# CLUE: CONTRASTIVE LEARNING TO UNLOCK EFFECTIVE LATENT REASONING IN LLMS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Latent reasoning has emerged as a popular alternative to textual chain-of-thought by carrying intermediate information through continuous representations. As LLMs are pretrained only on text rather than continuous states, post-hoc training is required for them to effectively process latent representations. However, under standard outcome-based reinforcement learning, this inherent text bias can limit the model's effective use of latent information, as the objective rewards final-answer correctness without explicitly encouraging the model to exploit the information conveyed by the latent representations. We address this challenge by introducing a decoupled architecture paired with a novel training objective. Because latent thoughts represent a fundamentally different knowledge formulation than text, our architecture separates latent generation from textual decoding: a trainable latent generator guides a lightweight, frozen text model, forcing learned adaptation exclusively through the latent channel. To explicitly encourage reliance on these continuous thoughts, we introduce a **C**ontrastive **L**earning objective to **U**nlock **E**ffective latent reasoning in LLMs (CLUE). CLUE adds a contrastive loss to the standard GRPO objective, ensuring that correct responses yields a higher log-likelihood under their own positive latent compared to negative latents sampled from other inputs. This provides dense supervision for latent utilization alongside answer correctness, without requiring ground-truth reasoning traces. Trained on OpenThoughts30K and evaluated across diverse mathematical reasoning and scientific question-answering benchmarks, our method improves average accuracy by 5.6 percentage points over direct optimization of the text-based model and outperforms related work by up to 3.5 percentage points.

## 1 INTRODUCTION

Chain-of-thought (CoT) improves language models' performance by allocating additional computation to intermediate reasoning steps (Wei et al., 2022; Kojima et al., 2022). However, expressing each step through autoregressive text generation reduces inference throughput. While language models must generate their final answers in text to communicate with humans, intermediate computations do not inherently require this discrete bottleneck. Latent reasoning offers an alternative by carrying intermediate information through continuous representations without decoding each step into language (de Luca & Fountoulakis, 2024; Hao et al., 2025; Shen et al., 2025; Xu et al., 2025a). To achieve this, recent methods introduce continuous reasoning by appending soft tokens (Zhang et al., 2025), undecoded hidden states (Shen et al., 2025; Hao et al., 2025; Yue et al., 2025), or learnable embeddings (Xu et al., 2025a; Aichberger & Hochreiter, 2026) directly to the processing sequence. However, adapting pretrained LLMs to use these representations remains challenging, as pretraining teaches them to process text tokens rather than compact latent states. Overcoming this textual prior is complicated by the lack of ground-truth latent thoughts, requiring supervision through final answers or alignment with textual reasoning traces (Kuzina et al., 2026; He et al., 2025; Cheng & Durme, 2024). Yet outcome supervision rewards correctness regardless of whether latent information contributed, allowing a capable LLM to rely on its existing textual reasoning without learning to use the latent channel effectively.

Following SoftCoT and SemCoT (Xu et al., 2025a; He et al., 2025), we explore this dynamic through a decoupled architecture in which a trainable, high-capacity latent generator conditions a lightweight, frozen text model. Freezing the text model constraints learned adaptation to the latent



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.

channel while retaining its pretrained knowledge for textual decoding. This enables a modular form of adaptation (Dery et al., 2026): the latent generator learns task-specific guidance, while autoregressive text generation is delegated to the computationally cheaper model.

While prior work motivates our decoupled architecture and the use of reinforcement learning (Xu et al., 2025a; He et al., 2025; Yue et al., 2025), outcome supervision does not explicitly reward the contribution of latent content. Rewards based on the final textual answer train the latent generator indirectly, but do not distinguish whether its representations provide useful guidance for the current problem. Without an explicit incentive, the text model may rely on its pretrained textual reasoning rather than draw on problem-specific latent guidance. Our experiments reveal this disconnect: standard outcome supervision through Group Relative Policy Optimization (GRPO) Shao et al. (2024) improves task performance, yet paired latents offer no statistically detectable accuracy advantage over those generated from unrelated problems. This suggests that the latent channel may only learn generic conditioning, resembling a learned soft prompt in function, rather than guidance tailored to each query. We therefore argue that training should optimize both answer correctness and the effective use of latent content to encourage the text model to draw on useful latent guidance rather than rely solely on its pretrained textual reasoning.

Building on this intuition, we introduce a **C**ontrastive **L**earning objective to **U**nlock **E**ffective latent reasoning in LLMs (CLUE). CLUE complements GRPO by using verified correct model-generated responses as training targets, scoring each response under its paired latents and negative latents generated from other inputs. To quantify relative latent utility, CLUE measures the difference in log-likelihood between these conditions and introduces a contrastive loss to increase this margin. This provides dense, input-specific supervision that final-answer correctness alone does not capture.

Our experiments on mathematical and scientific reasoning benchmarks show that CLUE outperforms the evaluated latent reasoning baselines on average at both model scales. Using Qwen3-4B and Qwen3-1.7B, it also exceeds direct text-based GRPO fine-tuning of Qwen3-1.7B, while achieving comparable average performance to GRPO on Qwen3-4B when paired with Qwen3-8B. Ablations show that contrastive training improves accuracy and produces a more consistent observed benefit from problem-specific latents, while probing experiments suggest that latent guidance varies by domain and shapes the generated reasoning traces. Compared to exchanging textual reasoning between models, latent communication achieves higher accuracy and improved token efficiency under the tested setups. Together, our findings show that improved accuracy can coexist with little measurable benefit from problem-specific latent information, motivating our decoupled architecture and contrastive objective to explicitly encourage and evaluate useful latent communication.

## 2 Related Work

**Chain-of-Thought.** Scaling test-time compute by generating intermediate reasoning tokens has significantly enhanced the ability of large language models to solve complex, multi-step problems (Nye et al., 2022). Central to this paradigm is Chain-of-Thought (CoT) prompting (Wei et al., 2022; Kojima et al., 2022), which asks models to explicitly articulate intermediate computational steps before producing a final answer. In addition, methods leverage scaled reasoning computation for training through supervised fine-tuning on self-generated trajectories (Zelikman et al., 2022) or reinforcement learning (DeepSeek-AI, 2025). However, all of these methods fundamentally rely on externalizing intermediate computational states into a human-readable format through autoregressive text generation, which may not necessarily the most efficient medium for a model's internal computation (de Luca & Fountoulakis, 2024; Hao et al., 2025).

**Latent Reasoning.** Latent reasoning instead maintains reasoning states as continuous internal representations, bypassing the discrete text bottleneck. While some architectures natively maintain internal continuous states, such as through depth-recurrence in looped transformers (Giannou et al., 2023; Saunshi et al., 2025; OpenAI, 2026), a broad body of work focuses to adapt existing LLMs to perform reasoning directly in the latent space. Here, these models are trained to externalize and consume continuous representations instead of text (Hao et al., 2025; Shen et al., 2025). Learning this behavior is challenging because pretraining emphasizes text tokens rather than latent reasoning states: appending unguided continuous vectors or pause tokens (Goyal et al., 2024) can fail to improve accuracy and may degrade performance in longer contexts (Lanham et al., 2023). To

overcome these training challenges, latent representations are typically aligned using dense textual supervision, e.g., by progressively replacing textual reasoning steps with continuous latents (Hao et al., 2025), employing distillation Shen et al. (2025), or enforcing compression (Kuzina et al., 2026; Cheng & Durme, 2024) and semantic similarity constraints (He et al., 2025) against ground-truth reasoning traces. Rather than fully replacing text, latent states can also guide textual reasoning (Yue et al., 2025). To preserve textual capabilities, multi-model architectures separate latent computation from text generation using trainable coprocessors (Kuzina et al., 2026; He et al., 2026) or dedicated assistants (Xu et al., 2025a;b; He et al., 2025). However, this separation alone does not resolve alignment: without explicit supervision, models may still favor their pretrained textual priors over relevant non-textual signals (Deng et al., 2025). CLUE addresses this challenge by combining a decoupled architecture with a contrastive objective alongside outcome-based RL, encouraging effective latent utilization under answer-only supervision without requiring intermediate reasoning traces.

# 3 METHOD

## 3.1 ARCHITECTURE

We model latent reasoning by explicitly decomposing latent and textual generation across two separate models, similar to (He et al., 2025; Xu et al., 2025a; He et al., 2026). Specifically, a latent decoder $L_\theta$ learns to generate continuous representations that a second model $T_\psi$ reads out to generate a textual answer. The main advantage of this approach is specialization, as each model is solely responsible for its respective domain. By keeping the text model's pretrained parameters unchanged, the latent model can learn to generate representations against a fixed target, allowing for easier alignment of the two models.

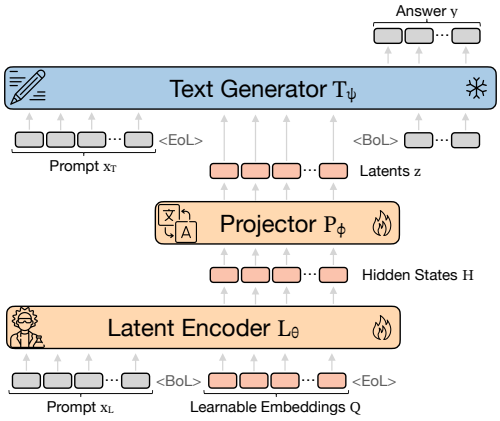


Figure 1: **Model Architecture.** We decompose latent and text generation across two specialized models. A high-capacity latent generator uses learnable embeddings to produce task-specific continuous representations through a projection. The lightweight, frozen text model conditions on these latents to decode the final textual answer.

**Latent State Generation.** To generate the continuous reasoning states, the latent encoder processes the input context $x_L$ together with $K$ learned embeddings $Q \in \mathbb{R}^{K \times d_L}$ in a single forward pass. We collect the final hidden states corresponding to these embeddings as $H(x_L) \in \mathbb{R}^{K \times d_L}$ and project them into the embedding space of $T_\psi$:

$$z = L_\theta(x_L) = P_\phi(H(x_L)) \in \mathbb{R}^{K \times d_L}. \quad (1)$$

Here, $d_L$ and $d_T$ denote the hidden dimension of the latent encoder and the embedding dimension of the text model, respectively. The projection $P_\phi$ is a residual MLP applied to each vector:

$$P_\phi(h) = D_\phi(h) + W_2\ \mathrm{GELU}\left(W_1\,\mathrm{LN}(h) + b_1\right) + b_2, \quad (2)$$

where $D_\phi$ is a learned affine map when the dimensions differ, and the identity otherwise. We jointly train the latent encoder's language-model weights, the embeddings $Q$, and the projection, collectively denoted by $\theta$.

**Textual Decoding.** The text generator processes its input $x_T$ together with the latent reasoning states $z$, which act as an input-conditioned soft prompt conveying intermediate information. The text decoder autoregressively generates additional textual reasoning steps and the final answer:

$$p_\psi(y \mid x_T, z) = \prod_{\ell=1}^{|y|} p_\psi(y_\ell \mid x_T, z, y_{<\ell}). \quad (3)$$

During this process, the text generator's parameters $\psi$ remain frozen, while gradients propagate through its computation back to the encoder-side parameters $\theta$, optimizing the latent states purely for downstream task utility.

**Input Formatting.** Both models receive the same problem, $x_L = x_T = x$, formatted using their respective tokenizers and chat templates. We place a latent block immediately after each template-generated *assistant* prefix, where the model would otherwise begin generating its response. Each block is enclosed by model-specific, learnable <BoL> and <EoL> boundary tokens.

**Model Selection.** Inspired by the allocation of complementary skills across specialized agents in multi-agent systems Wu et al. (2024); Hong et al. (2024); Chen et al. (2024), we implement our architecture with a deliberate capacity gap between its two components. Specifically, the larger latent encoder produces input conditioned representations for the smaller, frozen text generator. As a result, the stronger latent model can process the input and provide complementary information that the smaller text model could not easily derive on its own. This asymmetric design translates perfectly to distributed cloud and edge deployments. A large latent model hosted on the cloud can perform complex reasoning and transmit compressed continuous states to a smaller text model on an edge device. Because sharing latent states avoids streaming explicit textual reasoning steps, this setup is not only highly bandwidth efficient but also provides an inherent privacy advantage: the intermediate reasoning remains in an uninterpretable vector format, protecting sensitive data during transmission. Ultimately, this design enables a modular form of adaptation where the latent encoder provides specialized guidance while offloading standard autoregressive generation to the computationally cheaper local model.

### 3.2 Trainnig Latent Reasoning with Contrastive Learning

**Optizing Answer Correctness.** We first aim to learn continuous reasoning states that improve task performance by conveying actionable information to the text model. Because the $T_\psi$ remains frozen, the objective is to optimize the latent encoder's parameters $\theta$ to maximize expected task utility:

$$\max_{\theta} \mathbb{E}_{(x,t)\sim\mathcal{D}} \mathbb{E}_{y\sim p_\psi(\cdot|x_T, L_\theta(x_L))} \left[U(y,t)\right], \tag{4}$$

where $t$ denotes the ground truth target and $U$ measures task utility. Rather than relying on dense, token-level supervision, we directly optimize this objective via policy gradients using Group Relative Policy Optimization (GRPO). To measure task utility $U(y_i, t)$, we use a binary reward $r_i$ that measures the exact match of the generated final answer against the target. During training, GRPO samples a group of $Gg$ textual responses $\{y_1, y_2, \ldots, y_g\}$ from the joint model, computes their advantage $\hat{A}_i = \frac{r_i - \text{mean}(\{r_1, r_2, \ldots, r_g\})}{\text{std}(\{r_1, r_2, \ldots, r_g\})}$, and updates the policy by maximizing the following clipped surrogate objective:

$$\nabla_\theta \mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}, \{(y_i, z_i)\}_{i=1}^{g} \sim \pi_\theta(\cdot|x)} \left[ \frac{1}{g} \sum_{i=1}^{g} \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \nabla_\theta \log \pi_\theta(y_{i,t} | x, y_{i,<t}, z_i) \hat{A}_{i,t} \right] \tag{5}$$

where the policy $\pi_\theta$ denotes the joint distribution of the trainable latent encoder $L_\theta$ and the frozen text model $T_\psi$.

However, optimizing the objective in Equation 5 reveals a critical limitation. The advantage $\hat{A}_{i,t}$ relies exclusively on the final textual output $y_i$, providing no direct supervision on the latent states $z_i$. Consequently, the optimization process can treat the latents as generic tunable tokens, allowing them to lose their binding to the paired input query $x_i$ with respect to the text model. To prevent the latents from collapsing into query independent signals, training must focus on a second dimension alongside final answer correctness: the effective utilization of the latents. Since the lack of ground truth prevents direct supervision on $z_i$, we cannot simply force the latents toward a target representation. Instead, we constrain the optimization through a contrastive approach.

**Contrastive Trajectory Scoring.** To enforce problem-specific latent usage without relying on reference reasoning traces, CLUE introduces a contrastive, likelihood-based utility criterion. By

scoring identical verified trajectories under both the paired and random latents, we optimize for a utility margin that explicitly rewards information exchange through the latent channel.

For a problem $x_i$ and paired latent message $z_i = Z_{\theta,\phi}(x_i)$, let $\tau_{ij} = (y_{ij,1}, \ldots, y_{ij,T_{ij}})$ be the $j$th receiver trajectory sampled under this message. We define its normalized log-likelihood on a set of token positions $\mathcal{S}_{ij}$ as

$$\ell_\psi(\tau_{ij}; x_i, z) = \frac{1}{|\mathcal{S}_{ij}|} \sum_{t \in \mathcal{S}_{ij}} \log p_\psi \left( y_{ij,t} \mid x_i, z, y_{ij,<t} \right). \tag{6}$$

For each problem, we construct $K$ pertubed latents $\tilde{z}_{ik} = Z_{\theta,\phi}(\tilde{x}_{ik})$ from negative examples $\tilde{x}_{ik} \neq x_i$ drawn from the training set and compute the pairwise likelihood gaps

$$d_{ijk} = \ell_\psi(\tau_{ij}; x_i, z_i) - \ell_\psi(\tau_{ij}; x_i, \tilde{z}_{ik}), \tag{7}$$

For a given problem $x_i$, this measures whether the observed trajectory $\tau_{ij}$ is more likely under its paired latents $z_{ik}$ than under the sampled pertubed latents $\tilde{z}_{ik}$.

We apply the margin objective only to trajectories with verified correct final answers, encouraging paired-message preference for successful solutions rather than incorrect outputs. Let $\mathcal{D}_+$ denotes the distribution of selected correct trajectories from the current policy's rollouts, then we target an average likelihood advantage of at least $m > 0$:

$$\mathbb{E}_{(i,j)\sim\mathcal{D}_+} \left[ \frac{1}{K} \sum_{k=1}^{K} d_{ijk} \right] \geq m. \tag{8}$$

**Practical implementation.** For a training batch $\mathcal{B}$, let $\mathcal{I}_\mathcal{B}$ contain the selected verified trajectories. For each problem, we sample $K$ negative latents and optimize the joint objective:

$$\mathcal{L} = \mathcal{L}_{\text{GRPO}} + \lambda \mathcal{L}_{\text{CLUE}}(\mathcal{B}), \quad \mathcal{L}_{\text{CLUE}}(\mathcal{B}) = \frac{1}{K|\mathcal{I}_\mathcal{B}|} \sum_{(i,j)\in\mathcal{I}_\mathcal{B}} \sum_{k=1}^{K} \rho_\beta(-g_{ijk}), \tag{9}$$

where $\rho$ is a the softplus function and $g_{ijk} = \text{sg}[\ell_\psi(\tau_{ij}; x_i, z_i)] - \ell_\psi(\tau_{ij}; x_i, \tilde{z}_{ik})$. Note that we detach gradients through the paired latents likelihood in the margin. This directs CLUE toward reducing the likelihood of verified trajectories under negative messages and provides a complementary learning signal to GRPO, which already encourages higher likelihood for relative successful trajectories under the paired latents.

We control the contribution of $\mathcal{L}_{\text{CLUE}}$ adaptively. The dual variable $\lambda$ is updated by: $\lambda \leftarrow \text{clip}(0, \lambda + \eta(m - \bar{g}), \lambda_{\max})$, where $\bar{g}$ is the mean gap over all $g_{ijk}$ and $\eta > 0$ the controller step size, and $\Pi$ denotes projection. By decreasing $\lambda$ when the mean gap exceeds the target, the controller reduces pressure to further enlarge the margin.

## 4 Experiments

**Models and Baselines** We compare CLUE against two state-of-the-art latent reasoning approaches similar to our setup. We evaluate SoftCoT (Xu et al., 2025a), which uses an assistant LLM to generate latent representations for a textual base LLM. We use the same model combinations as in CLUE to initialize the assistant and receiver. Because their original method freezes both models and fine-tunes only an intermediate projection via SFT, we adapt their architecture to use GRPO for a fair comparison, denoting it as *SoftCoT*. As a native on-policy training method, we evaluate *HRPO* (Yue et al., 2025), which is an RL-based hybrid approach that integrates prior hidden states into sampled tokens via a learnable gating mechanism, utilizing both discrete and continuous representations. We retain the LoRA fine-tuning strategy used in the authors' implementation. Finally, to establish the performance of our text generating LLM without latent reasoning, we evaluate the pretrained model without fine-tuning (*zero-shot*) and after further GRPO training (*GRPO*),

**Training and Evaluation** We use the mathematical subset of OpenThoughts (Guha et al., 2025) as the training dataset, with exact matching of the textual answer as reward function. For CLUE

and SoftCoT, we use Qwen3-4B-Instruct-2507→Qwen3-1.7B and Qwen3-8B→Qwen3-4B as the model pairings (Yang et al., 2025), with 16 latent slots by default. All models are the instruct tuned versions with thinking diabled. We evaluate mathematical reasoning on GSM8K (Cobbe et al., 2021), Math500, consisting of 500 challanging problems from the Math dataset (Hendrycks et al., 2021), and AIME24 and AIME25 testing advanced high-school competition problem solving. We retain 500 examples from our training dataset to evaluate in-domain evaluation. For out-of-domain evaluation, GPQA tests graduate-level scientific knowledge and reasoning. We score extracted final answers using normalized exact answer matching and report *avg@8* accuracy on all datasets. For detailed settings and configurations we defer to Appendix A.1.

# 5 RESULTS

Table 1 compares CLUE with text-only and latent reasoning baselines across mathematical and scientific reasoning benchmarks at two text model scales. CLUE provides strong gains with the Qwen3-1.7B text model, outperforming direct GRPO fine-tuning across all evaluated benchmarks. This shows that even with its parameters frozen, the text model can outperform its directly GRPO-fine-tuned counterpart when guided by latents from a trainable generator. Compared with SoftCoT, which uses the same model combination but freezes both language models, CLUE performs substantially better on the mathematical benchmarks. One possible explanation is that generating useful latent representations requires adaptation beyond the model's pretrained ability to generate text. Whereas SoftCoT relies on the projection alone to bridge this gap, CLUE also trains the latent generator to produce continuous representations that support reasoning in the frozen text model.

Scaling up to Qwen3-8B→Qwen3-4B yields a more mixed comparison with direct GRPO. CLUE remains competitive and leads on AIME24 and GPQA, but direct GRPO achieves the highest average accuracy. We hypothesize that the larger pairing leaves less room for the generator to contribute beyond the text model's own reasoning. This depends not only on their initial capability gap, but also on what the generator learns during training. Since the generator receives feedback through the frozen text model, its adaptation may be limited by the guidance the receiver can use. This interpretation is consistent with SoftCoT's substantially weaker performance in the larger pairing: with both models frozen, it must rely on the projection alone to make their existing capabilities useful for communication. Together with the stronger gains of CLUE in the Qwen3-4B→Qwen3-1.7B pairing, these observations suggest that effective latent communication benefits from both complementary model capabilities and the flexibility to learn how to convey them.

Despite training only on mathematical problems, CLUE outperforms direct GRPO on GPQA at both scales. SoftCoT's leading GPQA result in the smaller pairing further suggests that preserving the text model's pretrained capabilities through a frozen receiver may benefit transfer beyond the training domain.

## 5.1 CONTRASTIVE TRAINING UNLOCKS MORE EFFECTIVE LATENT TOUGHTS

To isolate the contribution of the contrastive trajectory scoring within CLUE, we evaluate a baseline that optimizes the task objective (Equation 4) only using final answer supervision, i.e., relying entirely on the gradients provided by the standard GRPO loss. We refer to this setup as *CLUE w.o contr. loss*. We further investigate whether the performance gains of CLUE stem from the text model actively utilizing the latents during reasoning by intervening on the latent states during evaluation. For a given problem $i$, we compare performance using the *paired latents* $z_i = L_\theta(x_i)$ against *random latents* $\tilde{z}_i = L_\theta(x_j)$, where $j \neq i$ is sampled uniformly from the evaluation set. Crucially, substituting latents from other problems provides a more rigorous test than artificial corruption with noise or other perturbations in the latent representation space. Applying random noise, for example, risks pushing the continuous states out of distribution with respect to the text model, which could artificially degrade performance simply due to a distribution shift. In contrast, using latents generated from other valid queries guarantees that the representations remain in distribution. In our experiments, the text model retains its paired problem input $x_T = x_i$ and decoding protocol in both conditions.

**Influence of Contrastive Loss** The contrastive objective improves task accuracy while making paired latents more useful than random replacements. Under identical evaluation conditions, CLUE

Table 1: **Performance across reasoning benchmarks.** We report avg@8 accuracy on math reasoning datasets and the OOD scientific reasoning benchmark GPQA. CLUE and SoftCoT pair the Qwen3-1.7B and Qwen3-4B text models with Qwen3-4B and Qwen3-8B latent generators. The horizontal grey line separates textual from latent reasoning methods. Best results are shown in bold, and second-best results are underlined. Standard deviations are reported in Appendix A.3

| | ID | | | | OOD |
|---|---|---|---|---|---|
| | GSM8K | Math-500 | AIME24 | AIME25 | GPQA |
| | | | *Qwen3-1.7B* | | |
| zero-shot | 81.24 | 70.62 | 11.67 | 10.42 | 28.91 |
| GRPO | 85.65 | <u>80.10</u> | <u>29.58</u> | <u>20.41</u> | 26.58 |
| HRPO | 84.02 | 74.80 | 20.00 | 13.33 | 32.25 |
| SoftCoT | <u>88.45</u> | 76.28 | 15.83 | 14.58 | **37.62** |
| CLUE (ours) | **89.10** | **84.68** | **34.58** | **27.08** | <u>35.04</u> |
| | | | *Qwen3-4B* | | |
| zero-shot | 90.55 | 81.68 | 43.33 | 19.58 | 40.71 |
| GRPO | **92.87** | **91.10** | <u>55.83</u> | **49.58** | <u>45.95</u> |
| HRPO | 91.52 | 84.77 | 43.33 | 26.67 | 44.44 |
| SoftCoT | 91.40 | 87.40 | 26.67 | 23.33 | 41.91 |
| CLUE (ours) | <u>92.57</u> | <u>89.99</u> | **56.25** | <u>44.54</u> | **48.67** |

benefits from paired latents on all three datasets, with statistically significant [1] advantages on Math-500 and OpenThoughts, while the GPQA result remains suggestive. Without contrastive training, random replacements instead yield comparable or higher accuracy, although none of these differences is statistically significant. SoftCoT exhibits a similar pattern, with no statistically detectable accuracy difference between paired and random latents. This pattern suggests that outcome supervision alone can favor generic conditioning that remains useful even when messages are exchanged between problems. Small gains from random replacements may then reflect stochastic decoding variability rather than better guidance. CLUE, by contrast, shows a consistent observed preference for information tied to the current problem.

**Domain Transferability** Building on this finding, we investigate the level of semantic granularity at which latents provide useful steering within CLUE. Rather than evaluating purely random pairings, we systematically perform latent swapping based on task type to determine if the generated latents encode task dependent independent information. To this end, we utilize the GPQA dataset, which categorizes questions hierarchically into major scientific domains and specific subdomains. By replacing a target question's latent with one generated from a problem in a completely different major domain, we can isolate the effect of domain mismatch. We measure the resulting performance gap to determine the extent to which the models exchange finer grained, domain specific reasoning information.

As shown in Figure 2, the text model's performance varies depending on the source domain of the transferred latent. In general, cross-domain transfer generally has a negative effect on performance. The only exception is the application of Chemistry latents to Physics targets (+1.64%). Otherwise, domain mismatch consistently reduces accuracy, with the largest gaps occurring when sourcing latents from or targeting the Biology domain. Qualitative inspection suggests that the specific type of reasoning required by each task may explain the varying magnitudes of transfer gaps observed across domains. While Physics and Chemistry frequently require computations and structured reasoning, Biology questions more frequently involve mechanistic or factual inference. When measuring transferring latents from the exact same subdomain we observe no clear benefit over using latents from a different subdomain within the same major field (-0.68pp. vs -0.78pp.). Together, these observations

[1] Two-sided paired $t$-tests across eight decoding repetitions yield unadjusted $p = 0.041$, $0.023$, and approximately $0.25$ for CLUE, $0.53$, $0.062$, and $0.24$ w.o. contr. loss and $0.19$, $0.11$, and $0.96$ on Math-500, OpenThoughts, and GPQA, respectively. Corresponding values without contrastive training are approximately . We use a significance threshold of $p < 0.05$.

Table 2: **Ablation on Problem Specific Latent Utilization.** Accuracy with paired latents and latents generated from randomly selected other evaluation. Parentheses report changes relative to paired latents in percentage points. CLUE consistently benefits from paired latents, whereas removing the contr. loss eliminates this advantage. SoftCoT shows little change in accuracy when paired latents are replaced with random ones, suggesting limited dependence on problem-specific latent content.

| | | ID | | OOD |
|---|---|---|---|---|
| Method | Latent Source | Math-500 | OpenThoughts30K | GPQA |
| SoftCoT | Paired | 76.28 | 56.98 | **37.62** |
| | Random | 76.75 (+0.47) | 57.77 (+0.79) | 37.56 (-0.06) |
| CLUE | Paired | **84.68** | **64.60** | 35.04 |
| | Random ($\Delta$) | 84.08 (-0.60) | 63.40 (-1.20) | 33.64 (-1.40) |
| w.o. contr. loss | Paired | 82.63 | 62.80 | 28.91 |
| | Random ($\Delta$) | 82.83 (+0.20) | 63.80 (+1.00) | 30.61 (+1.70) |

motivate the hypothesis that latent transferability also depends on the compatibility of underlying reasoning operations, beyond topical similarity alone. Under this hypothesis, the model may utilize latent representations as adaptable reasoning skills rather than solely encoding factual knowledge.

**Reasoning Style** Task accuracy alone does not reveal how latent conditioning affects the reasoning traces of the frozen text model. Following Panfilov et al. (2026), we compare over 3,000 traces across GSM8K, Math-500, and GPQA, generated with paired versus random latents for CLUE and CLUE w.o. contr. loss. We train a logistic-regression classifier on hashed character 3–5-gram counts from complete responses, normalized to unit $\ell_2$ norm to reduce sensitivity to response length. Using five-fold stratified cross-validation grouped by problem, we report pooled out-of-fold ROC AUC. Permuting condition labels yields central 95% null intervals contained within $[0.47, 0.54]$ across datasets. We find that CLUE produces traces that are more stylistically distinguishable under perturbation (AUC $0.615$) compared to CLUE w.o. contr. loss, i.e., training only with final answer correctness (AUC $0.556$). This aligns with our earlier findings on active utilization: because CLUE trains the text model to rely on the latent channel for input-specific conditioning, perturbing those latents has a more measurable influence on the model's textual reasoning style than it does in the baseline.

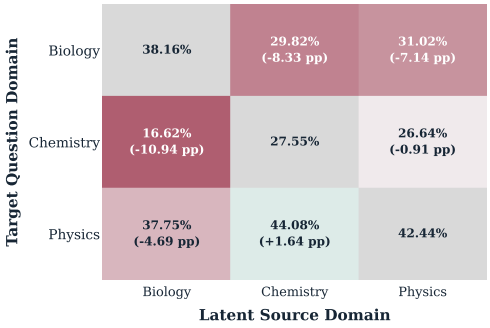


Figure 2: **Domain Transferability.** Performance degradation when replacing a question's latent within or across one from a different domain. Diagonal cells (gray boxes) show intra-domain and off-diagonal cells show cross-domain swaps. Higher $\Delta$ values represent a larger relative drop in performance. Cross-domain transfer generally degrades performance more strongly, indicating that latents encode useful domain-specific cues.

### 5.2 Comparison to Exchanging Text

To compare the usage of latent tokens against discrete tokens we introduce a text-based baseline (denoted as *Text*). In this setup, an upstream model generates a textual trace constrained by a strictly enforced generation cap that is then provided as additional input to the frozen text model. We strictly cap this generation at 128 tokens, preventing the baseline from relying on arbitrarily long reasoning traces. As shown in Table 3, discrete text communication yields substantially lower downstream performance. The *text-GRPO* baseline achieves an average accuracy of only 41.72%, trailing both *latent-GRPO* (51.73%) and CLUE (54.10%) by wide margins. Qualitative inspection reveals that under this strict budget, the upstream model often generates shallow heuristics (e.g., summarizing basic arithmetic: "Key equation: Daily earnings = (Total eggs laid - Eggs eaten - Eggs used for

Table 3: **Impact of the Typ of Communication Channel.** Performance comparison between CLUE (denoted as *Latent*) and a baseline using discrete text for information communication (*Text*). Both models use the Qwen3-4B→Qwen3-1.7B pairing, with token limits capped at 128 for text and 16 for latent representations.

| | ID | | | | OOD |
|---|---|---|---|---|---|
| Channel | GSM8K | Math-500 | AIME24 | AIME25 | GPQA |
| Text | 82.54 | 71.75 | 11.67 | 13.75 | 28.91 |
| Latent | **89.10** | **84.68** | **34.58** | **27.08** | **35.04** |

muffins)…"). Such brief textual hints may assist with simple problems, but they fail to provide any deeper insights required for complex reasoning tasks.

**Discrete Token Bottleneck.** While we find that natural language text is an inherently easier initial conditioning signal, benefiting from the frozen model's pre-existing text understanding capabilities, it ultimately imposes a low performance ceiling. Once past this initial alignment, latent tokens prove superior across two important dimensions: performance and efficiency. This text bottleneck is fundamentally tied to the limited bandwidth of discrete tokens, as current pretrained models may not be able to express complex reasoning within a small generation cap. To systematically verify this bandwidth limitation, we evaluate the conditioning efficiency of both discrete and text tokens across varying generation caps and model scales. As shown in the top panel of Figure 3, for the 4B → 1.7B configuration, latent communication rapidly achieves higher accuracy under tight token budgets compared to text-based conditioning. However, when scaling to a larger 8B → 4B configuration (Figure 3, bottom panel), the advantage provided by text conditioning remains minimal, and the relative performance gains of the latent model are less pronounced than in the smaller pairing. This is in line with our earlier observations on model scale, suggesting that useful guidance depends not only on the communication budget, but also on what the generator can contribute beyond the text model's own capabilities and how effectively it learns to communicate that information.

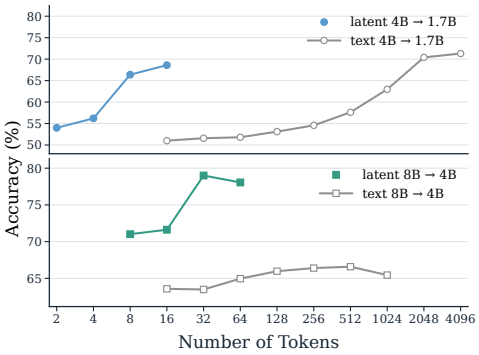


Figure 3: **Effect of thought-token budget across model scales.** We measure accuracy on OpenThoughts30K using a strict latent/text token budget.

## 6 Conclusion

Moving beyond textual chain-of-thought requires not only architectures that support continuous reasoning states, but also training objectives that make those states useful. Yet outcome-based training can improve task performance while leaving accuracy largely unchanged when we intervene on the latent channel, suggesting that higher accuracy need not reflect effective use of problem-specific latent information. CLUE addresses this gap by complementing outcome-based reinforcement learning with a contrastive objective that encourages the model to draw on problem-specific latent information without requiring reference reasoning traces. Pairing Qwen3-4B and Qwen3-1.7B for latent reasoning, CLUE achieves 54.10% average accuracy across five reasoning benchmarks, exceeding text-only GRPO fine-tuning of Qwen3-1.7B by 5.64 percentage points. These gains are accompanied by a more consistent advantage for problem-specific latents and greater sensitivity of the generated reasoning to latent interventions.Taken together, the results suggest that latent reasoning methods should explicitly evaluate and optimize the contribution of latent information to successful answers, rather than treating improved accuracy alone as evidence of effective latent reasoning. Our decoupled architecture demonstrates the value of explicitly encouraging latent utilization, motivating further work to test whether CLUE offers similar benefits across other reasoning architectures.

### AI USE STATEMENT

In this work, we used generative AI tools for: designing or providing feedback on research methodology or experiments, implementing methods, proposing or refining hypotheses, assisting with translation, supporting qualitative and thematic data analysis, interpreting results.

We have not used generative AI tools for: helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, and: generating synthetic data sets; assisting in the writing of proofs; cleaning and reformatting dataset are not applicable to this work.

Additionally, we used generative AI tools for: creating or modifying scientific figures or images, creating or editing software code, summarizing or analysing existing literature, brainstorming, editing a research paper to improve readability.

We have reviewed all AI-assisted work. All LLM-generated code and method implementations were manually reviewed and tested for correctness by the authors. AI-assisted literature summaries were cross-referenced against original the original sources to ensure factual accuracy. AI-supported data analysis and result interpretations were verified against the raw data and critically challenged. All edited text were proofread and inspected to guarantee they preserve the intended scientific meaning and accurately represent the underlying data and ideas. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

## REFERENCES

Lukas Aichberger and Sepp Hochreiter. Unlocking the working memory of large language models for latent reasoning, 2026. URL https://arxiv.org/abs/2605.30343.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 20094–20136, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/578e65cdee35d00c708d4c64bce32971-Paper-Conference.pdf.

Jeffrey Cheng and Benjamin Van Durme. Compressed chain of thought: Efficient reasoning through dense representations. *ArXiv*, abs/2412.13171, 2024. URL https://api.semanticscholar.org/CorpusID:274789675.

Karl Cobbe, Vineet Kosaraju, Mo 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*, abs/2110.14168, 2021. URL https://api.semanticscholar.org/CorpusID:239998651.

Artur Back de Luca and Kimon Fountoulakis. Simulation of graph algorithms with looped transformers, 2024. URL https://arxiv.org/abs/2402.01107.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Ailin Deng, Tri Cao, Zhirui Chen, and Bryan Hooi. Words or vision: Do vision-language models have blind faith in text? In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3867–3876, 2025. doi: 10.1109/CVPR52734.2025.00366.

Lucio M. Dery, Benedict Aaron Tjandra, Siavash Samiei, Adhiguna Kuncoro, Zohar Yahav, Jiajun Shen, and Arthur Szlam. Skillsmith: Learning to compose parametric skills and textual knowledge, 2026. URL https://arxiv.org/abs/2607.27497.

Angeliki Giannou, Shashank Rajput, Jy-Yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of

*Proceedings of Machine Learning Research*, pp. 11398–11442. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/giannou23a.html.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ph04CRkPdC.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. Openthoughts: Data recipes for reasoning models, 2025. URL https://arxiv.org/abs/2506.04178.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason E Weston, and Yuandong Tian. Training large language model to reason in a continuous latent space, 2025. URL https://openreview.net/forum?id=tG4SgayTtk.

Jie He, Richard He Bai, Sinead Williamson, Jeff Z. Pan, Navdeep Jaitly, and Yizhe Zhang. Clara: Bridging retrieval and generation with continuous latent reasoning, 2026. URL https://arxiv.org/abs/2511.18659.

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. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=1ZuzFUMtx6.

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, 2021. URL https://arxiv.org/abs/2103.03874.

Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. Metagpt: Meta programming for a multi-agent collaborative framework, 2024. URL https://arxiv.org/abs/2308.00352.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Anna Kuzina, Maciej Pióro, and Babak Ehteshami Bejnordi. Kava: Latent reasoning via compressed KV-cache distillation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ePrhcLbtGv.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning, 2023. URL https://arxiv.org/abs/2307.13702.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models, 2022. URL https://openreview.net/forum?id=iedYJm92o0a.

OpenAI. Gpt-6 astra: A new generation of intelligence. https://openai.com/de-DE/index/gpt-6-astra/, September 2026. Accessed: 2026-09-23.

Alexander Panfilov, David Schmotz, Ilia Shumailov, Luca Beurer-Kellner, Joachim Schaeffer, Ameya Prabhu, Jonas Geiping, and Maksym Andriushchenko. Stealing reasoning traces from proprietary llm apis, 2026. URL https://arxiv.org/abs/2608.09867.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J. Reddi. Reasoning with latent thoughts: On the power of looped transformers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=din0lGfZFd.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

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 Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 677–693, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.36. URL https://aclanthology.org/2025.emnlp-main.36/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_VjQlMeSB_J.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=BAakY1hNKS.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. Softcot: Soft chain-of-thought for efficient reasoning with llms, 2025a. URL https://arxiv.org/abs/2502.12134.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. Softcot++: Test-time scaling with soft chain-of-thought reasoning, 2025b. URL https://arxiv.org/abs/2505.11484.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zhenrui Yue, Bowen Jin, Huimin Zeng, Honglei Zhuang, Zhen Qin, Jinsung Yoon, Lanyu Shang, Jiawei Han, and Dong Wang. Hybrid latent reasoning via reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=LjtgTpWH71.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: self-taught reasoner bootstrapping reasoning with reasoning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

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, 2025. URL https://arxiv.org/abs/2505.15778.

# A APPENDIX

## A.1 IMPLEMENTATION

**Frameworks** We use the model code and weights provided through Huggingface and train our models with TRL. For memory and compute efficient generation during training and evaluation, we use vllm.

Table 4: Experiment Hyperparameter Settings.

| | |
|---|---|
| Group size | 8 |
| Gradient Accumulation | Steps 128 |
| Total batch size | 256 |
| Maximum training steps | 100 |
| Optimizer | AdamW |
| Learning rate | $10^{-5}$ |
| Weight decay | 0.01 |
| Learning-rate scheduler | constant with warmup |
| Warmup steps | 10 |
| Precision | BF16-mixed |
| Maximum prompt / competion length | (2,048 / 8192) |
| Sampling temperature | 1.0 |
| Top-$p$ | 1.0 |
| Top-$k$ | $-1$ (disabled) |
| KL coefficient $\beta$ | 0 |
| Loss type | dr_grpo |
| Clipping parameter (Lower / upper) | 0.20 / 0.28 |
| Number of latent tokens | 16 |
| Margin weight | 0.002 |
| Margin target | 0.003 |
| Margin temperature | 0.1 |
| Softplus temperature | 0.1 |
| Dual learning rate | 0.20 |
| Number negatives / group | 2 |
| Token scope | last 256 tokens |

Table 5: Maimum Prompt and Completion length for evaluation tasks

| Dataset | Prompt / Completion length |
|---|---|
| GSM8k | 2048 / 16384 |
| Math500 | 2048 / 16384 |
| AIME24 | 2048 / 32768 |
| AIME25 | 2048 / 32768 |
| GPQA | 2048 / 16384 |

**Hyperparameter Sensitivity in CLUE.** We observe that an excessively large target encourages overly aggressive suppression of trajectory likelihood at the expense of answer correctness. Enforcing greater separation therefore does not necessarily improve task performance. The best-performing target margin varies across model configurations and training setups, suggesting that contrastive separation and outcome optimization need to be balanced based on the exact configuration.

**Evaluation settings.** For both training and evaluation, we build each prompt using the predefine prompt templates of the respecitve models. We prepend a system message that directs the LLM to perform step-by-step internal reasoning before generating its final answer. The user query is then appended, containing the question and optionally answer choices. In case of CLUE, we insert the latent representations directly after the assistant tag before the text generation. We report the system instructions and user-message templates used for evaluation below.

**Example Prompt for: AIME24, AIME25, MATH-500, GSM8k**

```
<system>
You are an expert logical and mathematical reasoning assistant.
For the problem given, think carefully through the steps
required to reach the solution. Put the final answer inside
\boxed{} at the end of your response.
</system>
<user>
{question}
</user>
<assistant>
```

**Example Prompt for: GPQA**

```
<system>
Solve the following multiple choice question by choosing one of
the options A, B, C, or D. Please reason step by step, and put
your final answer within \boxed{ANSWER} where ANSWER is one of
ABCD.
</system>
<user>
{question}
(A) {option_A}
(B) {option_B}
(C) {option_C}
(D) {option_D}
</user>
<assistant>
```

**Example Prompt for Formatting Latent Reasoning**

```
\begin{promptbox}{GPQA}
\begin{verbatim}
<system>
{system_prompt}
</system>
<user>
{question}
</user>
<assistant>
{latent_representations}
```

### A.2 Training Dynamics.

we examine the training dynamics of CLUE alongside the latent baselines in Figure 4. SoftCoT initially exhibits the steepest rise in mean reward but quickly converges and plateaus. This rapid early alignment is likely due to its small number of learnable parameters as both LLMs remain frozen, which initially speeds up alignment between the models but subsequently restricts the capacity for ongoing adaptation. Furthermore, HRPO consistently yields the lowest observed reward throughout training. HRPO relies on parameter-efficient fine-tuning via LoRA, severely limiting its ability to adapt to the training task.

In contrast, CLUE and CLUE w.o. contr. loss initially improve at a similar trajectory, but their reward gap widens noticeably between the 5k and 15k generation marks. During the latter half of this interval, CLUE's contrastive margin rises substantially. We hypothesize that this growing distinction contributes to CLUE's reward advantage: outcome-only optimization may favor broadly use-

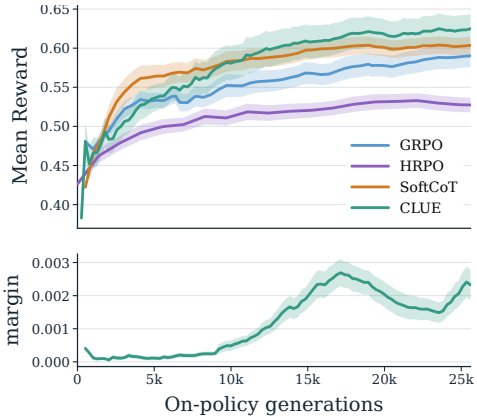


Figure 4: **Training dynamics.** CLUE develops a sustained reward advantage over the baselines as the contrastive likelihood margin emerges.

ful steering representations, whereas CLUE encourages input-specific utility through dense, token-likelihood-based supervision on correct trajectories.

### A.3 Additional Results

Table 6: For readability reasoning we report evaluation results with standard deviation in this following table. Results are reported as mean ± standard-deviation

| | ID | | | | OOD |
|---|---|---|---|---|---|
| | GSM8K | Math-500 | AIME24 | AIME25 | GPQA |
| | | | *Qwen3-1.7B* | | |
| zero-shot | 81.24 ± 0.52 | 70.62 ± 1.10 | 11.67 ± 4.40 | 10.42 ± 3.40 | 28.91 ± 2.30 |
| GRPO | 85.65 ± 0.68 | 80.10 ± 0.90 | 29.58 ± 4.52 | 20.41 ± 3.30 | 26.58 ± 1.66 |
| HRPO | 84.02 ± 0.99 | 74.80 ± 1.03 | 20.00 ± 5.02 | 13.33 ± 4.36 | 32.25 ± 3.26 |
| SoftCoT | 88.45 ± 0.62 | 76.28 ± 1.00 | 15.83 ± 4.63 | 14.58 ± 3.56 | **37.62** ± 2.07 |
| CLUE (ours) | **89.10** ± 0.39 | **84.68** ± 1.20 | **34.58** ± 4.36 | **27.08** ± 3.40 | 35.04 ± 1.80 |
| | | | *Qwen3-4B* | | |
| zero-shot | 90.55 ± 0.58 | 81.68 ± 1.07 | 43.33 ± 7.00 | 19.58 ± 3.54 | 40.71 ± 2.28 |
| GRPO | **92.87** ± 0.31 | **91.10** ± 0.41 | 55.83 ± 7.07 | **49.58** ± 8.05 | 45.95 ± 2.40 |
| HRPO | 91.52 ± 0.37 | 84.77 ± 0.63 | 43.33 ± 4.71 | 26.67 ± 5.18 | 44.44 ± 3.35 |
| SoftCoT | 91.40 ± 0.43 | 87.40 ± 1.95 | 26.67 ± 4.36 | 23.33 ± 2.36 | 41.91 ± 1.86 |
| CLUE (ours) | 92.57 ± 0.18 | 89.99 ± 0.58 | **56.25** ± 5.18 | 44.54 ± 4.69 | **48.67** ± 2.35 |