# NOT ALL SHORT CHAINS ARE EQUAL: HOW COT COMPRESSION AFFECTS SFT AND RL

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

## ABSTRACT

Chain-of-thought (CoT) compression has attracted growing interest as a way to reduce training and inference costs. However, compression ratio and accuracy may not fully reflect the value of compressed trajectories as training data. Similar compression ratios may lead to different outcomes in supervised fine-tuning (SFT) and subsequent reinforcement learning (RL). To investigate this, we conduct an end-to-end study on mathematical reasoning, comparing uncompressed data, prompt-based Chain-of-Draft compression, and self-distillation-based CRISP compression. On Qwen3-8B, SFT on Chain-of-Draft data yields 12.78 percentage points higher AIME accuracy than SFT on CRISP data. Its AIME advantage is concentrated on problems that the starting model can solve occasionally but inconsistently. In subsequent RL, Chain-of-Draft retains higher final accuracy than CRISP. These findings suggest that the value of short-CoT training data depends not only on response length, but also on how well it preserves the model's ability to solve problems it initially solves inconsistently. Since SFT also provides the initialization for RL, evaluating this training value should consider both immediate SFT outcomes and final performance after RL.

## 1 INTRODUCTION

Chain-of-thought (CoT) reasoning enables large language models (LLMs) to solve complex problems through intermediate reasoning steps. Beyond their role during inference, CoT also serves as training data for developing reasoning capabilities (Ho et al., 2023; Magister et al., 2023). OpenThoughts and Open-R1 supply reasoning trajectories for supervised fine-tuning (SFT) (Guha et al., 2026; Hugging Face, 2025), while reasoning-model training pipelines combine SFT with subsequent reinforcement learning (RL) (Guo et al., 2025). The training and inference costs of long CoT have motivated various compression methods (Aggarwal & Welleck, 2025; Zhang et al., 2025; Luo et al., 2026; Yang et al., 2025b; Han et al., 2025). However, these methods may retain different reasoning content and produce trajectories with different value for SFT (Yang et al., 2026) and subsequent RL. This raises the question: *what makes short CoT effective as training data?*

Chain-of-Draft (CoD) encourages brief reasoning steps through prompting (Xu et al., 2025), whereas CRISP learns concise reasoning through on-policy self-distillation (Sang et al., 2026). Compression is commonly evaluated through response length, answer accuracy, and inference efficiency. These metrics characterize the generated responses but do not fully establish their value as supervision: a short, correct response is not necessarily an effective training example. Similar average lengths may conceal differences in the reasoning content retained, with consequences for both SFT and subsequent RL.

We investigate this question through an end-to-end comparison of uncompressed trajectories (Raw), prompt-compressed trajectories generated with CoD, and self-distilled trajectories generated with CRISP. Using the same training protocol, we evaluate their effects on SFT and subsequent RL, focusing on mathematical reasoning with code generation as an additional evaluation task. We examine whether average trajectory length reflects training value, which problems account for performance differences, and whether these differences persist after RL.

Despite similar average trajectory lengths, CoD outperforms CRISP after SFT on Qwen3-8B by an average of **12.78** percentage points on AIME 2024-2026 (Zhang & Math-AI, 2024; 2025; 2026) and by **3.81** percentage points on LiveCodeBench (v6) (Jain et al., 2024). Grouping AIME problems by



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.

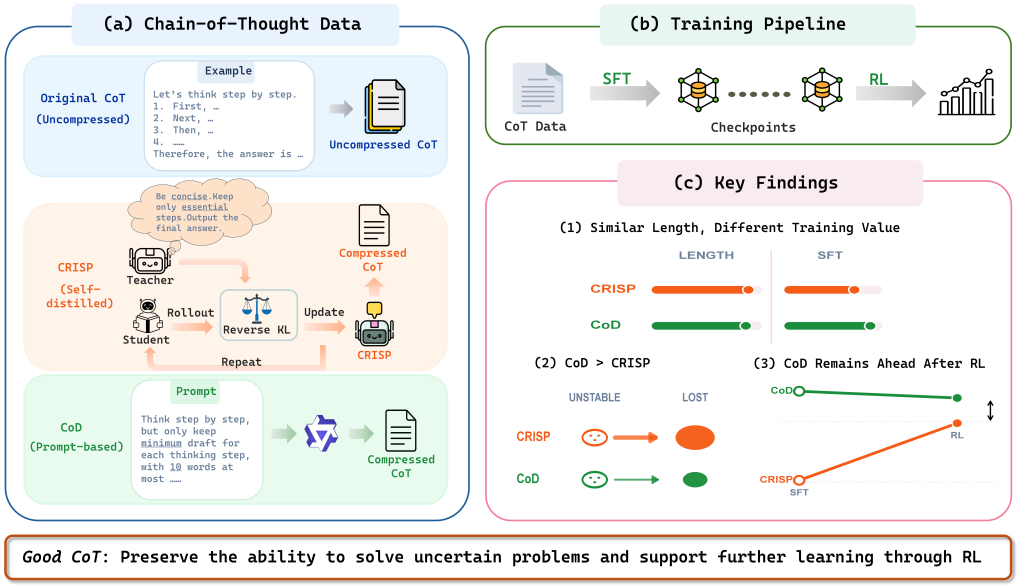



Figure 1: **Overview of the data construction, training pipeline, and main findings.** (a) Raw uses the original OpenR1 responses, while CoD and CRISP generate responses to a largely shared question pool using concise prompting and a self-distilled model, respectively. (b) Each dataset is used for SFT followed by RL. (c) On Qwen3-8B, CoD and CRISP exhibit different training value despite producing targets of similar average length: CoD yields stronger SFT performance, yields fewer mixed-to-0/8 transitions on initially inconsistently solved problems, and retains an advantage after RL. The mini-plots illustrate qualitative trends.

the base model's success across eight samples, we find that the advantage is concentrated on problems it solves occasionally but inconsistently. After SFT, fewer of these problems yield no correct answers across eight samples with CoD than with CRISP, revealing differences in the preservation of problem-solving ability beyond accuracy and trajectory length. Under a fixed RL training budget, the AIME gap narrows on Qwen3-8B but widens on Qwen3-4B. CoD retains higher final accuracy than CRISP on both models and across the tested Qwen3-8B SFT checkpoints. The differing RL gains highlight the need to consider both SFT performance and RL outcomes when assessing training value, as the effects of short-CoT data persist through subsequent training. The overall process of our study is shown in Figure 1.

Our contributions can be summarized as follows:

- We investigate what makes short CoT effective as training data through an end-to-end comparison of uncompressed, prompt-compressed, and self-distilled reasoning trajectories, tracking their effects through SFT and subsequent RL.
- We show that similar average training lengths can yield substantially different SFT outcomes, with CoD retaining higher post-RL accuracy than CRISP across the tested initializations.
- We identify preservation of the ability to solve uncertain problems as a key consideration in evaluating short-CoT training data. Problem-level analysis shows that CoD better preserves success on initially inconsistently solved problems, providing evidence for assessing compressed data through the capabilities they support beyond their length.

## 2 Related Work

**Chain-of-Thought Compression.** CoT compression aims to reduce training and inference costs. Chain-of-Draft prompts concise reasoning steps that retain essential information, without additional

training (Xu et al., 2025). TokenSkip prunes less informative tokens and fine-tunes models on the compressed data (Xia et al., 2025), while CoT-Valve controls reasoning length through parameter-space adjustments (Ma et al., 2025). RL-based approaches include O1-Pruner (Luo et al., 2026), ThinkPrune (Hou et al., 2025), and DAST (Shen et al., 2025). CRISP instead encourages concise reasoning through on-policy self-distillation (OPSD), minimizing token-level reverse KL divergence between the student and a concisely prompted teacher on student-generated trajectories (Sang et al., 2026). Our study evaluates the resulting trajectories as supervision, comparing their training value at similar compression levels.

**Reasoning Datasets.** OpenThoughts studies reasoning-data generation and curation across mathematics, coding, and science (Guha et al., 2026), while OpenR1-Math-220k provides mathematical problems with DeepSeek-R1-generated CoT solutions and verification information (Hugging Face, 2025). Other SFT resources include NuminaMath-CoT, AceReason-1.1-SFT, and Nemotron-Cascade (LI et al., 2024; Liu et al., 2026b; Wang et al., 2025); DAPO-Math-17k provides problems and reference answers for RL reward verification (Yu et al., 2025).

**Interactions Between SFT and RL.** The effectiveness of SFT also depends on how it prepares a model for subsequent RL. Kang et al. (2026) show that higher SFT accuracy does not reliably predict better post-RL performance and examine how data composition, including short-only versus variable-length supervision, affects this relationship. Zhang et al. (2026) attribute similar discrepancies to the mismatch between offline SFT data and online policy distributions, proposing PEAR to reweight the SFT loss through importance sampling. Other studies associate limited RL improvement after excessive SFT with reduced model plasticity (Liu et al., 2026a) and link checkpoint rank inversion in code-generation experiments to entropy collapse and diminished group-relative learning signals (Aphale & Liu, 2026). We examine how compression strategies affect both SFT and RL outcomes, using multiple SFT checkpoints to assess whether differences between the resulting datasets persist after RL.

# 3 Study Design and Experimental Setup

## 3.1 Training Data and Response Length

We compare three SFT datasets built from a shared question pool: self-distilled CRISP, prompt-compressed CoD, and uncompressed Raw.

**CRISP.** CRISP(Sang et al., 2026) learns concise generation through self-distillation, using a teacher conditioned on a conciseness instruction to supervise student-generated trajectories. We train Qwen3-8B with CRISP on DAPO-Math-17K (Yu et al., 2025), then generate responses to OpenR1-Math-220k questions and retain approximately 60,000 verifier-correct responses.

**CoD.** For each CRISP-selected question, we generate up to four Qwen3-8B responses using CoD (Xu et al., 2025). We select the shortest correct, non-truncated response, falling back to the shortest non-truncated response if none is correct. Questions with only truncated responses are excluded.

**Raw.** We retrieve the original OpenR1-Math-220k responses for the CRISP-selected questions as the uncompressed reference.

**Response length.** We directly tokenize the full response ($L_{\text{response}}$) and separately tokenize the text between the thinking tags ($L_{\text{think}}$) and after `</think>` ($L_{\text{post}}$), excluding the tags from both segments:

$$L_{\text{response}} = L_{\text{think}} + L_{\text{post}} + L_{\text{delim}}. \tag{1}$$

The residual $L_{\text{delim}}$ accounts for delimiters and whitespace/boundary tokenization differences; all counts use `add_special_tokens=False`.

Wei et al. (2022) define CoT as intermediate natural language reasoning steps leading to the final output. Our division into thinking and post-thinking segments follows the response format; the post-thinking segment may also contain derivations and explanations. Unless otherwise specified, response length refers to the directly tokenized full response defined above. Table 1 shows similar full-response lengths for CRISP and CoD, enabling comparison of training value at similar compression levels.

Table 1: **Length and token allocation of the SFT datasets.** Mean lengths describe untruncated responses in the SFT datasets, measured with the Qwen3-8B tokenizer without automatically added special tokens. Full-response and segment lengths follow Equation 1. Reduction is relative to Raw.

| Dataset | Full response | Think | Post-think | Reduction (%) |
|---|---|---|---|---|
| Raw | 5,303.4 | 4,777.7 | 521.7 | – |
| CRISP | 3,968.2 | 3,499.3 | 464.8 | 25.18 |
| CoD | 3,855.0 | 3,841.1 | 9.9 | 27.31 |

Despite similar total lengths, CoD contains more Think tokens and far fewer Post-think tokens than CRISP, highlighting differences in token allocation that average length alone does not capture.

### 3.2 Training and Evaluation

**Training.** Qwen3-8B is our primary model; Qwen3-4B (Yang et al., 2025a) and Olmo-3-7B-Think (Olmo et al., 2025) provide additional comparisons. Each model undergoes one epoch of full-parameter SFT per dataset using LLaMA-Factory (Zheng et al., 2024), with shared core hyperparameters within each model. Subsequent GRPO (Shao et al., 2024) in verl (Sheng et al., 2025) uses a shared set of 10,000 OpenR1-Math-220k questions (Hugging Face, 2025) excluded from SFT. Checkpoint selection is specified with each experiment; detailed configurations are in the appendix.

**Evaluation.** We evaluate mathematics on AIME 2024-2026 (90 problems), with an output limit of 30,000 tokens, and additionally evaluate code generation on LiveCodeBench (Jain et al., 2024) for selected models, with a limit of 32,768 tokens. For $K$ samples per problem, we report empirical Pass@1 as accuracy, averaging the within-problem correct fraction; Pass@$K$ is the fraction of problems with at least one correct sample. Detailed decoding and scoring settings are provided in the appendix.

### 3.3 Analysis Protocol

**Problem-level analysis.** To examine where performance differences arise, we group AIME problems by the starting model's success rate, using $K = 8$ responses per problem:

$$\begin{aligned} \mathcal{G}_0 &= \{i : \hat{p}_i(\theta_0) = 0\}, \\ \mathcal{G}_{\text{mix}} &= \{i : 0 < \hat{p}_i(\theta_0) < 1\}, \\ \mathcal{G}_1 &= \{i : \hat{p}_i(\theta_0) = 1\}. \end{aligned} \tag{2}$$

where $\theta_0$ is the starting model and $\hat{p}_i(\theta_0)$ is its fraction of correct responses among the $K$ samples for problem $i$. These groups correspond to problems answered incorrectly in all samples, correctly in some samples, and correctly in all samples, respectively.

We further examine how problems in the initial mixed group transition after SFT:

$$T_{\text{mix}\to c}(\theta) = \frac{1}{|\mathcal{G}_{\text{mix}}|} \sum_{i \in \mathcal{G}_{\text{mix}}} \mathbf{1}[s_i(\theta) = c], \qquad c \in \{0, \text{mix}, 8\}. \tag{3}$$

Here, $s_i(\theta) \in \{0, \text{mix}, 8\}$ records whether none, some, or all eight responses to problem $i$ are correct. The equation gives the fraction of initially mixed problems in each outcome after training; the three fractions sum to one.

## 4 Training Outcomes from SFT to RL

### 4.1 Similar Compression Ratios, Different SFT Outcomes

**Main results on mathematical reasoning.** Despite similar training-response lengths, CoD and CRISP yield substantially different SFT outcomes (Table 2). On Qwen3-8B, CoD achieves 64.86% AIME Pass@1, exceeding CRISP by 12.78 percentage points and Raw by 14.17 points.

Table 2: **Cross-model mathematical results after SFT and RL.** Pass@1 (%) on AIME24-26, with 30 problems per year and eight samples per problem; Avg. pools all 90 problems. RL runs for 156 steps from Base (Direct RL) or the final SFT checkpoint. Within each model and column, bold/underlined values mark the best/second-best results, excluding Base and Direct RL.

| Model | Data | Stage | AIME24 | AIME25 | AIME26 | Avg. |
|---|---|---|---|---|---|---|
| **Qwen3-8B** | Base | – | 75.42 | 67.08 | 64.17 | 68.89 |
| | Direct RL | RL | 70.42 | 65.00 | 67.08 | 67.50 |
| | Raw | SFT | 59.58 | 47.92 | 44.58 | 50.69 |
| | | RL | 63.75 | 45.42 | 50.00 | 53.06 |
| | CRISP | SFT | 63.33 | 45.00 | 47.92 | 52.08 |
| | | RL | **70.42** | 51.25 | 57.92 | 59.86 |
| | CoD | SFT | <u>70.00</u> | **62.08** | **62.50** | **64.86** |
| | | RL | 64.58 | <u>60.83</u> | <u>60.42</u> | <u>61.94</u> |
| **Qwen3-4B** | Base | – | 70.42 | 61.25 | 61.67 | 64.44 |
| | Direct RL | RL | 70.83 | 66.25 | 63.75 | 66.94 |
| | Raw | SFT | <u>57.50</u> | 42.08 | <u>47.50</u> | 49.03 |
| | | RL | 54.17 | 43.33 | 47.08 | 48.19 |
| | CRISP | SFT | 47.92 | 38.33 | 37.08 | 41.11 |
| | | RL | 49.58 | 42.08 | 42.08 | 44.58 |
| | CoD | SFT | 52.92 | <u>47.50</u> | 47.08 | <u>49.17</u> |
| | | RL | **62.92** | **55.00** | **51.25** | **56.39** |
| **Olmo-3-7B-Think** | Base | – | 70.83 | 64.17 | 69.17 | 68.06 |
| | Raw | SFT | <u>64.58</u> | <u>56.67</u> | <u>54.58</u> | <u>58.61</u> |
| | CRISP | SFT | 59.17 | 49.17 | 50.42 | 52.92 |
| | CoD | SFT | **67.50** | **57.08** | **55.83** | **60.14** |

*Note.* Standalone SFT/Base and RL evaluations are independently sampled. Olmo-3-7B-Think has no complete matched RL comparison, so only Base and SFT results are shown.

For Qwen3-8B, all three SFT models fall below the base model's 68.89% Pass@1. Given the strong reasoning performance of Qwen3-8B, this decline may reflect a mismatch between the training data and its existing reasoning behavior, as well as differences between the training and evaluation data's distributions. Relative to Raw-SFT, however, CoD substantially mitigates the decline, reducing the gap to the base model from 18.20 to 4.03 percentage points. CRISP provides only a small improvement over Raw despite achieving a similar average compression level to CoD. Thus, the degree of compression alone does not adequately characterize the downstream value of short-CoT supervision. During Qwen3-8B evaluation, CRISP produces the shortest responses, while CoD and Raw produce similar, longer responses (Table 9). Section 5 examines which problems may account for these differences.

Table 2 also reports results on Qwen3-4B and Olmo-3-7B-Think, extending the comparison across model scales and families. CoD outperforms CRISP by 8.06 and 7.22 percentage points on both models. However, its differences from Raw are smaller, at 0.14 and 1.53 points. The consistent finding across the three students is the advantage of CoD over CRISP, whereas a clear advantage over uncompressed data is observed only for Qwen3-8B.

**Transfer to code generation.** We additionally evaluate the mathematics-trained models on LiveCodeBench v6. For Qwen3-8B, CoD achieves 44.76% Pass@1 after SFT, compared with 40.95% for CRISP and 42.10% for Raw (Table 11). CoD leads on both tasks for Qwen3-8B, but code transfer is model-dependent: CoD underperforms CRISP on Qwen3-4B and scores similarly to it on Olmo-3-7B-Think.

### 4.2 Do the Differences Persist after RL?

**Performance after RL.** We train the SFT model with RL for one epoch, corresponding to 156 training steps in our setup. Starting from the final Qwen3-8B SFT checkpoints, CoD achieves 61.94%

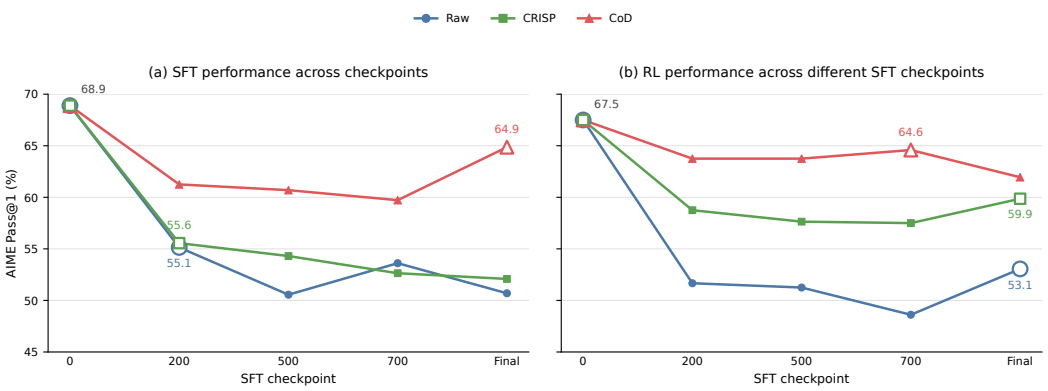



Figure 2: **SFT and RL performance across initialization checkpoints.** Qwen3-8B AIME24-26 Pass@1 (a) after SFT and (b) after 156 RL steps from each SFT checkpoint. Checkpoint 0 denotes Base in (a) and Direct RL in (b); Final is SFT step 916 for Raw/CRISP and 913 for CoD. SFT and RL evaluations are independently sampled.

Pass@1 on AIME, compared with 59.86% for CRISP and 53.06% for Raw (Table 2). The CoD-CRISP gap narrows from 12.78 percentage points after SFT to 2.08 points after RL, while CoD retains the highest endpoint accuracy among the three data conditions.

Direct RL from the base model reaches 67.50%, exceeding all three SFT-initialized models. Thus, CoD leads to a stronger final model than Raw or CRISP under this setup, although none of the three SFT approaches improves upon Direct RL.

**Results across SFT checkpoints.** We additionally initialize RL from SFT steps 200, 500, and 700 (Figure 2).

For CoD on Qwen3-8B, the final checkpoint has the highest SFT accuracy, whereas step 700 yields the highest post-RL accuracy among evaluated checkpoints. Nevertheless, CoD $>$ CRISP $>$ Raw holds after RL at every matched nonzero SFT depth. On Qwen3-4B, CoD also achieves higher post-RL accuracy than both CRISP and Raw (Table 2). These results distinguish checkpoint selection from the comparison between datasets; Section 5 discusses their implications for evaluating short-CoT supervision.

## 5 Analysis

We examine where the performance differences in Section 4 arise, how they relate to trajectory content and model outputs, and what RL reveals about training value. The main analysis focuses on Qwen3-8B on AIME; cross-model results and diagnostics appear in Table 2 and Appendices C and D.

### 5.1 Where Do the Performance Differences Arise?

Following Section 3.3, we group AIME problems by the base model's outcomes across eight sampled responses per problem: all correct (8/8), partly correct (mixed, 1-7/8), and none correct (0/8). These are sampled outcomes, not absolute solvability labels. Figure 3(a,b) further divides the mixed group into 4-7/8 and 1-3/8.

**The advantage concentrates on inconsistently solved problems.** Figure 3(a) shows that CoD outperforms CRISP by 21.25 and 19.32 percentage points in 4-7/8 and 1-3/8, whereas the difference is smaller on problems the base model consistently solves. One possible explanation is that the model, trained on CoD data, preserves the reasoning behavior needed to explore solutions when success is uncertain. This may help it reach correct answers more often on mixed problems, where it already demonstrates some ability to solve the task. This also makes the accuracy of mixed problems a possible indicator for evaluating the value of CoT data.

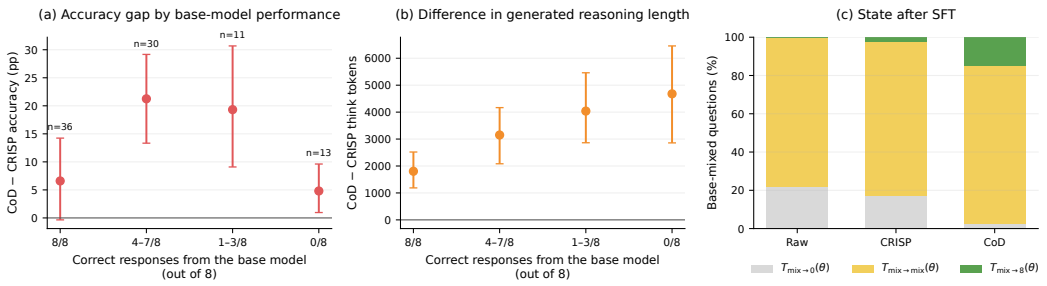



Figure 3: **Accuracy, reasoning length, and state transitions by base-model performance.** Groups reflect Base-model successes out of eight; $n$ is the group size. (a,b) Paired CoD-CRISP differences in Pass@1 (percentage points) and mean thinking tokens, with 95% question-bootstrap intervals. (c) Post-SFT fractions with 0, 1–7, or 8 correct samples among the 41 Base-mixed questions, following $T_{\text{mix}\to c}$ in Section 3.3.

**Generated reasoning provides supporting evidence.** Despite similar average training lengths, CoD generates longer thinking segments than CRISP, with larger differences on problems the base model solves less consistently (Figure 3(b)). This pattern is consistent with the interpretation that concise supervision can leave room for more extended solution exploration at evaluation time. In the mixed groups, longer reasoning accompanies higher accuracy. However, the largest length difference occurs in the 0/8 group, where the accuracy advantage remains small. Therefore, additional reasoning appears most useful where successful solutions are already occasionally attainable; length alone does not guarantee success.

**State transitions show that correct solutions remain accessible.** Figure 3(c) shows fewer mixed→0/8 transitions under CoD, reflected in a lower $T_{\text{mix}\to 0}$, alongside more mixed→8/8 transitions. Thus, CoD's advantage involves both fewer problems yielding entirely incorrect samples and more problems yielding consistently correct samples. Together with the length results, this suggests that CoD better preserves opportunities for successful reasoning on uncertain problems, rather than merely encouraging longer outputs. These observations support the exploration-based interpretation, although they do not directly measure the breadth or diversity of reasoning paths.

These findings suggest that effective short-CoT supervision should preserve successful reasoning on uncertain problems, beyond simply reducing trajectory length.

### 5.2 Why Do Similar Training Lengths Lead to Different Outcomes?

CoD and CRISP have similar average training lengths but different SFT accuracies (Tables 1 and 2). Their models also generate different response lengths (Table 9). We examine whether the parts of trajectories shortened help explain these differences in training value, providing insight into what makes short CoT effective as training supervision.

**Similar total lengths conceal different segment lengths.** Figure 4(a) decomposes training responses using the boundaries defined in Section 3.1. CoD contains more thinking tokens than CRISP on average (3,841 versus 3,499), but far fewer post-thinking tokens (10 versus 465). Thus, comparable total lengths do not mean comparable amounts of intermediate reasoning supervision. CoD combines a longer thinking segment with a more concise subsequent output, leaving more room for intermediate reasoning within the training response. Such examples may provide richer demonstrations of how to work through a problem, helping the trained model sustain reasoning when the solution is uncertain. This interpretation concerns the composition of the training responses; it does not assume that all post-thinking text is redundant.

Figure 5 illustrates the content difference: Raw and CRISP include a post-thinking explanation, whereas CoD ends with the answer immediately after the think block. CoD retains more think tokens despite a similar full-response length; this example illustrates token allocation rather than establishing a training effect.

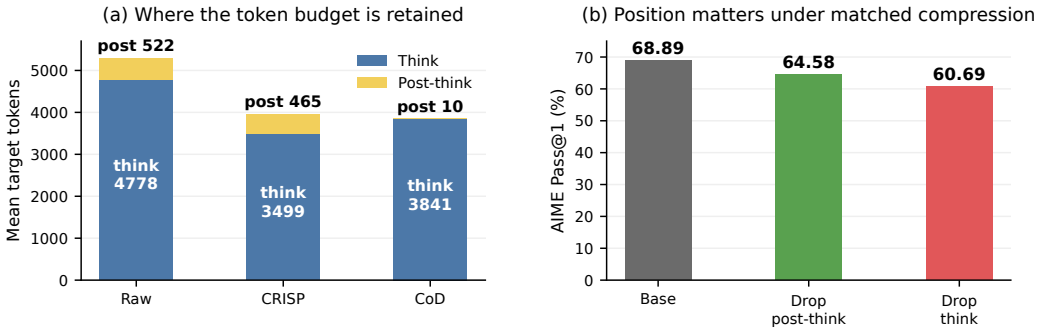



Figure 4: **Where compression removes tokens matters.** (a) Mean token counts between the thinking tags and after </think> across all examples actually used for SFT, using the Qwen3-8B tokenizer and the definitions in Section 3.1. (b) AIME 2024-2026 Pass@1 after one epoch of SFT on the same 4,096 Raw examples: deleting post-thinking exposition versus deleting a matched amount of thinking content. Both SFT runs start from the same Qwen3-8B checkpoint; Base is evaluated independently before our SFT or RL. Evaluation uses 90 problems with eight samples each.

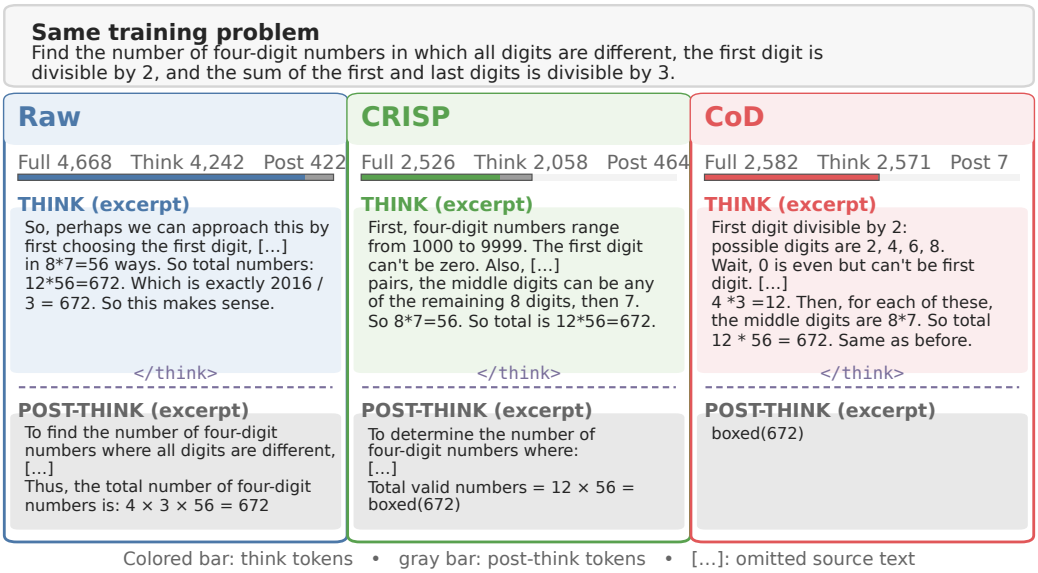



Figure 5: **Response content for the same training problem.** All three responses pass answer verification. Excerpts split at </think>; bracketed ellipses mark omissions. Counts cover unabridged responses and segments as defined in Section 3.1; bars show think and post-think counts on a shared scale.

**Matched deletion tests whether location matters.** To examine this explanation, we conduct a small controlled ablation using the same 4,096 mutually verified Raw training examples. One condition removes the entire post-thinking exposition while retaining the thinking segment and final answer; the other retains the exposition and final answer but removes contiguous, balanced blank-line blocks from the thinking segment. The deletion budget is 64–1,024 tokens and at most 25% of the original thinking length; the resulting targets are matched within 16 tokens for each training example. Both models start from the same Qwen3-8B base and undergo one epoch of SFT, with the final checkpoints selected without AIME feedback.

Figure 4(b) shows that removing post-thinking exposition yields 64.58% Pass@1, compared with 60.69% when removing thinking content—a difference of 3.89 percentage points (paired problem-bootstrap 95% CI: 0.97-6.94). With deletion amounts matched, the difference cannot be attributed

simply to one condition having shorter targets. Instead, it suggests that retaining the thinking segment provides more effective supervision than retaining an equivalent amount of post-thinking exposition under this intervention. This single-seed experiment supports the relevance of deletion location, although disrupting the coherence of thinking blocks may also contribute to the difference.

Together, these findings suggest that similarly short trajectories can differ in training value because they preserve different reasoning content. Effective short CoT should therefore balance conciseness with retaining the intermediate reasoning that supports subsequent problem solving.

### 5.3 How Does RL Change the Differences?

When SFT is followed by RL, evaluating CoT training data should consider both immediate SFT performance and the outcomes of subsequent RL. We therefore examine whether the differences observed after SFT persist under a fixed RL training budget. Motivated by this view, we conduct RL experiments to ask two questions: do the advantages of different CoT datasets persist after RL, and does the strongest SFT checkpoint provide the best initialization for RL in our experiments? These comparisons assess the downstream value of CoT supervision through the resulting model performance.

**CoT data affects performance beyond the SFT stage.** The CoD-CRISP accuracy gap narrows after RL on Qwen3-8B but widens on Qwen3-4B (Table 2). CoD nevertheless retains higher post-RL accuracy in both comparisons, and its advantage over Raw and CRISP holds across all evaluated Qwen3-8B SFT checkpoints (Figure 2). Thus, its stronger SFT performance is accompanied by stronger final performance after further optimization. Meanwhile, the different changes in the gaps show that RL does not affect all initializations equally. Evaluating CoT data only after SFT would miss how these differences develop during the subsequent training stage.

**The strongest SFT checkpoint isn't necessarily the best RL initialization.** Within the CoD condition, the final checkpoint achieves the highest SFT accuracy, whereas initialization from step 700 produces the highest accuracy after one epoch of RL (Figure 2). Selecting solely by SFT accuracy would therefore miss the best observed post-RL result. This illustrates why immediate task performance is an incomplete measure of an initialization's value: a checkpoint with lower SFT accuracy can still lead to a stronger model after further optimization.

Together, these results support assessing CoT data by both its immediate learning outcomes and the performance achieved through subsequent RL. CoD's advantage persists in our comparisons, but its best RL initialization cannot be identified from SFT accuracy alone. Effective CoT supervision should therefore provide both useful learning signals for SFT and a suitable starting point for continued training.

## 6 Limitations and Conclusion

**Limitations.** Our study focuses on mathematical reasoning, with code generation serving as an additional transfer evaluation, and covers a limited set of compression methods and model families. Our analysis of SFT training trajectories primarily examines response length and content structure. Semantic properties, such as whether an intermediate step is necessary, valid, or redundant, are harder to assess and quantify reliably. In addition, our RL experiments are not repeated across multiple training seeds, limiting our assessment of the robustness of post-RL performance differences and SFT-checkpoint rankings.

**Conclusion.** We studied compressed CoT as training data for SFT and subsequent RL. Despite similar average response lengths, Chain-of-Draft and CRISP produce substantially different training outcomes. Chain-of-Draft's SFT advantage is concentrated on problems that the original model solves inconsistently, with fewer such problems becoming unsuccessful across all sampled attempts. Our content analysis and controlled deletion experiment provide preliminary evidence that which parts of a response are retained matters for SFT performance, beyond the amount of compression. Subsequent RL does not eliminate the performance differences, and the best SFT checkpoint does not necessarily provide the best RL initialization. These findings suggest assessing short-CoT data by its preservation of success on uncertain problems and its effects on both SFT and RL.

### AI USE STATEMENT

We used generative AI tools to polish the language and improve readability, including descriptions of the experimental setup and some captions. We also used these tools for basic figure-presentation suggestions, such as using line plots to show changes across SFT checkpoints or RL training steps. The authors reviewed and revised the suggested wording and presentation choices, checked their consistency with the experimental results, and take full responsibility for the final manuscript, including its text, figures, results, and claims.

### REPRODUCIBILITY STATEMENT

To support reproducibility, we document the experimental pipeline and the settings used to obtain our results. Section 3 introduces the training-data variants, SFT-to-RL procedure, and evaluation metrics. Appendix A details data construction and filtering, training hyperparameters, and evaluation protocols. Appendix B provides training-data diagnostics and the controlled deletion-ablation setup. Appendices C, D, and E report detailed mathematical reasoning and code-generation results, including SFT checkpoint trajectories and fixed-budget RL comparisons. Appendix F specifies the samples, model checkpoints, token-level estimators, aggregation methods, and bootstrap procedures used for the NLL and KL diagnostics. These descriptions provide the information needed to reproduce our experimental procedures and assess the reported findings.

## REFERENCES

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.

Siddharth Aphale and Kelly Liu. Sft overtraining predicts rank inversion via entropy collapse under rlvr. *arXiv preprint arXiv:2606.18487*, 2026.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. In *International Conference on Learning Representations*, volume 2026, pp. 108059–108130, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware llm reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 24842–24855, 2025.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 14852–14882, 2023.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*, 2025.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL `https://github.com/huggingface/open-r1`.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint*, 2024.

Feiyang Kang, Michael Kuchnik, Karthik Padthe, Marin Vlastelica, Ruoxi Jia, Carole-Jean Wu, and Newsha Ardalani. Quagmires in sft-rl post-training: When high sft scores mislead and what to use instead. *The Fourteenth International Conference on Learning Representations (ICLR 2026)*, 2026.

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. https://huggingface.co/AI-MO/NuminaMath-CoT, 2024.

Runze Liu, Jiashun Liu, Xu Wan, Yuqian Fu, and Ling Pan. When rl fails after sft: Rejuvenating model plasticity for robust sft-to-rl handoff. *arXiv preprint arXiv:2606.09932*, 2026a.

Zihan Liu, Zhuolin Yang, Yang Chen, Chankyu Lee, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Acereason-nemotron 1.1: Advancing math and code reasoning through sft and rl synergy. In *International Conference on Learning Representations*, volume 2026, pp. 70010–70028, 2026b.

Haotian Luo, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Xiaochun Cao, Dacheng Tao, Naiqiang Tan, and Li Shen. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 14242–14257, 2026.

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.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1773–1781, 2023.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. CRISP: Compressed reasoning via iterative self-policy distillation. *arXiv preprint arXiv:2603.05433*, 2026. URL https://arxiv.org/abs/2603.05433v6. Version 6.

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 Saloni Potdar, Lina Rojas-Barahona, and Sebastien Montella (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 2322–2331, Suzhou (China), November 2025. Association for Computational Linguistics. ISBN 979-8-89176-333-3. doi: 10.18653/v1/2025.emnlp-industry.160. URL https://aclanthology.org/2025.emnlp-industry.160/.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Boxin Wang, Chankyu Lee, Nayeon Lee, Sheng-Chieh Lin, Wenliang Dai, Yang Chen, Yangyi Chen, Zhuolin Yang, Zihan Liu, Mohammad Shoeybi, et al. Nemotron-cascade: Scaling cascaded reinforcement learning for general-purpose reasoning models. *arXiv preprint arXiv:2512.13607*, 2025.

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.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. Tokenskip: Controllable chain-of-thought compression in llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3351–3363, 2025.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Junjie Yang, Ke Lin, and Xing Yu. Think when you need: Self-adaptive chain-of-thought learning. *arXiv preprint arXiv:2504.03234*, 2025b.

Yuming Yang, Mingyoung Lai, Wanxu Zhao, Xiaoran Fan, Zhiheng Xi, Mingqi Wu, Chiyue Huang, Jun Zhao, Haijun Lv, Jian Tong, et al. Which reasoning trajectories teach students to reason better? a simple metric of informative alignment. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42123–42150, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2025.

Dylan Zhang, Yufeng Xu, Haojin Wang, Qingzhi Chen, and Hao Peng. Good sft optimizes for sft, better sft prepares for reinforcement learning. *arXiv preprint arXiv:2602.01058*, 2026.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3716–3730, 2025.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2026, 2026.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 3: system demonstrations)*, pp. 400–410, 2024.

Table 3: **Dataset sizes.** Numbers of examples retained after data construction and their train–validation splits.

| Dataset | Retained pool | Training | Validation |
|---|---|---|---|
| CRISP | 59,101 | 58,601 | 500 |
| Raw | 59,101 | 58,601 | 500 |
| CoD | 58,877 | 58,377 | 500 |

# A Experimental Details

## A.1 Training-data Construction

**CRISP.** We follow CRISP (arXiv v6, formerly OPSDC) (Sang et al., 2026) to train a Qwen3-8B compression model through on-policy self-distillation. We first generate responses to 17,398 problems from DAPO-Math-17K (Yu et al., 2025) and retain 13,766 problems with verifier-approved responses, split into 11,012 training and 2,754 validation prompts. The student receives a standard problem-solving prompt, whereas the teacher receives an additional instruction to reason concisely. Training minimizes token-level reverse KL on student-generated trajectories, with the teacher updated from the student every 50 steps. We train for one epoch (343 steps), using a learning rate of $10^{-6}$, a global batch size of 32, and a maximum student rollout length of 8,192 tokens.

The resulting model generates responses to OpenR1-Math-220k problems (Hugging Face, 2025), using temperature 0.6, top-$p$ 0.95, top-$k$ 20, and a maximum output length of 30,000 tokens. Of 93,013 UUID-deduplicated records, 59,161 pass MathVerify; excluding 60 responses that reach the generation limit leaves 59,101 correct, non-truncated examples. We do not require an accepted response to be shorter than its Raw counterpart. Using seed 42, we divided the retained examples into 58,601 training examples and 500 validation examples.

**Raw.** For each problem selected for CRISP, we retrieve its original response from OpenR1-Math-220k using the corresponding record identifier. Raw inherits the same problem prompts, training–validation assignments, and ordering as CRISP. The two datasets therefore contain exactly the same problems within each split and differ in their response targets.

**CoD.** We apply Chain-of-Draft prompting (Xu et al., 2025) to the original Qwen3-8B model on the same 59,101 selected problems, generating up to four candidates per problem. The original CoD prompt requests at most five words per reasoning step. We relax this guideline to ten words to allow more room for expressing mathematical derivations while still encouraging concise reasoning.

Generation uses the same temperature, top-$p$, top-$k$, and output limit as CRISP data generation. For each problem, we select the shortest correct, non-truncated candidate when available; otherwise, we retain the shortest non-truncated candidate, even if it fails answer verification, to preserve problem coverage. Problems for which all candidates are truncated are excluded. This produces 49,482 verifier-approved targets and 9,395 fallback targets, with 224 problems excluded. The resulting 58,877 examples are divided into 58,377 training examples and 500 validation examples using seed 42. Table 3 provides an overview of the total number of datasets.

**Prompt examples.** The CRISP self-distillation prompts used in our implementation are shown below, with {`problem`} denoting the problem text.

**CRISP student prompt.**

```
Solve the following math problem step by step.  The last
line of your response should be of the form Answer:  $Answer
(without quotes) where $Answer is the answer to the problem.
{problem}
Remember to put your answer on its own line after "Answer:".
```

**CRISP teacher prompt.**

Table 4: **Supervised fine-tuning hyperparameters.**

**(a) Shared settings**

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| Fine-tuning | Full parameter | Epochs | 1 |
| Optimizer | AdamW (fused) | Precision | bf16 |
| Maximum sequence length | 16,384 | Per-device batch size | 1 |
| Effective global batch | 64 | LR scheduler | Cosine |
| Warmup ratio | 0.1 | Weight decay | 0 |
| Gradient clipping | 1.0 | Random seed | 42 |
| Distributed training | DeepSpeed ZeRO-3 | Save / evaluation interval | 100 steps |

**(b) Student-specific settings**

| Student | GPUs | Grad. acc. | LR | Template |
|---|---|---|---|---|
| Qwen3-8B | 8 | 8 | $10^{-6}$ | qwen3 |
| Qwen3-4B | 4 | 16 | $10^{-5}$ | qwen3 |
| Olmo-3-7B-Think | 8 | 8 | $10^{-6}$ | olmo3_think |

*Note.* Grad.acc. denotes gradient accumulation steps. CoD uses 224 fewer training examples, resulting in 913 steps rather than 916 under the same one-epoch schedule.

```
Solve the following math problem concisely and correctly.
Be direct | avoid unnecessary elaboration, redundant steps,
or restating the problem.  Focus only on the key reasoning
steps needed to reach the answer.

The last line of your response should be of the form Answer:
$Answer (without quotes) where $Answer is the answer to the
problem.

{problem}

Remember to put your answer on its own line after "Answer:".
```

**CoD compression prompt.**

```
Think step by step, but only keep a minimum draft for each
thinking step, with 10 words at most.  Return the answer at
the end of the response after a separator ####.
```

**Comparability across datasets.** All three datasets are constructed from a common selected problem pool, with exact split-level alignment between Raw and CRISP. CoD retains 99.62% of this pool, and its final training set is only 0.38% smaller. Within each student model, we use the same core SFT hyperparameters and one training epoch, resulting in 916 steps for Raw/CRISP and 913 for CoD. The datasets are therefore closely matched in size and training updates, although target correctness, candidate-selection procedures, and final split membership are not fully matched.

### A.2 Supervised Fine-tuning

We perform full-parameter SFT with LLaMA-Factory on Raw, CRISP, and CoD, using the same core hyperparameters within each student model. Table 4 summarizes the shared settings and model-specific configurations. After chat templating, prompt–response sequences exceeding 16,384 tokens are right-truncated rather than discarded. Under the Qwen3 template, this affects 1.02%, 1.44%, and 1.86% of the Raw, CRISP, and CoD training samples, respectively. Table 1 reports full-response lengths before SFT truncation. Final-SFT comparisons use the end-of-epoch checkpoint for each dataset.

Table 5: **Reinforcement-learning hyperparameters of Qwen3-8B and Qwen3-4B.** All formal comparisons use the common endpoint at RL step 156.

| Parameter | Setting |
|---|---|
| Algorithm | GRPO |
| Training data | 10,000 |
| Train batch / PPO mini-batch | 64 / 64 |
| PPO micro-batch per GPU | 1 |
| Rollouts per prompt | 8 |
| Maximum prompt / response | 2,048 / 16,384 tokens |
| Actor learning rate | 1e-6 |
| Actor KL loss type | low_var_kl |
| Actor KL coefficient | 0.001 |
| Entropy coefficient | 0 |
| Rollout backend | vLLM |
| GPU memory utilization | 0.75 |
| Training budget | 1 epoch, 156 optimizer steps |
| Evaluation frequency | 50 steps |

Table 6: **Evaluation protocol.**

| Setting | AIME24-26 | LiveCodeBench v6 |
|---|---|---|
| Tasks | 90 | 175 |
| Samples per task | 8 | 6 |
| Temperature | 0.6 | 0.6 |
| Top-p | 0.95 | 0.95 |
| Top-k | 20 | 20 |
| Maximum output tokens | 30,000 | 32,768 |

### A.3 Reinforcement Learning

We use GRPO implemented in verl, with configurations summarized in Table 5. All initializations receive the same ordered set of 10,000 OpenR1-Math problems excluded from SFT and the same training budget. For Qwen3-8B, we initialize RL from SFT checkpoints 200, 500, 700, and the final checkpoint, and include a baseline of RL trained directly from the original model as the baseline. All primary endpoint comparisons use RL step 156, rather than selecting the best RL checkpoint separately for each run.

### A.4 Evaluation Protocol

The evaluation settings are shown in Table 6.

**Mathematical reasoning.** We evaluate on AIME 2024, 2025, and 2026, with 30 problems per year and 90 problems in total. Each checkpoint generates eight responses per problem, using temperature 0.6, top-$p$ 0.95, top-$k$ 20, and a maximum output length of 30,000 tokens.

**Code generation.** Code is an additional evaluation task and is not used for SFT or RL training. The Qwen3-8B evaluations use OpenCompass on a 175-problem LiveCodeBench v6 evaluation

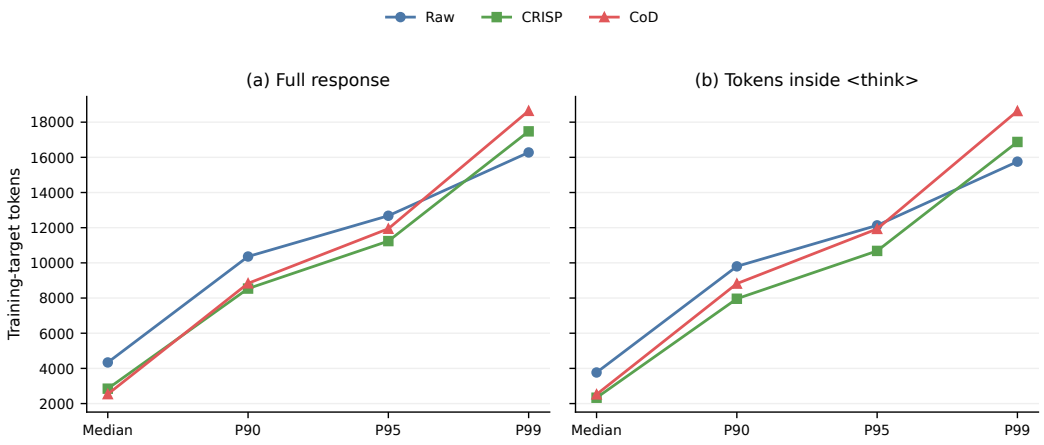



Figure 6: **Full-corpus target-length distributions.** The median and upper quantiles of full-response length (a) and tokens inside `<think>` (b) are computed over every example actually used for SFT: 58,601 Raw, 58,601 CRISP, and 58,377 CoD examples. Lengths use the Qwen3-8B tokenizer and the definitions in Section 3.1, without automatically added special tokens; the upper quantiles expose differences hidden by mean length alone.

set, with six responses per problem. Decoding uses temperature 0.6, top-$p$ 0.95, top-$k$ 20, and a maximum output length of 32,768 tokens. Reasoning text is removed before testing the final code, with an execution timeout of six seconds.

### A.5 Implementation and Reproducibility

Data splitting uses seed 42, and the CRISP generator is the final self-distillation checkpoint at step 343. We identify downstream checkpoints by their SFT initialization step and, where applicable, RL update step. Training-target length statistics use a common Qwen3-8B tokenizer across the three datasets, with `add_special_tokens=False`; full-response and segment counts follow Section 3.1. The principal runs use one training seed per configuration.

## B Training-Data Properties and Controlled Ablations

This section supplements the main-text analysis with training-target length distributions, surface-form and formatting diagnostics, and a controlled pilot comparing two deletion positions at nearly identical token budgets.

### B.1 Full Training-Length Distributions

Figure 6 reports the median, 90th, 95th, and 99th percentiles of full-response length and think-block length for each complete training corpus. Full-response length directly tokenizes the complete assistant response, including explicit thinking tags; think-block length separately tokenizes only the text between the tags. Both use `add_special_tokens=False`, as in Section 3.1. These statistics supplement the corpus averages by showing how compression affects both typical examples and the upper tail. The length comparisons refer to the three data variants under the same tokenization.

Both CRISP and CoD have shorter full responses than Raw at the median and the 90th and 95th percentiles. However, this ordering reverses at the 99th percentile, where both compressed datasets exceed Raw and CoD has the longest responses. A similar upper-tail pattern appears for think-block length. CoD also retains more think tokens than CRISP at the displayed percentiles. Thus, a lower average response length does not imply uniformly shorter targets: compression changes the length distribution, and the two compressed corpora retain different amounts of text within the think block.

Table 7: **Surface-form diagnostics of the complete training corpora.** Repetition is lexical and does not measure semantic redundancy; filler and backtracking are marker counts per 1,000 think tokens.

| Method | Rep. 4-gram | Rep. line | Filler | Backtrack | Think close | Boxed |
|---|---|---|---|---|---|---|
| Raw | 29.12 | 2.42 | 7.89 | 3.67 | 99.85 | 99.98 |
| CRISP | 29.24 | 1.38 | 12.91 | 4.68 | 100.00 | 99.97 |
| CoD | 28.97 | 1.31 | 12.46 | 4.68 | 100.00 | 100.00 |

Table 8: **Matched deletion-position ablation.** Both arms use 4,096 training examples, one training seed, and nearly identical deletion budgets. Pass@1 and Pass@8 are evaluated on AIME24-26 with 90 questions and eight generations per question. This pilot is supplementary and does not by itself establish a general causal mechanism.

| Variant | Deleted | Processed len. | Pass@1 | Pass@8 |
|---|---|---|---|---|
| Drop post-think | 495.02 | 5341.28 | 64.58 | 84.44 |
| Drop think | 494.93 | 5341.37 | 60.69 | 83.33 |

### B.2 Surface-form and Formatting Audits

Table 7 reports repeated 4-grams, repeated lines, filler and backtracking markers, think-block closure, and boxed-answer formatting across the complete training corpora. Filler and backtracking counts are normalized per 1,000 think tokens. The repetition and marker statistics describe lexical properties of the targets, while think-block closure and boxed-answer rates describe formatting. In particular, the presence of a boxed answer does not establish answer correctness.

Repeated 4-gram rates are similar across Raw, CRISP, and CoD (29.12%, 29.24%, and 28.97%, respectively). Both compressed datasets have lower repeated-line rates than Raw, but higher filler and backtracking marker frequencies. CRISP and CoD are especially close on these measures: their repeated-line rates are 1.38% and 1.31%, and both have 4.68 backtracking markers per 1,000 think tokens. Think-block closure and boxed-answer rates are near 100% for all three datasets. These diagnostics provide little separation between CRISP and CoD despite their different downstream training outcomes. They therefore serve as surface-form checks rather than direct measures of semantic redundancy, productive exploration, or reasoning quality.

### B.3 Matched Deletion-position Ablation

**Design.** To examine whether the location of removed text matters beyond the amount of compression, we conduct a paired pilot using 4,096 training examples derived from the same Raw responses. The *drop post-think* variant removes explanatory text after the think block while preserving the final answer. The *drop think* variant instead removes a similar number of tokens from within the think block, preserving the post-think text and final answer. Both variants are used to fine-tune the same Qwen3-8B starting model under matched training settings, with one training seed per variant. The original Qwen3-8B model before our SFT or RL, referred to as Base, is only a descriptive reference evaluated with separate samples; the controlled comparison is between the two deletion variants.

**Deletion budgets and evaluation.** Table 8 reports mean deletion amounts of 495.02 and 494.93 tokens for drop post-think and drop think, respectively. Their mean processed target lengths are 5,341.28 and 5,341.37 tokens, differing by only 0.09 tokens. Evaluation uses all 90 AIME 2024–2026 questions, with eight sampled responses per question, and reports Pass@1 and Pass@8.

**Results.** Drop post-think achieves 64.58% Pass@1, compared with 60.69% for drop think, a difference of 3.89 percentage points. The corresponding Pass@8 scores are 84.44% and 83.33%. Under nearly identical deletion budgets, the two interventions therefore lead to different SFT outcomes, providing preliminary evidence that where text is removed matters alongside how much is removed.

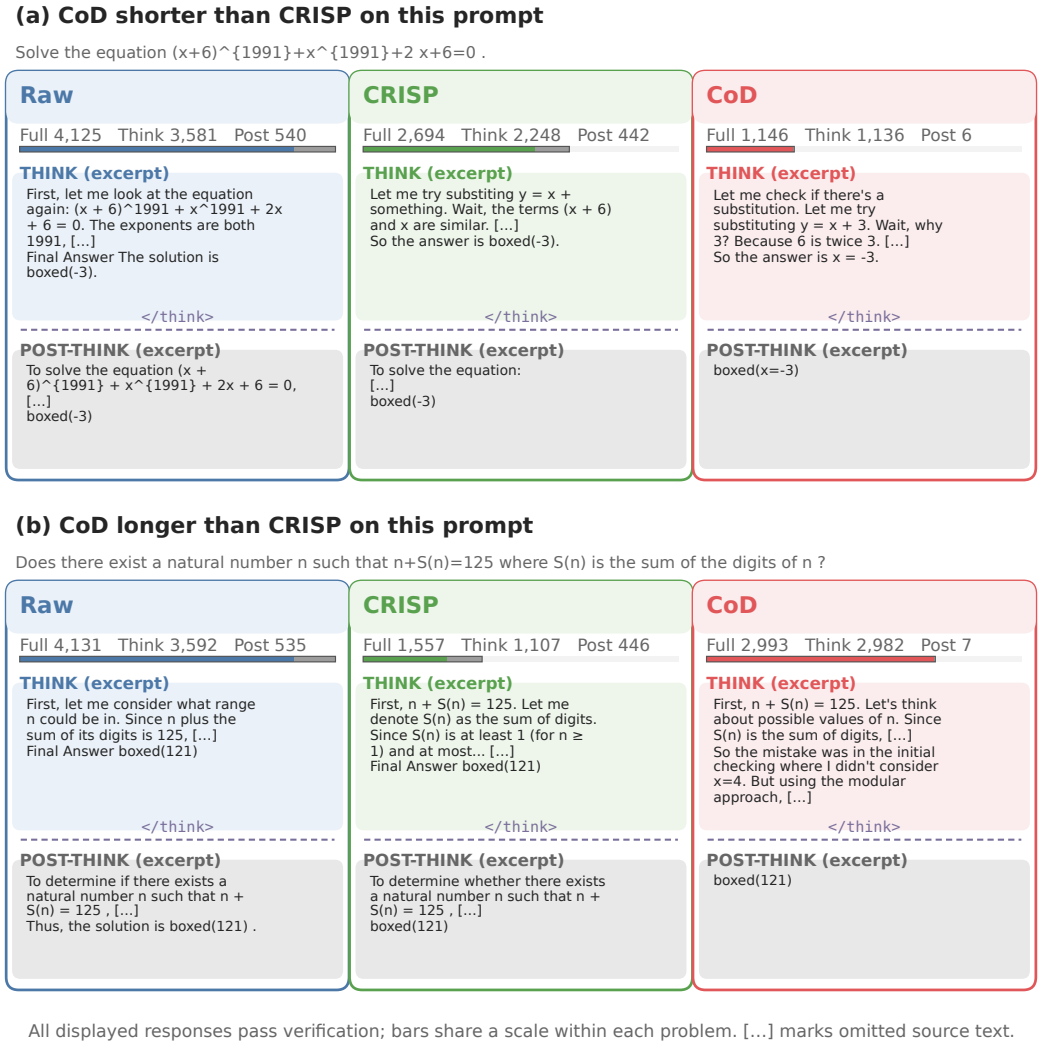



Figure 7: **Variation in response length on matched problems.** Raw, CRISP, and CoD responses all pass answer verification. CoD is shorter than CRISP in (a) and longer in (b). Token counts refer to the full, unabridged responses and their segments, following Section 3.1; bars share a scale within each problem. Bracketed ellipses mark omitted text.

This is a small-scale, single-seed deletion experiment on Raw responses; it does not reproduce the generative compression procedures used by CRISP and CoD or isolate the full explanation for their performance gap. Its evidence concerns SFT and does not directly establish an effect on subsequent RL.

### B.4 Additional Examples of Variation in Response Length

Figure 7 complements the main-text example in Figure 5. In (a), CoD uses 1,146 tokens compared with CRISP's 2,694; in (b), it uses 2,993 compared with 1,557. These examples illustrate variation across problems: the corpus-level mean ordering need not hold for an individual response, and greater length alone does not establish lower training value.

Table 9: **Cross-model standalone SFT diagnostics on AIME24-26.** All systems are evaluated on 90 questions with eight generations per question. Accuracy, state proportions, and truncation rates are percentages; mean lengths are in tokens. Answer entropy measures dispersion of final answer strings, not reasoning-route diversity.

| Model | Method | Pass@1 | Pass@8 | 0/8 | 8/8 | Mixed | Mean len. | Entropy | Trunc. |
|---|---|---|---|---|---|---|---|---|---|
| Qwen3-8B | Base | 68.89 | 85.56 | 14.44 | 40.00 | 45.56 | 15,672 | 0.707 | 11.25 |
| | Raw | 50.69 | 77.78 | 22.22 | 16.67 | 61.11 | 12,308 | 1.451 | 3.33 |
| | CRISP | 52.08 | 77.78 | 22.22 | 15.56 | 62.22 | 10,208 | 1.204 | 5.69 |
| | CoD | 64.86 | 88.89 | 11.11 | 27.78 | 61.11 | 12,618 | 0.906 | 7.50 |
| Qwen3-4B | Base | 64.44 | 78.89 | 21.11 | 38.89 | 40.00 | 14,956 | 0.800 | 9.03 |
| | Raw | 49.03 | 75.56 | 24.44 | 21.11 | 54.44 | 13,083 | 1.437 | 6.25 |
| | CRISP | 41.11 | 75.56 | 24.44 | 12.22 | 63.33 | 13,415 | 1.470 | 21.11 |
| | CoD | 49.17 | 76.67 | 23.33 | 13.33 | 63.33 | 14,006 | 1.228 | 17.78 |
| Olmo-3-7B-Think | Base | 68.06 | 86.67 | 13.33 | 37.78 | 48.89 | 16,954 | 0.696 | 14.44 |
| | Raw | 58.61 | 84.44 | 15.56 | 31.11 | 53.33 | 14,341 | 1.007 | 5.97 |
| | CRISP | 52.92 | 77.78 | 22.22 | 25.56 | 52.22 | 13,630 | 1.269 | 5.83 |
| | CoD | 60.14 | 81.11 | 18.89 | 31.11 | 50.00 | 15,422 | 1.012 | 8.06 |

# C Detailed Cross-model Math Results

This section provides sampling diagnostics, SFT checkpoint trajectories, and problem-level analyses for Qwen3-8B, Qwen3-4B, and Olmo-3-7B-Think. All results use AIME 2024-2026, comprising 90 problems with eight sampled responses per problem. Base denotes the original model before our SFT or RL, and final SFT results use the final training checkpoint rather than the checkpoint with the highest evaluation accuracy.

## C.1 Full Sampling Diagnostics

Table 9 supplements Pass@1 with Pass@8, the proportions of problems in the 0/8, mixed (1–7/8), and 8/8 states, mean output length, answer entropy, and truncation rate. The state proportions summarize the same eight samples used for Pass@8: a problem is in 0/8 if none of its sampled responses is correct. Answer entropy describes the dispersion of final answer strings and is not a measure of reasoning-route diversity.

Relative to CRISP, CoD has higher Pass@8 and fewer 0/8 problems for all three models. The reduction in 0/8 is largest for Qwen3-8B, from 22.22% to 11.11%, and smaller for Qwen3-4B and Olmo-3-7B-Think. CoD also has lower answer entropy than CRISP across all three models. However, higher Pass@1 does not imply a uniformly better sampling profile: for Olmo-3-7B-Think, CoD has higher Pass@1 than Raw but lower Pass@8 (81.11% versus 84.44%).

CoD produces longer complete responses than CRISP within each model, while their truncation-rate ordering varies across models. Consequently, neither output length nor truncation rate alone accounts for the accuracy ordering. Length comparisons are made within each student model; absolute token counts across different tokenizers are not directly comparable.

## C.2 SFT Checkpoint Trajectories

Figure 8 shows AIME Pass@1 at intermediate and final SFT checkpoints, with the corresponding Base score as a horizontal reference. For Qwen3-8B, CoD remains above both Raw and CRISP throughout the displayed checkpoints. For Qwen3-4B and Olmo-3-7B-Think, CoD also remains above CRISP across the displayed trajectory, whereas its ordering relative to Raw changes during

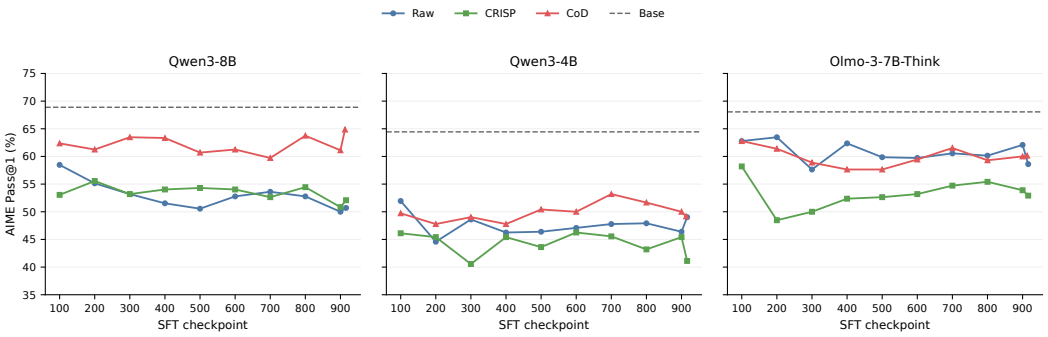



Figure 8: **Standalone SFT trajectories across student models.** AIME24-26 Pass@1 is shown across saved SFT checkpoints for Qwen3-8B, Qwen3-4B, and Olmo-3-7B-Think; dashed lines mark each Base model. Each point uses 90 questions with eight samples per question. The trajectories show that performance need not improve monotonically with additional SFT updates.

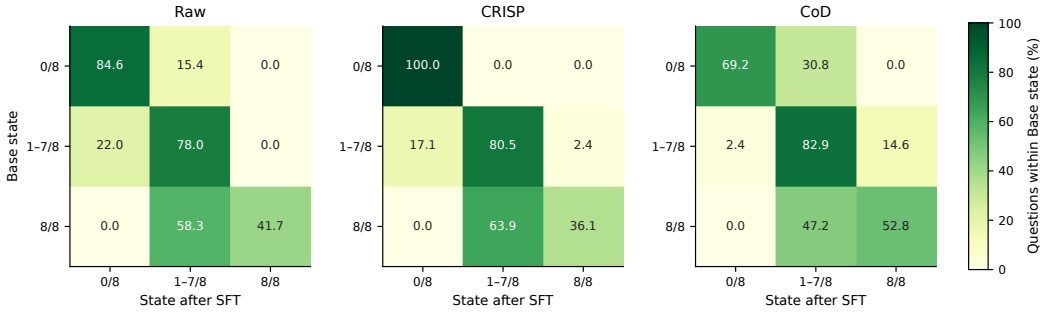



Figure 9: **Complete base-to-SFT state transitions.** For each Qwen3-8B training corpus, rows give the Base state and columns the final SFT state, where 0/8 means always incorrect, 1–7/8 means mixed, and 8/8 means always correct across eight samples. Each cell is the conditional percentage of questions within its Base-state row, so every row sums to 100%.

training. The separation between CoD and CRISP thus appears before the final checkpoint and persists through the observed training process.

The trajectories are not monotonic, and an intermediate checkpoint can outperform the final one. We retain the final-checkpoint comparison to avoid selecting a different evaluation peak for each method. These curves describe within-run training dynamics; they do not measure variability across independent training seeds.

### C.3 Complete Base-to-SFT State Transitions

Figure 9 reports the complete Qwen3-8B transition matrices from Base to each final SFT model. Rows indicate the Base state and columns indicate the state after SFT; each row is normalized by the number of problems in that Base group. This expands the main-text analysis of initially inconsistent problems to include the initially unsolved and consistently solved groups.

Among the 41 Base-mixed problems, 22.0% become 0/8 after Raw SFT and 17.1% after CRISP SFT, compared with 2.4% after CoD SFT. Conversely, the proportions becoming 8/8 are 0.0%, 2.4%, and 14.6%, respectively. CoD also retains more of the Base-8/8 problems in the 8/8 state (52.8%, versus 41.7% for Raw and 36.1% for CRISP). Among Base-0/8 problems, 30.8% become mixed under CoD, compared with 15.4% under Raw and 0.0% under CRISP; none become 8/8. These are changes in observed finite-sample states, rather than definitive acquisition or loss of problem-solving ability.

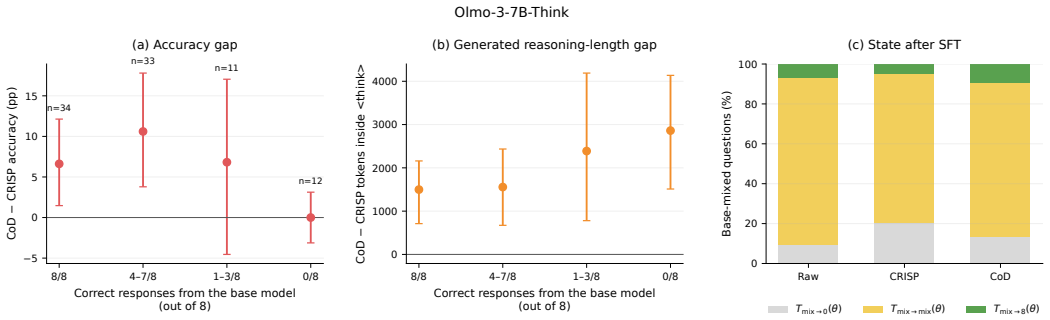



Figure 10: **Frontier analysis on Olmo-3-7B-Think.** Questions are grouped by the Base model's correct responses out of eight. Panels (a) and (b) show paired CoD–CRISP differences in Pass@1 percentage points and mean <think> tokens, with 95% question-bootstrap intervals. Panel (c) reports $T_{\text{mix}\to c}(\theta)$, the fraction of Base-mixed questions whose SFT state $s_i(\theta)$ becomes $c \in \{0, \text{mix}, 8\}$.

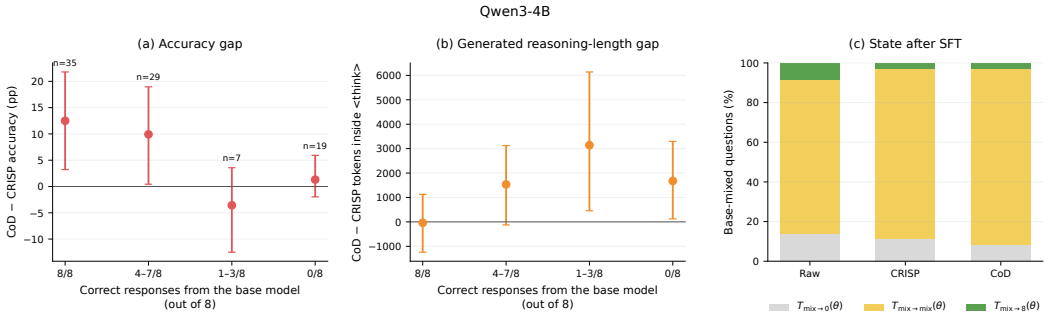



Figure 11: **Frontier analysis on Qwen3-4B.** Questions are grouped by the Base model's correct responses out of eight. Panels (a) and (b) show paired CoD–CRISP differences in Pass@1 percentage points and mean <think> tokens, with 95% question-bootstrap intervals. In (c), $s_i(\theta) \in \{0, \text{mix}, 8\}$ denotes 0, 1–7, or 8 correct samples, and each segment is $T_{\text{mix}\to c}(\theta) = \Pr[s_i(\theta) = c \mid s_i(\theta_0) = \text{mix}]$.

### C.4 Cross-model Reasoning-frontier Analyses

Figure 10 and Figure 11 extend the grouped analysis to Qwen3-4B and Olmo-3-7B-Think. For each model, problems are grouped by that model's own Base correct-response count: 8/8, 4–7/8, 1–3/8, and 0/8. The figure compares CoD and CRISP in accuracy and generated think-block length, and shows the post-SFT states of initially mixed problems. The group membership is model-specific, so the panels do not represent identical difficulty partitions between models.

For Olmo-3-7B-Think, CoD has positive accuracy gaps in the 8/8 and both mixed subgroups, with the largest point estimate in the 4–7/8 group and essentially no gap in the 0/8 group. It also generates more think tokens in all four groups. For Qwen3-4B, the pattern differs: the accuracy advantage is largest in the 8/8 group and remains positive in the 4–7/8 group, but reverses in the 1–3/8 group, which contains only seven problems. CoD generates more think tokens in that subgroup despite its lower accuracy. Longer reasoning therefore does not uniformly translate into better performance.

In both models, fewer initially mixed problems become 0/8 under CoD than under CRISP, although the difference is smaller for Qwen3-4B than for Qwen3-8B. Taken together, these panels extend the CoD–CRISP comparison beyond the primary model while showing that the location and size of its benefits vary across students and problem groups.

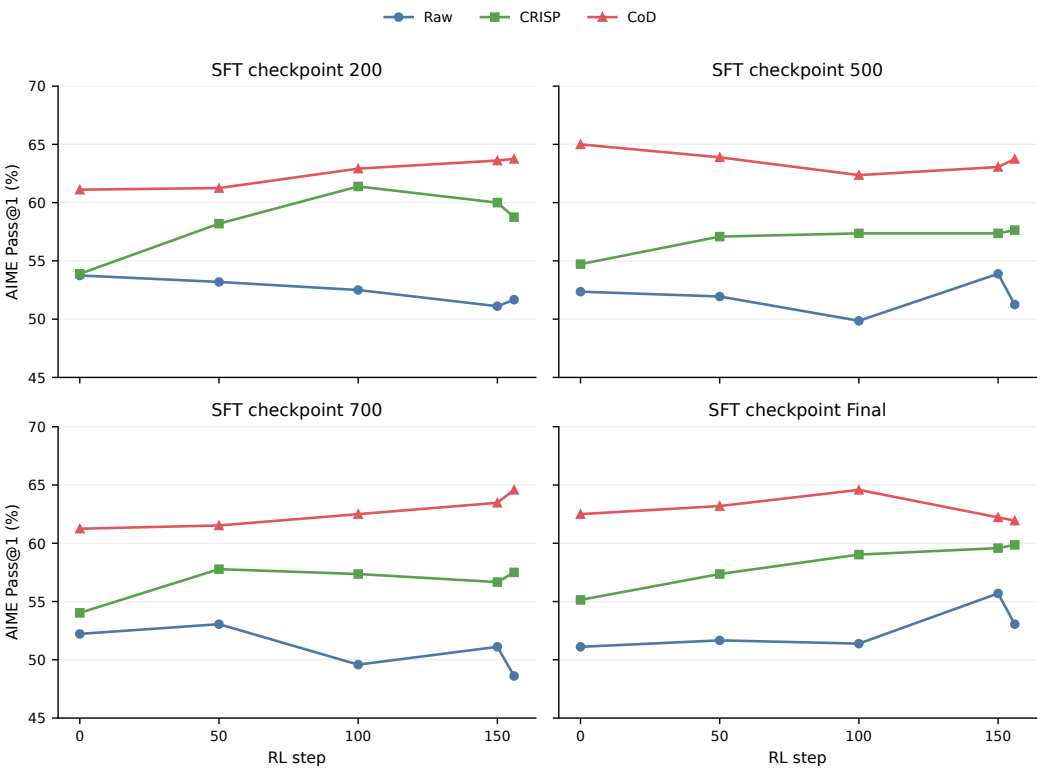



Figure 12: **RL evaluation trajectories from different SFT checkpoints.** Each panel fixes a Qwen3-8B SFT initialization checkpoint and follows AIME24–26 Pass@1 through RL steps 0, 50, 100, 150, and 156. Every evaluation contains 90 questions with eight samples per question. Final corresponds to SFT step 916 for Raw/CRISP and 913 for CoD.

# D Additional RL Results and Training Dynamics

This section supplements the main-text RL results with evaluation trajectories, training dynamics, complete fixed-endpoint diagnostics, and accuracy–length comparisons. The trajectory and length analyses concern Qwen3-8B, while the endpoint table additionally includes Qwen3-4B. We distinguish the improvement achieved during RL from the final performance reached under a fixed RL budget, since these provide different information about an SFT initialization.

## D.1 Evaluation Trajectories during RL

Figure 12 shows independent AIME evaluations during RL for Qwen3-8B initialized from SFT checkpoints 200, 500, 700, and the final checkpoint. Each panel compares Raw, CRISP, and CoD at a matched SFT depth, with the final checkpoint corresponding to step 916 for Raw and CRISP and step 913 for CoD. Comparisons use the common RL endpoint at step 156 rather than selecting each run's highest-scoring checkpoint. RL step 0 is evaluated separately from the standalone SFT results, so the two sets of sampled scores need not coincide exactly.

CoD remains above CRISP and Raw at all displayed RL evaluation points across the four initialization depths. However, the trajectories differ: CRISP improves from its starting point in each panel, whereas CoD's changes are smaller and depend on the SFT checkpoint. For the final SFT initialization, CoD improves early in RL and subsequently declines, while CRISP ends above its starting score. The resulting gap narrows, but CoD retains the higher endpoint accuracy. Thus, a larger gain during RL does not necessarily imply a better final model, and the initialization gap is not eliminated within the observed training budget.

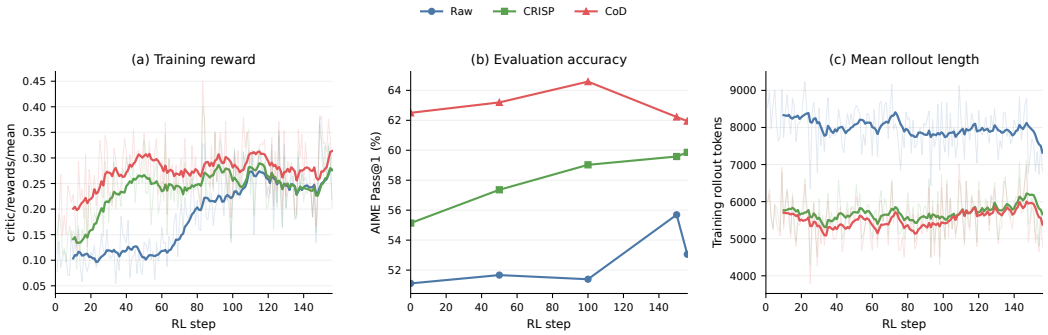



Figure 13: **Training reward, evaluation accuracy, and rollout length during RL.** Curves compare Qwen3-8B runs initialized from the final Raw, CRISP, and CoD SFT checkpoints. Panels (a) and (c) show per-step training reward and mean rollout length in light colors, with 10-step moving averages in dark colors; (b) shows independent AIME24–26 Pass@1 evaluations on 90 questions with eight samples each. Training reward is measured on training rollouts and is distinct from evaluation accuracy.

Table 10: **Cross-model results at the fixed RL endpoint.** All values are percentages. All entries use RL step 156 on AIME24-26 (90 questions, eight samples per question). Pass@8 is the fraction of questions solved at least once; 0/8 is its complement. The 0/8, mixed, and 8/8 columns partition questions by whether zero, one to seven, or all eight samples are correct.

| Model | Initialization | Pass@1 | Pass@8 | 0/8 | Mixed | 8/8 |
|---|---|---|---|---|---|---|
| Qwen3-8B | Direct RL | 67.50 | 82.22 | 17.78 | 42.22 | 40.00 |
| | Raw | 53.06 | 77.78 | 22.22 | 64.44 | 13.33 |
| | CRISP | 59.86 | 77.78 | 22.22 | 48.89 | 28.89 |
| | CoD | 61.94 | 81.11 | 18.89 | 55.56 | 25.56 |
| Qwen3-4B | Direct RL | 66.94 | 82.22 | 17.78 | 41.11 | 41.11 |
| | Raw | 48.19 | 75.56 | 24.44 | 53.33 | 22.22 |
| | CRISP | 44.58 | 76.67 | 23.33 | 67.78 | 8.89 |
| | CoD | 56.39 | 81.11 | 18.89 | 58.89 | 22.22 |

### D.2 RL Training Dynamics

Figure 13 presents training reward, independent AIME accuracy, and mean training rollout length for Qwen3-8B initialized from the final SFT checkpoints. Training reward is measured on the RL training prompts, whereas AIME accuracy is evaluated independently. Their trajectories therefore need not change together or remain monotonic.

Raw begins with lower training reward and rises substantially during the run, approaching the compressed-data initializations later in training. This convergence in training reward does not eliminate its lower AIME accuracy. Meanwhile, Raw produces substantially longer training rollouts than either CRISP or CoD throughout the displayed period. The two compressed-data initializations have broadly similar training rollout lengths despite their different AIME performance. These observations motivate reporting independent evaluation alongside reward and rollout length: neither training reward nor the amount of generated training text alone captures the downstream value of the initialization.

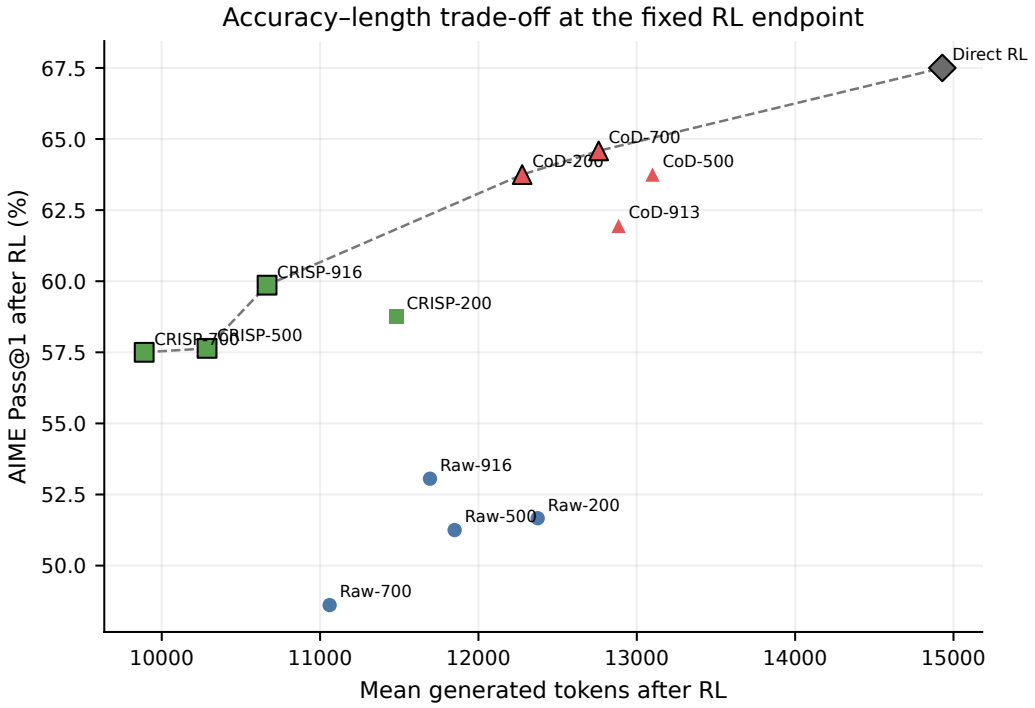



Figure 14: **Accuracy–length trade-off after fixed-budget RL.** Each point is a Qwen3-8B initialization evaluated at RL step 156, with mean generated tokens on the horizontal axis and AIME24–26 Pass@1 on the vertical axis. Black-outlined points and the dashed curve form the observed Pareto frontier: no evaluated point is both shorter and more accurate. Labels identify the SFT corpus and checkpoint; Direct RL starts from the Base model.

### D.3 Fixed-endpoint Results

Table 10 reports results at RL step 156 for Qwen3-8B and Qwen3-4B initialized from the final SFT checkpoints, together with Direct RL from the original model. Evaluation uses 90 AIME 2024–2026 problems with eight sampled responses per problem. In addition to Pass@1 and Pass@8, the table reports the proportions of problems with zero, one to seven, or eight correct responses. These are endpoint sampling diagnostics rather than evidence of stability across training seeds.

Among the three SFT-based initializations, CoD achieves the highest Pass@1 and Pass@8 for both models. For Qwen3-8B, its Pass@1 is 61.94%, compared with 59.86% for CRISP and 53.06% for Raw. For Qwen3-4B, the corresponding scores are 56.39%, 44.58%, and 48.19%. CoD also has the lowest 0/8 proportion among the SFT-based variants in both models. The full state distribution adds nuance: Qwen3-8B CRISP has a larger 8/8 proportion than CoD (28.89% versus 25.56%), despite its lower overall Pass@1 and Pass@8. No single state proportion fully characterizes the performance difference.

Direct RL achieves higher Pass@1 than all three SFT-based initializations for both models, at 67.50% for Qwen3-8B and 66.94% for Qwen3-4B. The comparison therefore supports CoD's stronger endpoint among the evaluated SFT data variants, without establishing that compressed-data SFT improves on directly applying RL to the original model.

### D.4 Accuracy–Length Trade-offs

Figure 14 compares Qwen3-8B endpoint accuracy with mean generated response length on the evaluation set. Each point uses RL step 156, while its suffix identifies the preceding SFT checkpoint. These evaluation lengths are distinct from the training rollout lengths in Figure 13.

Table 11: **Cross-model transfer to code generation.** Pass@1 (%) on 175 LiveCodeBench tasks with six generations per task. Base denotes the original model before our SFT or RL; its score is listed in the SFT column for reference.

| Method | SFT | RL |
|---|---|---|
| **Qwen3-8B** | | |
| Base | 46.48 | – |
| Raw | 42.10 | 42.86 |
| CRISP | 40.95 | 43.33 |
| CoD | 44.76 | 45.14 |
| **Qwen3-4B** | | |
| Base | 46.00 | – |
| Raw | 36.38 | 35.43 |
| CRISP | 34.19 | 35.33 |
| CoD | 27.05 | 23.81 |
| **Olmo-3-7B-Think** | | |
| Base | 44.10 | – |
| Raw | 41.43 | – |
| CRISP | 37.62 | – |
| CoD | 37.90 | – |

*Note.* Qwen3-4B SFT uses a learning rate of $1 \times 10^{-5}$; its RL runs start from these SFT models. Only SFT results are available for Olmo-3-7B-Think. Dashes indicate unavailable or inapplicable entries.

The observed endpoints occupy different parts of the accuracy–length trade-off. CRISP generally produces shorter evaluation responses with lower accuracy, while CoD reaches higher accuracy at greater output lengths. Direct RL attains the highest accuracy and also generates the longest responses among the plotted settings. All displayed Raw endpoints are dominated by at least one CRISP endpoint that is both shorter and more accurate. Within CoD, later SFT does not uniformly improve the trade-off: the final-SFT initialization is less accurate and produces longer responses than the checkpoint-200 initialization after RL. The plotted frontier is descriptive of these evaluated configurations; it is not a claim about an optimal allocation of training or inference compute.

## E Code Generalization Results

We report LiveCodeBench results for Qwen3-8B, Qwen3-4B, and Olmo-3-7B-Think in Table 11. Code generation is used exclusively for evaluation; no code data are used in SFT or RL. The SFT results evaluate model checkpoints obtained by fine-tuning on each of the three mathematical CoT datasets: Raw, CRISP, and CoD. The RL results evaluate checkpoints obtained by further mathematical RL starting from the corresponding SFT models. Thus, these results assess whether the effects of mathematical CoT supervision and subsequent RL transfer to code generation. The evaluation protocol is described in Appendix A.4.

**After SFT.** For Qwen3-8B, CoD achieves 44.76% Pass@1, exceeding CRISP at 40.95% and Raw at 42.10%, while remaining below Base at 46.48%. For Qwen3-4B, Raw achieves the highest Pass@1 among the three SFT variants at 36.38%, followed by CRISP at 34.19% and CoD at 27.05%. CoD therefore performs worse than CRISP on code despite its higher mathematical reasoning accuracy. For Olmo-3-7B-Think, CoD and CRISP obtain similar scores, at 37.90% and 37.62%, respectively, with only a 0.28-percentage-point numerical difference; both remain below

Raw at 41.43%. All SFT variants score below the corresponding original models, whose Pass@1 scores are 46.00% for Qwen3-4B and 44.10% for Olmo-3-7B-Think.

**After RL.** For Qwen3-8B, CoD retains the highest score among the SFT-based variants at 45.14%, followed by CRISP at 43.33% and Raw at 42.86%; all remain below Base at 46.48%. The available Qwen3-4B RL results preserve the ordering Raw $>$ CRISP $>$ CoD, with Pass@1 scores of 35.43%, 35.33%, and 23.81%, respectively. Further mathematical RL does not reverse CoD's disadvantage on this code evaluation. These endpoints also remain below the original model's 46.00%. Olmo-3-7B-Think has SFT evaluations only, so no post-RL comparison is made for that model.

These results show that the relative value of the training-data variants depends on both the student model and the evaluation task. In particular, CoD's advantage over CRISP on mathematical reasoning does not consistently extend to code generation: the ordering reverses for Qwen3-4B, while the difference is small for Olmo-3-7B-Think. The findings therefore support evaluating compressed reasoning data on downstream tasks individually rather than assuming uniform transfer from mathematics.

# F Target NLL and Policy KL Diagnostics

To better understand the downstream accuracy differences between CoT training datasets, we examine two possible explanations. First, the datasets may differ in how readily the original model predicts their targets, which could affect learning during SFT. Second, training on different targets may induce different degrees of departure from the original policy, potentially relating to downstream performance. We investigate these possibilities using target NLL under the original model and policy-to-base KL after SFT, respectively. These diagnostics include Qwen3-8B and do not involve RL checkpoints.

## F.1 Diagnostic protocol

**Matched samples.** We construct a common pool of 10,000 training questions by fixed-hash sampling from 57,880 IDs shared by the three datasets, checking that question texts agree and answers do not conflict, and removing duplicate questions after normalization. For target NLL, we draw 1,000 matched questions with seed 20260909, requiring all three prompt–target sequences to fit within 16,384 tokens under the Qwen3-8B chat template. Overlength examples are excluded rather than truncated. The same 1,000 questions are used for all three target variants, and all 3,000 sequences are successfully scored.

For training-domain KL, an independent fixed-hash sample with seed 20260910 selects 500 prompts from the common pool. This sample overlaps with the NLL sample on 55 questions. For unseen-domain KL, we use all 90 AIME 2024–2026 questions, with 30 per year. None has an exact normalized-prompt match in any of the three training sets. Each policy receives the same prompts within each domain and generates one trajectory per prompt.

**Models and uncertainty.** The reference model is the original Qwen3-8B, before any SFT or RL in our study. This model alone scores the training targets for NLL. For KL, we compare each final SFT policy against this reference: checkpoint 916 for Raw and CRISP, and checkpoint 913 for CoD. A Base-versus-Base implementation check yields values close to zero. Error bars are 95% nonparametric percentile bootstrap confidence intervals from 5,000 resamples, using the 2.5th and 97.5th percentiles of the bootstrap mean. Resampling units are questions for NLL and trajectories for KL; with one trajectory per prompt, the latter also corresponds to question-level resampling. Intervals are calculated separately for each method and are marginal intervals, not paired confidence intervals for between-method differences.

### F.2 TARGET NEGATIVE LOG-LIKELIHOOD

For target variant $m$ and question $x_i$, let $y_i^{(m)}$ contain $T_i^{(m)}$ assistant-target tokens. We compute

$$L_i^{(m)} = -\frac{1}{T_i^{(m)}} \sum_{t=1}^{T_i^{(m)}} \log \pi_{\text{base}}\left(y_{i,t}^{(m)} \mid x_i, y_{i,<t}^{(m)}\right),$$
$$\text{NLL}_{\text{macro}}^{(m)} = \frac{1}{N} \sum_{i=1}^{N} L_i^{(m)}. \tag{4}$$

The scored span is the complete assistant target, including reasoning, post-think text, the final answer, and the assistant end token <|im_end|>. Prompt and prompt-template tokens are excluded from the loss. Averaging first over tokens and then over questions assigns equal weight to each question, rather than weighting longer targets more heavily. All values use natural logarithms and are reported in nats per token.

Figure 15(a) shows that CRISP has the lowest macro NLL (0.281), followed by CoD (0.323) and Raw (0.471). Thus, the original model assigns higher average per-token log likelihood to CRISP targets than to CoD targets on this matched sample. Nevertheless, CoD produces stronger downstream Qwen3-8B SFT results in our experiments. The ordering therefore does not support using lower target NLL alone to rank the training value of these datasets. NLL measures compatibility with the reference model's token predictions; it does not directly measure reasoning correctness or the usefulness of the supervision.

### F.3 SFT-POLICY DIVERGENCE FROM THE ORIGINAL MODEL

For each SFT policy $\pi_m$, trajectories are sampled with temperature 1, top-$p = 1$, and top-$k$ disabled. The maximum generation length is 30,000 tokens within a 32,768-token context. On each policy-generated trajectory, we evaluate the sampled log-probability ratio

$$K_i^{(m)} = \frac{1}{T_i^{(m)}} \sum_{t=1}^{T_i^{(m)}} \log \frac{\pi_m\left(y_{i,t}^{(m)} \mid x_i, y_{i,<t}^{(m)}\right)}{\pi_{\text{base}}\left(y_{i,t}^{(m)} \mid x_i, y_{i,<t}^{(m)}\right)},$$
$$\widehat{\text{KL}}_{\text{macro}}^{(m)} = \frac{1}{N} \sum_{i=1}^{N} K_i^{(m)}. \tag{5}$$

This is a sampled, length-normalized log-ratio diagnostic in the policy-to-base direction, rather than an exact vocabulary-summed KL or an unnormalized sequence-level KL. We score the entire generated response, including the assistant end token when generated, and exclude prompt tokens. Each policy is evaluated along its own generated prefixes.

All trajectories are scored successfully, including those that reach the generation limit. Truncation affects 0/500, 1/500, and 2/500 training-domain trajectories for Raw, CRISP, and CoD, respectively; the corresponding AIME counts are 2/90, 1/90, and 8/90. The reported means therefore describe the observed responses, including capped prefixes, rather than only completed responses.

Figure 15(b) shows a reversal in the numerical ordering across prompt sets. On training prompts, divergence is highest for CoD, followed by CRISP and Raw. On unseen AIME prompts, Raw has the highest divergence, followed by CRISP and CoD. The training-domain ordering therefore does not carry over to the evaluation domain. In particular, CoD's stronger AIME performance coexists with lower measured divergence from Base on AIME, so its advantage cannot be characterized simply as greater departure from the original policy.

**Interpretation and limitations.** Together, these diagnostics show that neither lower target NLL nor a domain-independent ranking of policy divergence summarizes the observed training outcomes. They complement the downstream evaluations by testing these candidate summaries; they do not establish that exploration is preserved or identify a causal mechanism. The analysis covers one model family and one training run per variant. NLL excludes overlength targets, while KL uses one

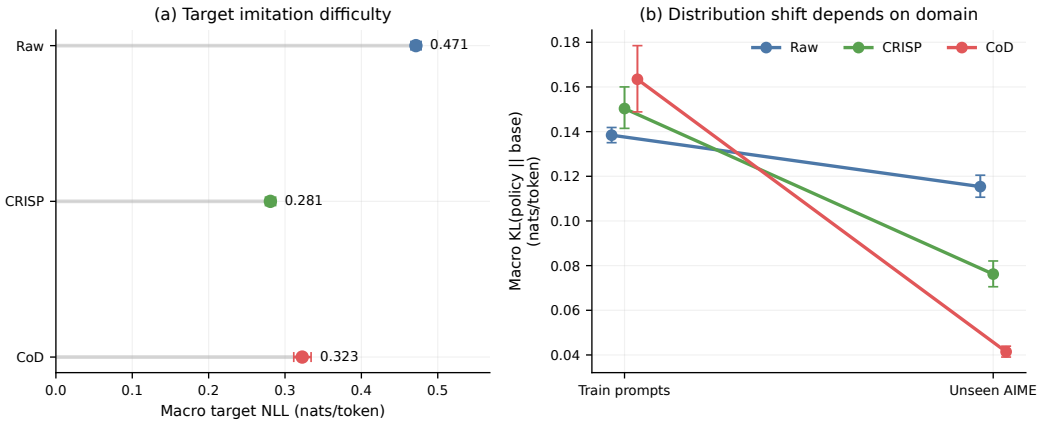



Figure 15: **Target NLL and SFT-policy divergence for Qwen3-8B.** (a) Original-model NLL on three matched target variants for 1,000 training questions. (b) Sampled policy-to-base log-ratio estimates for final SFT policies on 500 training prompts and all 90 AIME 2024–2026 prompts, with one trajectory per prompt. Both panels average within each sequence and then equally across sequences; units are nats per token. Error bars show separately computed 95% percentile bootstrap confidence intervals from 5,000 resamples. No RL policies are included.

trajectory per prompt and includes unequal truncation rates, especially for CoD on AIME. Moreover, methods generate different prefixes, so their KL estimates reflect both their token distributions and the contexts they visit. Bootstrap intervals describe uncertainty in these observed samples, not training-seed variation or repeated-generation variability for each fixed prompt. Exact prompt matching also does not exclude semantic overlap. These scope limits should guide interpretation of the numerical orderings.