces useful candidate diversity before policy optimization. GRPO further improves performance across all benchmarks, suggesting that it complements latent-path learning by helping the shared decoder better exploit the distinct latent conditions constructed by the projector. The clearest evidence of this complementarity appears on ASDiv-Aug and CommonsenseQA. Third, SoftPaths achieves the strongest average performance

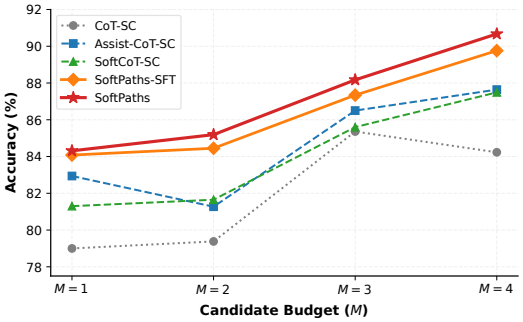


Figure 4: Answer accuracy on GSM8K under matched candidate budgets at inference. Sampling baselines draw M responses from a shared condition, whereas SoftPaths decodes from M learned latent paths per query.

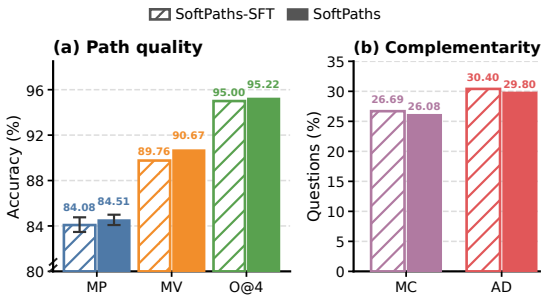


Figure 5: Path quality and outcome complementarity on GSM8K. (a) MP, MV, and O@4 denote mean-path, majority-vote, and Oracle@4 accuracy; (b) MC and AD denote mixed correctness and cross-path answer disagreement.

at $M = 4$, with a larger improvement over competing methods than at $M = 1$. This reversal suggests that the individual paths make different errors and that majority voting can recover correct predictions unavailable from a single selected path. In contrast, conventional self-consistency baselines obtain diversity primarily through repeated stochastic decoding from a shared condition. The stronger and more consistent gains of SoftPaths therefore support learned latent conditions as an effective source of complementary candidates for answer aggregation.

4.3 Efficiency Analysis

Figure 3 shows that SoftPaths consistently achieves higher accuracy with fewer visible tokens than the matched CoT baselines. This improvement is already evident at $M = 1$, indicating that the efficiency gain arises from more effective latent-conditioned generation rather than from multi-path voting. The magnitude of the gain varies across tasks. On ASDiv-Aug, where the baseline is already close to the performance ceiling, SoftPaths primarily reduces visible generation while maintaining strong accuracy. On CommonsenseQA, it provides the largest token reduction together with a substantial accuracy improvement, suggesting that latent conditioning is particularly beneficial when explicit reasoning requires extensive verbalization. MATH is omitted from the shared-axis figure because of its different token scale, but it exhibits the same overall trend. The advantage persists at $M = 4$, where SoftPaths achieves higher accuracy than CoT-SC while generating fewer total tokens across all candidates. This result is notable because the reported cost includes all four responses, showing that the reduction in per-response decoding remains effective under multi-candidate inference. Overall, SoftPaths improves reasoning accuracy without relying on longer visible reasoning traces. These results concern visible language-space decoding and do not imply an equivalent reduction in end-to-end latency or total inference computation. We examine how different sources of candidate diversity affect accuracy and whether the gains of SoftPaths arise from complementary path-conditioned predictions.

Candidate Scaling. Figure 4 shows that repeated sampling from a shared condition does not consistently benefit from a larger candidate budget. A plausible explanation is that stochastic samples from the same condition often preserve correlated reasoning errors. Consequently, additional responses may repeat an incorrect solution rather than introduce useful evidence, and majority voting can reinforce the shared error. By contrast, both SoftPaths variants improve monotonically across the evaluated budgets. Because each candidate is generated from a different learned latent path, increasing M introduces variation in the reasoning condition rather than only in textual decoding. This structured diversity provides a more reliable basis for answer aggregation, allowing SoftPaths to maintain the highest accuracy at every budget and outperform the strongest sampling baseline by more than three percentage points at $M = 4$. The growing advantage of SoftPaths over SoftPaths-SFT further indicates that GRPO contributes more than a uniform improvement in single-path accuracy. During training, the decoder is exposed to the latent paths in turn and learns to interpret their different representations. This adaptation can improve prediction reliability across paths, allowing small path-level gains to yield larger benefits after voting.

Path Complementarity. Figure 5 supports this interpretation. The narrow range of individual-path accuracy suggests that the gains are not driven by a single dominant path; all paths provide useful standalone conditions. At the same time, the improvement from mean-path accuracy to majority voting confirms that the paths make complementary errors. The remaining gap between majority voting and Oracle@4 shows that correct predictions are available for additional questions but are sometimes supported by too few paths to win the vote. Thus, path diversity provides useful information even when simple majority voting cannot fully exploit it. After GRPO, majority-vote accuracy improves while mixed correctness and answer disagreement decrease slightly. This pattern suggests that policy optimization does not improve performance by merely increasing output diversity. Instead, it appears to convert part of the disagreement into correct consensus while preserving enough path-specific variation to recover errors through aggregation.

4.4 Ablation Studies

We conduct ablation studies on GSM8K under a fixed four-candidate decoding budget to examine latent-path construction and reward design. Table 2 shows three main findings. First, replacing the MoE projector with a shared linear projector or removing the path contrastive loss reduces accuracy and increases visible-token usage. The w/o MoE projector variant replaces the MoE projector with a shared linear projector. Notably, retaining MoE routing without the contrastive loss yields lower accuracy than the shared linear variant, suggesting that routing alone does not ensure useful path diversity. Under identical answer supervision, different paths may learn similar representations, limiting their complementarity during answer aggregation. These results support combining path-specific routing with explicit diversity regularization. Second, correctness optimization alone does not ensure efficient generation. Removing both the format and length rewards reduces SoftPaths to correctness-only GRPO, which improves accuracy over the w/o GRPO variant but increases visible-token usage. Retaining the format reward reduces token usage, although with a slight decrease in accuracy. Third, removing length regularization decreases accuracy and increases visible-token cost. This suggests that shorter generation need not compromise answer quality. The reward penalizes correct responses longer than the within-group mean correct-response length, while incorrect responses receive zero reward regardless of length. This design encourages concise generation while retaining correctness as a prerequisite for positive reward, supporting a better balance between answer accuracy and visible-token cost during inference.

Table 2: Ablation studies of SoftPaths on GSM8K under a four-candidate decoding budget. Stage I ablates latent-path construction, while Stage II ablates GRPO and its reward components. Tok. denotes average visible tokens generated per query, including all four responses.

Stage	Variant	Acc. (%)↑	Tok.↓
Stage I	**SoftPaths-SFT (Ours)**	**89.76**	**818.8**
	w/o path contrastive loss	88.40	840.0
	w/o MoE projector	89.01	845.6
Stage II	**SoftPaths (Ours)**	**90.67**	**786.8**
	w/o length reward	89.99	816.0
	w/o format & length rewards	90.14	834.8
	w/o GRPO	89.76	818.8

5 Conclusion

In this work, we introduced **SoftPaths**, a two-stage framework for learning functionally diverse latent reasoning paths. In the first SFT stage, an MoE projector constructs multiple input-dependent latent conditions by composing shared projection experts through path-specific routers. Answer supervision preserves task-relevant information, while a contrastive diversity objective mitigates path collapse. In the second stage, the learned experts are frozen, and the routers are activated round-robin across training instances to construct path-specific latent conditions. Conditioned on each path, GRPO optimizes the backbone using a correctness-gated length reward that prioritizes answer correctness while discouraging unnecessary verbalization. This path-conditioning strategy enables the backbone model to better exploit diverse latent guidance for accurate and concise reasoning. Experiments across multiple reasoning benchmarks show that SoftPaths outperforms competitive explicit and latent CoT methods while requiring fewer generated tokens.

AI USE STATEMENT

We used generative AI tools for two limited purposes during the preparation of this manuscript. First, we used them to improve the language, grammar, clarity, and organization of the text. Second, we used them to assist with the design and refinement of selected figures. All AI-assisted text and visual materials were reviewed, edited, and finalized by the authors. No AI tools were used to generate data, conduct experiments, or make scientific decisions. The authors take full responsibility for the content of this work.

ETHICS STATEMENT

This work studies latent reasoning and inference efficiency using publicly available benchmarks, including GSM8K, ASDiv-Aug, MATH, CommonsenseQA, and StrategyQA. The study does not involve human participants, user studies, personal data, or sensitive information. We use these datasets solely for research evaluation and follow their respective licenses and usage terms. The proposed method is intended to improve the efficiency and reliability of language-model reasoning. However, benchmark accuracy does not guarantee factual correctness, robustness, or safe behavior in real-world settings. Model outputs may still contain errors or undesirable content, and SoftPaths should not be used as a substitute for human judgment in high-stakes domains such as healthcare, law, or finance. We take responsibility for the ethical use and final content of this work.

REPRODUCIBILITY STATEMENT

The main text and appendices provide all details necessary to reproduce our work, including the model architecture, training procedures, objective functions, reward design, hyperparameters, datasets, evaluation protocols, and inference settings. These descriptions are sufficient to implement SoftPaths and reproduce the reported experiments.

REFERENCES

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Jeffrey Cheng and Benjamin Van Durme. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*, 2024.

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

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

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

Liangliang Dong, Lianlei Shan, and Shuaimin Li. Self-softcot: A self-consistent framework via position-aware latent space reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32393–32414, 2026.

Razvan-Gabriel Dumitru, Darius Peteleaza, Vikas Yadav, and Liangming Pan. ConciseRL: Conciseness-guided reinforcement learning for efficient reasoning models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 17099–17123, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.927. URL https://aclanthology.org/2025.findings-emnlp.927/.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. In *International Conference on Learning Representations*, volume 2024, pp. 27896–27923, 2024.

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.

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

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.

Yinhan He, Wendy Zheng, Yaochen Zhu, Zaiyi Zheng, Lin Su, Sriram Vasudevan, Qi Guo, Liangjie Hong, and Jundong Li. Semcot: Accelerating chain-of-thought reasoning through semantically-aligned implicit tokens. *Advances in Neural Information Processing Systems*, 38:43455–43485, 2026.

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.

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.

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 *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. Cot-valve: Length-compressible chain-of-thought tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6025–6035, 2025.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing english math word problem solvers. In *Proceedings of the 58th annual meeting of the Association for Computational Linguistics*, pp. 975–984, 2020.

Ziqing Qiao, Yongheng Deng, Jiali Zeng, Dong Wang, Lai Wei, Guanbo Wang, Fandong Meng, Jie Zhou, Ju Ren, and Yaoxue Zhang. Concise: Confidence-guided compression in step-by-step efficient reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 8021–8040, 2025.

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

Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. Dast: Difficulty-adaptive slow-thinking for large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 2322–2331, 2025a.

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

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

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

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

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

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. Softcot: Soft chain-of-thought for efficient reasoning with llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23336–23351, 2025a.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. Softcot++: Test-time scaling with soft chain-of-thought reasoning. *arXiv preprint arXiv:2505.11484*, 2025b.

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.

Zhen Zhang, Xuehai He, Weixiang Yan, Ao Shen, Chenyang Zhao, and Xin Wang. Soft thinking: Unlocking the reasoning potential of llms in continuous concept space. *Advances in Neural Information Processing Systems*, 38:168990–169012, 2026.

A Additional Method Details

A.1 Additional Details of Latent-Path Learning

Warm-up and Projector Optimization. Stage I consists of a short task warm-up followed by latent-path optimization. During warm-up, the assistant model, base model, and MoE projector remain frozen, and only a temporary LoRA adapter attached to the base model is optimized using the full response $Y^{\text{full}} = [R; A]$. The rationale R is included in this phase to provide a task-adapted decoding interface before learning the latent paths. After warm-up, the temporary adapter is frozen together with the assistant and base models, and only the MoE projector is optimized. Unlike warm-up, projector optimization uses only the final answer A as supervision. Stage II subsequently initializes a fresh LoRA adapter for latent-conditioned reinforcement learning.

Answer-Loss Masking. During projector optimization, only tokens belonging to the final answer A contribute to the autoregressive cross-entropy objective. The prompt and latent-placeholder positions serve only as conditioning context and therefore do not contribute to the prediction loss. Thus, the MoE projector is trained to encode information useful for predicting the final answer without explicitly reproducing the textual rationale.

Contrastive Pair Construction. For each latent path $T_{\text{soft}}^{(k)}$, we apply stochastic dropout to obtain a perturbed view $\widetilde{T}_{\text{soft}}^{(k)}$. The pair $(T_{\text{soft}}^{(k)}, \widetilde{T}_{\text{soft}}^{(k)})$ is treated as a positive pair, whereas $(T_{\text{soft}}^{(k)}, \widetilde{T}_{\text{soft}}^{(j)})$ for $j \neq k$ forms a negative pair. Similarity is computed using cosine similarity over the flattened latent-path representations. This construction encourages invariance to stochastic perturbations within each path while maintaining separation among different path-specific representations generated for the same input.

Table 3: Summarized single-response comparison on GSM8K. Highlights indicate redundant visible reasoning, while red and green text denote incorrect and correct reasoning, respectively. Tok.denotes the number of tokens in the complete response.

Case: Resolving Multiple Ingredient Proportions
Problem. Tanya fills a 10-ounce jar with a mixture of salt, oil, fragrance, citrus zest, and sugar. She uses equal amounts of citrus zest and fragrance and equal amounts of salt and sugar. The amount of oil is twice the amount of salt, and the amount of salt is twice the amount of zest. How many ounces of oil does she use? **Correct relation.** Let z denote the amount of zest. Then $z + z + 2z + 2z + 4z = 10 \Rightarrow z = 1 \Rightarrow \text{oil} = 4$.
CoT: Incorrect **Tok.: 470** **Critical error.** The response first derives $z = f$ but then introduces the incompatible relations $s = z$ and $s = 2z$, preventing it from constructing the correct capacity constraint. Instead of resolving this inconsistency, the response repeatedly considers alternative interpretations of the contradictory relations. It ultimately uses $z = 10/4$ without accounting for all five ingredients. **Final answer:** 2.5 (Incorrect).
SoftCoT: Correct **Tok.: 362** **Core reasoning.** The response preserves the required relations and obtains $z + z + 2z + 2z + 4z = 10 \Rightarrow z = 1 \Rightarrow \text{oil} = 4$. However, it describes each operation separately, including variable definition, relation substitution, equation construction, coefficient collection, equation solving, and back-substitution. **Final answer:** 4 (Correct).
SoftPaths: Correct **Tok.: 298** **Core reasoning.** Using a single reference variable Z, the response directly obtains $4Z+Z+Z+2Z+2Z = 10 \Rightarrow Z = 1 \Rightarrow \text{oil} = 4$. **Final answer:** 4 (Correct).
Comparison. CoT introduces inconsistent constraints and produces an incorrect answer, whereas SoftCoT reaches the correct answer through additional visible steps. SoftPaths preserves the required constraint structure in a more compact response, reducing visible generation by 36.6% relative to CoT and by 17.7% relative to SoftCoT.

A.2 Additional Details of Latent-Conditioned GRPO

Latent-Conditioned Rollout Sampling. During Stage II, the assistant model and MoE projector remain frozen, and only a freshly initialized base-model LoRA adapter is optimized. Under the active latent condition $\mathbf{X}_b^{(k)}$ selected as described in Sec. 3.4, let $\pi_{\omega_{\text{old}}}$ denote the rollout policy, where ω_{old} represents the policy parameters at the time of response generation. These parameters remain fixed while the sampled responses are used to update the current policy parameters ω. Conditioned on $\mathbf{X}_b^{(k)}$, the rollout policy samples a group of G textual responses:

$$Y_g \sim \pi_{\omega_{\text{old}}}\left(\cdot \mid \mathbf{X}_b^{(k)}\right), \qquad g = 1, \ldots, G, \tag{10}$$

where g indexes the responses within the rollout group.

Group-Relative Advantage Estimation. Given the rewards $\{r_g\}_{g=1}^{G}$ of a rollout group, GRPO Shao et al. (2024) computes the group-relative advantage of each response as

$$\widehat{\mathcal{A}}_g = r_g - \frac{1}{G}\sum_{j=1}^{G} r_j. \tag{11}$$

Using the group mean as the baseline avoids training a separate critic model. The resulting advantages are used in the standard clipped GRPO objective with $\pi_{\omega_{\text{old}}}$ as the rollout policy and a frozen initial policy π_{ref} for KL regularization. All policy probabilities are evaluated under the same latent-conditioned input $\mathbf{X}_b^{(k)}$, and the KL penalty is applied to the policy objective rather than included in the scalar reward. Only the LoRA parameters ω are updated during Stage II.

B Additional Experimental Details

B.1 Implementation Details

Implementation Details. We use LLaMA-3.1-8B-Instruct as the base model and LLaMA-3.2-1B-Instruct as the assistant model Grattafiori et al. (2024). The MoE projector uses four shared experts, top-2 routing, and four latent paths, with eight continuous thought vectors per path during projector training and one during GRPO and inference. SoftPaths-SFT uses one warm-up epoch followed by four projector-only epochs with a learning rate of 8×10^{-5}, while GRPO optimizes a rank-16 LoRA adapter for one epoch with a learning rate of 1×10^{-6} using $G = 8$ rollouts, a temperature of 1.0, and top-p of 0.95.

B.2 Evaluation Protocol

Evaluation Protocol. We report answer accuracy for candidate budgets of $M = 1$ and $M = 4$. At $M = 4$, sampling baselines generate four responses from a shared condition, whereas SoftPaths decodes one response from each latent path; the extracted answers are aggregated by majority voting. All methods use the same task instructions, answer formats, and applicable sampling parameters, with a temperature of 0.7 and top-p of 0.95. Each trained method uses a fixed checkpoint across decoding seeds. The efficiency analysis reports the visible-token cost defined in Sec. 4.3.

C Qualitative Examples

C.1 Qualitative Case Study

Table 3 compares single responses to a GSM8K problem involving interdependent proportional constraints. A correct solution requires the model to represent all relations within a consistent constraint system. All methods are evaluated with $M = 1$, thereby isolating reasoning correctness and visible generation length without voting or post-hoc candidate selection.

This example illustrates the distinction between reasoning correctness and visible generation length. The table summarizes the complete responses, and the highlighted descriptions do not represent an exact token-level alignment. Table 1 and Fig. 3 provide the aggregate evidence for accuracy and efficiency, respectively. This case serves as a qualitative illustration rather than causal evidence for a specific mechanism.