WHEN TOKENIZERS DIVERGE, REASONING CAN TRANSFER: CROSS-TOKENIZER ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation provides dense teacher guidance on student-generated trajectories, but token-level distribution matching typically assumes a shared vocabulary. We present **Cross-Tokenizer On-Policy Distillation (CT-OPD)** for transferring mathematical reasoning across model families. Decode-Verified Chunk Alignment (DVCA) constructs corresponding text spans despite differences in token boundaries. The Cross-Tokenizer Rank-Divergence Score (CRDS) then combines rank-guided direction, log-probability magnitude, and entropy-based confidence to produce a student-token advantage. This auxiliary signal is annealed while the outcome-reward advantage remains active throughout training. Using a Qwen2.5-32B-SPEAR teacher, CT-OPD improves over outcome-only RL on all five evaluated benchmarks for both Youtu-Llm-2B-Base and Llama-3.1-8B-Instruct, with AIME24 gains of 6.66 and 3.55 percentage points, respectively. It achieves the best student result on nine of the ten model–benchmark combinations. Component ablations support the combined use of rank, magnitude, and confidence weighting, highlighting the importance of both correspondence and scoring in cross-tokenizer distillation.

1 INTRODUCTION

Reinforcement learning (RL) with verifiable rewards improves mathematical reasoning in large language models Shao et al. (2024); Guo et al. (2025); Yu et al. (2025), but smaller students must explore long trajectories with sparse outcome feedback. *On-policy distillation* complements this signal by querying a stronger teacher on student-generated prefixes, including mistakes. Across model families, however, different vocabularies and token boundaries prevent direct next-token distribution matching. Retokenizing a response lets the teacher score its content, but does not determine how to supervise the student's original actions.

For mathematical reasoning, the distinction between trajectory-level success and local decisions is important. A response may contain a useful intermediate calculation, an unnecessary detour, and a recoverable error before reaching its final answer. The outcome reward assesses the resulting trajectory, whereas teacher feedback can vary along that trajectory. Scoring the student's own prefixes makes this guidance relevant to the states the student actually visits during learning. In tool-augmented settings, these states also include returned observations that inform the next decision, while the policy update acts on the assistant's generated reasoning and tool calls.

Two issues must be addressed together: *correspondence* and *scoring*. Independently decoding pieces of a multi-byte character can corrupt byte-count-based alignment; tool-augmented reasoning Feng et al. (2025) also requires separating generated actions from contextual observations. Even after alignment, mean token log-probabilities depend on segmentation and calibration. Their signed gap can reflect output sharpness as well as preference for the realized text. Building on existing cross-tokenizer distillation (Section 2), we ask how aligned feedback should determine the direction, strength, and action-level assignment of an auxiliary update.

We propose **Cross-Tokenizer On-Policy Distillation (CT-OPD)**. *Decode-Verified Chunk Alignment (DVCA)* pairs contiguous token spans with matching clean decoded content despite unequal token counts. The *Cross-Tokenizer Rank-Divergence Score (CRDS)* combines rank-guided direction, log-

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.

probability-gap magnitude, and teacher-entropy confidence. Each student action in a chunk receives the same score; unmatched actions retain outcome supervision without auxiliary feedback. Only the CRDS coefficient is annealed, keeping outcome-reward optimization active throughout training.

The resulting training signal connects two levels of supervision. At the local level, aligned chunks provide a common text unit for aggregating each model's native token statistics. Rank comparison sets the direction of guidance, the log-probability gap scales its strength, and entropy weighting emphasizes concentrated teacher predictions. At the trajectory level, the outcome advantage reinforces task success across the sampled response. Their additive combination lets intermediate decisions receive distinct teacher feedback even when trajectories share the same outcome reward. Annealing gradually shifts the relative emphasis from this combined signal toward outcome-based learning.

Using a Qwen2.5-32B-SPEAR teacher, CT-OPD improves over outcome-only RL on all five evaluated benchmarks for Youtu-Llm-2B-Base and Llama-3.1-8B-Instruct, with average gains of 3.59 and 3.63 percentage points and the best student result on nine of ten model–benchmark combinations.

Our contributions are summarized as follows:

- We develop a rank-guided, confidence-weighted chunk advantage, with a signed-gap fallback when mean ranks tie.
- We integrate DVCA, uniform action-level broadcasting, and auxiliary-only annealing with an independent outcome advantage.
- We evaluate two cross-tokenizer students on five reasoning benchmarks and analyze scoring, alignment, and annealing through ablations and sensitivity studies.

2 Related Work

Knowledge distillation and on-policy learning. Distillation uses soft targets Hinton et al. (2015) or teacher-generated sequences Kim & Rush (2016), including for encoder compression Sanh et al. (2019); Jiao et al. (2020). MiniLLM uses reverse KL Gu et al. (2024); GKD studies student-generated trajectories and joint reward optimization Agarwal et al. (2024); DistiLLM uses skew divergences Ko et al. (2024); and GAKD explores adversarial distillation Pu et al. (2026). CT-OPD builds on on-policy supervision with an auxiliary advantage for heterogeneous tokenizers.

Cross-tokenizer distribution and likelihood matching. ULD and MultiLevelOT use optimal transport Boizard et al. (2024); Cui et al. (2025); DSKD learns shared-space projections Zhang et al. (2024); and CDM uses contextual mappings Chen et al. (2025). ALM compares shared-text chunk probabilities through a binarized divergence, optionally with outcome-chunk debiasing and hidden-state distillation Minixhofer et al. (2025). GOLD combines sequence merging with direct vocabulary matching and a ULD-style loss for unmatched tokens Patiño et al. (2025). SimCT extends supervision to short multi-token continuations realizable by both tokenizers Sun et al. (2026). These methods match distributions or shared-text likelihoods, whereas CT-OPD aggregates mean realized-token statistics from sampled chunks into an auxiliary policy-gradient advantage.

Chunk-level credit assignment. Niu et al. (2026) align synchronized decoded chunks and allocate teacher chunk log-probabilities proportionally to student token log-probabilities. CT-OPD uses a rank-guided, entropy-weighted advantage: mean ranks determine direction, and an absolute mean log-probability gap determines magnitude when ranks differ. Each student action receives the full chunk score alongside an independent outcome advantage, with annealing applied to the auxiliary term.

Byte-space and exact-span supervision. BPM maps teacher probability mass through shared byte prefixes, retaining unmatched mass in a residual category Wang et al. (2026). SimpleOPD matches individual tokens occupying identical text spans and uses student-reference KL and termination-token advantage masking to control length and stability He et al. (2026). DVCA admits multi-token span pairs with unequal token counts, with CRDS supplying chunk-level feedback.

Confidence-weighted guidance. EGRSD combines reward-grounded direction, likelihood-ratio magnitude, and teacher-entropy confidence in self-distillation; its gate has a positive floor, and

its lookahead variant uses subsequent entropy Ke et al. (2026). CT-OPD shares the direction–magnitude–confidence decomposition but uses cross-tokenizer rank direction, mean log-probability-gap magnitude, and exponential entropy attenuation without a positive floor. Outcome feedback remains a separate advantage.

3 METHODOLOGY

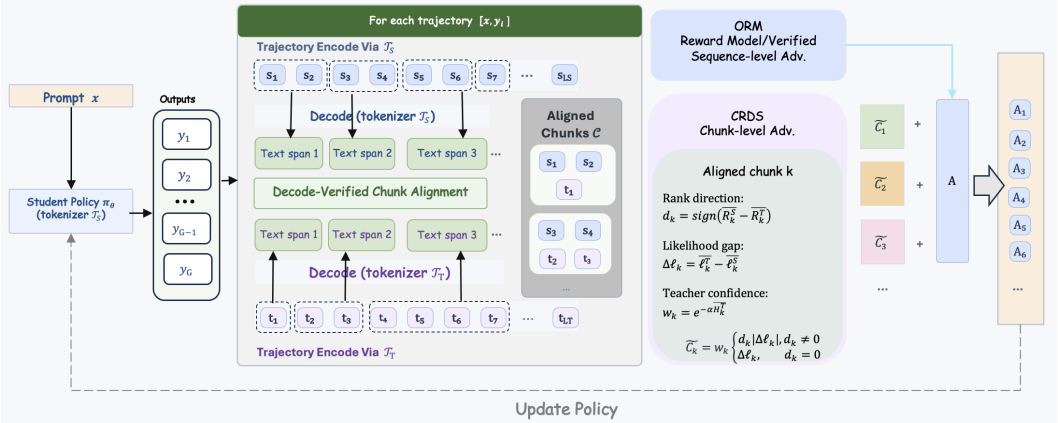


Figure 1: Overview of CT-OPD. DVCA aligns student-generated trajectories into chunks with matching decoded content. CRDS combines rank direction, log-probability-gap magnitude, and teacher-entropy weighting (Eqs. 10–11). Each chunk score is broadcast to its student tokens and combined with the sequence-level outcome advantage for policy optimization. During training, the CRDS coefficient decreases linearly while the outcome coefficient stays at one (Eq. 14).

Figure 1 summarizes CT-OPD. We first formalize the on-policy setting (§3.1), then describe decode-verified chunk alignment (§3.2), and finally construct the CRDS advantage and combine it with outcome feedback (§3.3).

3.1 PROBLEM FORMULATION

Let π_θ denote the student policy and π_T a frozen teacher. Each model $M \in \{S, T\}$ has a tokenizer $\mathcal{T}_M$ and a conversation serializer $\mathcal{F}_M$. Given an initial task x, the student interacts with the tool environment to produce a trajectory τ, consisting of its generated assistant messages and the intervening tool observations. The two models encode corresponding views of this interaction using their own native formats:

$$\mathbf{s} = \mathcal{T}_S(\mathcal{F}_S(x, \tau)), \qquad \mathbf{t} = \mathcal{T}_T(\mathcal{F}_T(x, \tau)). \tag{1}$$

For the student, Eq. 1 denotes a lossless representation of the recorded rollout, preserving the original generated token IDs and their conditioning prefixes. The teacher representation is constructed from the same observed interaction. Alignment operates locally on corresponding response content within each model's native conversation format.

Write $L_S = |\mathbf{s}|$ and $L_T = |\mathbf{t}|$. Let $\mathcal{I}_S$ be the positions associated with observed student-generated actions, and let $\mathcal{I}_T$ contain their eligible counterparts in the teacher representation. Prompt tokens, tool observations, and template-inserted context remain available for conditioning but are not themselves student action targets. Each model is scored on its full native prefix:

$$\ell_i^S = \log \pi_{\theta_{\text{old}}}(s_i \mid \mathbf{s}_{<i}), \qquad R_i^S = \text{rank}_{\pi_{\theta_{\text{old}}}}(s_i \mid \mathbf{s}_{<i}), \quad i \in \mathcal{I}_S, \tag{2}$$

$$\begin{aligned} \ell_j^T &= \log \pi_T(t_j \mid \mathbf{t}_{<j}), \qquad R_j^T = \text{rank}_{\pi_T}(t_j \mid \mathbf{t}_{<j}), \\ H_j^T &= \mathcal{H}[\pi_T(\cdot \mid \mathbf{t}_{<j})], \qquad j \in \mathcal{I}_T. \end{aligned} \tag{3}$$

Here R is the 0-based rank of the realized token, defined as the number of vocabulary items with strictly higher probability. Student statistics are evaluated at the rollout-policy snapshot θ_{old} and held fixed during the update. Each prefix includes the initial prompt and all preceding interaction

messages. The chunk alignment below establishes correspondence between the two token sequences for aggregating these statistics.

3.2 Decode-Verified Chunk Alignment

Verifying decoded spans. Different tokenizers can divide the same response into different numbers of tokens. DVCA searches within corresponding response segments for clean, contiguous spans with equal decoded content. Let D_M denote span decoding for model M. An accepted chunk pair (S_k, T_k) satisfies

$$D_S(\mathbf{s}[S_k]) = D_T(\mathbf{t}[T_k]), \qquad S_k \subseteq \mathcal{I}_S, \quad T_k \subseteq \mathcal{I}_T. \tag{4}$$

Starting at the current pair of cursors, DVCA expands candidate spans until it finds a verified pair within a bounded search. It then advances both cursors to the ends of that pair. Invalid partial decodes are rejected; bounded recovery records skipped or unresolved spans as unmatched. Candidate expansion and recovery stay within the corresponding response segments. Algorithm 2 gives the local procedure.

Joint decoding. Computing byte lengths from isolated token decodes can introduce incorrect alignment boundaries. A token containing only part of a multi-byte character may decode to the replacement character `U+FFFD`; re-encoding that replacement corrupts the byte count. DVCA checks jointly decoded spans, preserving the character across token boundaries. Appendix B illustrates this isolated-decoding failure mode.

Correspondence and action validity. The accepted pairs form the correspondence set

$$\mathcal{C} = \{(S_k, T_k)\}_{k=1}^{K}. \tag{5}$$

Student spans are non-overlapping, and every score is mapped back to the original generated token positions. Prompt tokens, tool observations, and template-inserted markers are excluded from the action targets. Unmatched actions receive zero CRDS advantage while retaining their independent outcome-reward signal. Appendix C extends this interface to typed fields and explicitly paired control events when native tool serializations differ.

3.3 Cross-Tokenizer Rank-Divergence Score (CRDS)

Given correspondence units $\mathcal{C}$, the *Cross-Tokenizer Rank-Divergence Score* (CRDS) constructs a local teacher signal. Unlike approaches that transport full predictive distributions Cui et al. (2025) or learn projection heads Zhang et al. (2024), it uses realized-token log-probabilities, ranks, and teacher entropy, aggregated over each aligned chunk.

3.3.1 Chunk-Level Statistic Aggregation

For $M \in \{S, T\}$, let $I_k^S = S_k$ and $I_k^T = T_k$ be the two token index sets in unit k. We use mean aggregation:

$$\begin{aligned} \bar{\ell}_k^M &= \frac{1}{|I_k^M|} \sum_{i \in I_k^M} \ell_i^M, \qquad \bar{R}_k^M = \frac{1}{|I_k^M|} \sum_{i \in I_k^M} R_i^M, \\ \bar{H}_k^T &= \frac{1}{|T_k|} \sum_{j \in T_k} H_j^T. \end{aligned} \tag{6}$$

Mean aggregation accommodates unequal token counts within an aligned pair. The statistic $\bar{\ell}_k^M$ measures average log-probability per token under model M's tokenization.

3.3.2 Rank-Guided Divergence Scoring

The log-probability gap $\bar{\ell}_k^T - \bar{\ell}_k^S$ combines two effects across heterogeneous tokenizers: a *directional* preference for the realized text within each model, and a *calibration* offset associated with model size, vocabulary size, and output sharpness. These effects jointly determine the sign of the gap. We separate the direction and magnitude of guidance by using realized-token ranks for direction and the absolute log-probability gap for magnitude.

To illustrate the calibration effect, consider a latent within-model preference score u_k^M for the realized chunk in a simple affine model:

$$\bar{\ell}_k^M = a_M + b_M u_k^M, \qquad b_M > 0, \tag{7}$$

where a_M and b_M absorb model-specific calibration effects. If one uses the raw difference directly, its sign becomes

$$\operatorname{sign}(\bar{\ell}_k^T - \bar{\ell}_k^S) = \operatorname{sign}\big((a_T - a_S) + b_T u_k^T - b_S u_k^S\big). \tag{8}$$

A shared affine calibration ($a_T = a_S$ and $b_T = b_S$) preserves the sign of $u_k^T - u_k^S$ in this model. Model-specific offsets and scales can reverse it, so a chunk can exhibit $u_k^T > u_k^S$ while having $\bar{\ell}_k^T < \bar{\ell}_k^S$.

We therefore decouple the signal into two complementary components. The *rank* R measures how a model prioritizes the realized token among its candidates ($R = 0$ for top-1). On a fixed candidate set, it is invariant to a strictly increasing transformation of logits. Formally, for logits $\mathbf{z}$, realized token v, and any strictly increasing function f,

$$\operatorname{rank}_{\operatorname{softmax}(\mathbf{z})}(v) = \operatorname{rank}_{\operatorname{softmax}(f(\mathbf{z}))}(v), \tag{9}$$

whereas probabilities generally change. CRDS uses cross-tokenizer rank comparison to set an ordinal direction and the absolute log-probability gap to scale its magnitude. For correspondence unit k,

$$\mathrm{CRDS}_k = \begin{cases} \operatorname{sign}(\bar{R}_k^S - \bar{R}_k^T) \cdot |\bar{\ell}_k^T - \bar{\ell}_k^S|, & \text{if } \bar{R}_k^S \neq \bar{R}_k^T, \\ \bar{\ell}_k^T - \bar{\ell}_k^S, & \text{otherwise.} \end{cases} \tag{10}$$

When the mean ranks differ, rank supplies an ordinal direction and $|\bar{\ell}_k^T - \bar{\ell}_k^S|$ scales the update; ties fall back to the signed log-probability gap. Section 4 evaluates this direction rule through a raw-gap ablation.

3.3.3 Entropy-Aware Confidence Weighting

The teacher's predictive uncertainty varies across positions. Entropy provides a confidence proxy: concentrated teacher predictions receive stronger CRDS weighting, while diffuse predictions receive weaker weighting:

$$\widetilde{C}_k = \mathrm{CRDS}_k \cdot \exp\big(-\alpha \bar{H}_k^T\big), \tag{11}$$

where entropy is measured in nats over the teacher vocabulary and $\alpha \geq 0$ controls attenuation. Low-entropy, concentrated predictions receive greater weight; setting $\alpha = 0$ removes the weighting.

The unit scores are projected back onto the student action grid:

$$A_i^{\mathrm{CRDS}} = m_i^{\mathrm{gen}} \sum_{k \in \mathcal{K}_{\mathrm{valid}}} \widetilde{C}_k \, \mathbf{1}[i \in S_k], \tag{12}$$

where m_i^{gen} marks observed generated actions, and $\mathcal{K}_{\mathrm{valid}}$ indexes verified chunk pairs. Student chunks are non-overlapping, and each generated position in S_k receives the full chunk score $\widetilde{C}_k$. Unmatched actions receive zero CRDS advantage and retain their independent outcome-reward supervision.

For each prompt, we sample G trajectories and use a group-relative outcome-reward advantage:

$$\bar{r} = \frac{1}{G}\sum_{g=1}^{G} r_g, \qquad d_r = \frac{1}{G}\sum_{g=1}^{G} |r_g - \bar{r}|, \qquad A_{g,i}^{\mathrm{ORM}} = \frac{r_g - \bar{r}}{d_r + \epsilon}, \tag{13}$$

where $\epsilon > 0$ stabilizes the denominator. The normalization uses mean absolute deviation; equal-reward groups have zero outcome advantage. The advantage is shared across observed assistant-action positions in sample g. The outcome term has coefficient one, while the auxiliary CRDS term is annealed:

$$\eta(t) = \max\left(1 - \frac{t}{T_{\max}}, 0\right), \qquad A_{g,i} = A_{g,i}^{\mathrm{ORM}} + \eta(t) A_{g,i}^{\mathrm{CRDS}}. \tag{14}$$

Here t counts policy-update steps and $T_{\max} > 0$ is the annealing horizon. Early updates combine dense teacher feedback with the outcome reward. At $t \geq T_{\max}$, the objective reduces to outcome-reward optimization.

3.3.4 Optimization Objective and Reverse-KL View

CT-OPD optimizes the mixed advantage with the clipped policy-gradient objective used in PPO- and DAPO-style training Schulman et al. (2017); Yu et al. (2025):

$$\mathcal{L}_{\text{CT-OPD}}(\theta) = -\mathbb{E}\Big[\sum_{i\in\mathcal{I}_S} \min\Big(\rho_i(\theta)A_i, \; \text{clip}(\rho_i(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}})A_i\Big)\Big], \tag{15}$$

where $\rho_i(\theta) = \pi_\theta(s_i \mid \mathbf{s}_{<i})/\pi_{\theta_{\text{old}}}(s_i \mid \mathbf{s}_{<i})$, and the expectation is over prompts and rollout trajectories. Advantages, alignment decisions, and teacher statistics are stop-gradient quantities; gradients flow through the student likelihood in ρ_i. The objective sums over action positions before averaging trajectories, retaining the corresponding response-length weighting.

For distributions p_θ and p_T on a common output space, negative reverse KL has the score-function gradient

$$\nabla_\theta[-D_{\text{KL}}(p_\theta \| p_T)] = \mathbb{E}_{y\sim p_\theta}[(\log p_T(y) - \log p_\theta(y))\nabla_\theta \log p_\theta(y)], \tag{16}$$

under the usual support and differentiability conditions. This identity motivates the teacher–student log-probability gap as an on-policy signal on a shared output space. For heterogeneous tokenizers, CT-OPD instead constructs a policy-gradient surrogate from aligned chunk statistics. Rank-guided direction and entropy weighting transform the gap into the CRDS advantage used in Eq. 15.

4 Experiments

4.1 Experimental Setup

Models and Data. Our teacher is Qwen2.5-32B Yang et al. (2024), post-trained for mathematical reasoning with SPEAR Qin et al. (2025). We evaluate two students with tokenizers different from the teacher's: the base model **Youtu-Llm-2B-Base** Lu et al. (2025) and **Llama-3.1-8B-Instruct** Grattafiori et al. (2024). Training prompts are drawn from DAPO-Math-17K Yu et al. (2025). Evaluation covers AIME24, AIME25, GSM8K Cobbe et al. (2021), MATH500 Hendrycks et al. (2021), and OlympiadBench He et al. (2024). Training rollouts and evaluation generations use a maximum response length of 16,384 tokens. The reward function and training configuration are given in Appendix A.

Baselines and Configurations. Table 1 includes the initial student checkpoint (**Base**) and four training baselines: **SFT** on teacher-generated traces Kim & Rush (2016); **GRPO**, the outcome-reward RL baseline without teacher distillation Shao et al. (2024); **GOLD**; and **DSKD**, dual-space knowledge distillation Zhang et al. (2024). CT-OPD adds the auxiliary CRDS signal to outcome-reward optimization. The outcome reward combines answer correctness with the tool-use adjustment in Appendix A; outcome-only training optimizes this reward without the distillation advantage.

4.2 Main Results

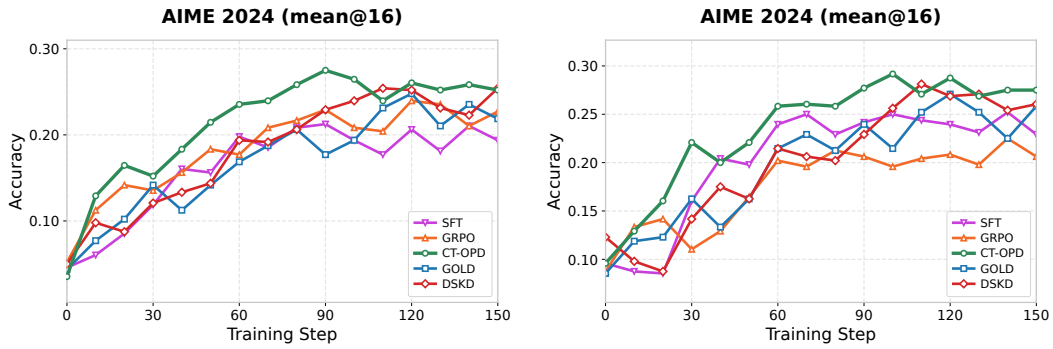
Table 1 compares the two cross-tokenizer students. CT-OPD achieves the highest average for each student and the best result on nine of the ten model–benchmark combinations.

Youtu-Llm-2B-Base. CT-OPD improves the five-benchmark average from 39.04% with GRPO to 42.63%, and exceeds DSKD by 1.30 percentage points. It outperforms GRPO on all five benchmarks, improving AIME24 from 22.50% to 29.16% and AIME25 from 20.83% to 24.79%. It obtains the strongest results on both AIME benchmarks, GSM8K, and OlympiadBench. DSKD performs best on MATH500, where CT-OPD trails by 0.95 points (53.04% versus 53.99%).

Llama-3.1-8B-Instruct. CT-OPD achieves the best result on every benchmark. It improves the five-benchmark average from 37.82% with GRPO to 41.45%, and exceeds DSKD, the strongest competing baseline by average accuracy, by 1.96 points. On AIME24 and AIME25, the gains over GRPO are 3.55 and 5.42 points, respectively. Together, the two student configurations demonstrate improvements across different tokenizer families and student initializations, covering both a 2B base model and an 8B instruction-tuned model.

Table 1: Mathematical reasoning results (mean@16, %). Both students use tokenizers different from the Qwen teacher's. **Bold** indicates the best result within each student block; the teacher is shown for reference. Avg. is the unweighted mean over the five benchmarks.

Method	AIME24	AIME25	GSM8K	MATH500	OlympiadBench	Avg.
Teacher: Qwen2.5-32B-SPEAR						
Teacher	66.25	60.19	94.86	78.54	43.27	68.62
Cross-Tokenizer Student: Youtu-Llm-2B-Base						
Base	8.95	7.43	75.37	48.35	18.47	31.71
SFT	25.20	22.71	77.95	50.71	21.84	39.68
GRPO	22.50	20.83	78.52	51.13	22.23	39.04
GOLD	25.83	23.12	77.03	52.63	23.95	40.51
DSKD	27.08	22.70	79.24	**53.99**	23.66	41.33
CT-OPD	**29.16**	**24.79**	**80.16**	53.04	**26.01**	**42.63**
Cross-Tokenizer Student: Llama-3.1-8B-Instruct						
Base	4.16	2.91	70.73	46.12	14.30	27.64
SFT	21.25	18.54	73.95	50.71	17.84	36.46
GRPO	23.95	19.16	75.62	51.13	19.26	37.82
GOLD	24.79	20.62	74.36	51.49	18.37	37.93
DSKD	25.41	21.45	77.41	52.93	20.23	39.49
CT-OPD	**27.50**	**24.58**	**79.31**	**54.34**	**21.52**	**41.45**



(a) Llama-3.1-8B-Instruct (b) Youtu-Llm-2B-Base

Figure 2: AIME24 mean@16 accuracy at 10-step intervals over 150 policy-update steps. Both panels show cross-tokenizer students. Accuracy is plotted as a fraction, with panel-specific vertical-axis ranges.

4.3 Convergence Analysis

Figure 2 shows an early accuracy advantage for CT-OPD in both student configurations. On Llama, CT-OPD reaches approximately 0.21 by step 50 and peaks near 0.275 around step 90. On Youtu, it reaches approximately 0.22 by step 30 and peaks near 0.29 around step 100. DSKD is competitive later in training and exceeds CT-OPD at some checkpoints. These curves indicate stronger early progress per policy update.

4.4 Ablation Studies

We conduct ablation experiments to quantify the contribution of each component. All ablations use the Qwen2.5-32B-SPEAR → Youtu-Llm-2B-Base cross-tokenizer configuration and report results on AIME24, AIME25, and MATH500.

Table 2: Component ablations for Qwen2.5-32B-SPEAR → Youtu-Llm-2B-Base (mean@16, %). Each variant changes the named component of CT-OPD. The no-CRDS variant is the GRPO baseline.

Configuration	AIME24	AIME25	MATH500
CT-OPD (full)	**29.16**	**24.79**	**53.04**
CRDS Components			
w/o rank (log-prob gap only)	25.21	23.51	52.32
w/o log-prob magnitude (sign only)	24.72	22.36	52.01
w/o entropy weighting ($\alpha = 0$)	19.79	17.13	49.43
Alignment Algorithm			
Byte-prefix alignment	1.04	0.63	3.27
Annealing Strategy			
Constant CRDS weight ($\eta = 1$)	27.21	22.29	52.80
No CRDS ($\eta = 0$, GRPO only)	22.50	20.83	51.13
Fixed CRDS weight ($\eta = 0.5$)	28.72	23.78	52.34

Table 3: Text-alignment quality on 1,000 sampled traces (%). Text match measures the percentage of proposed chunk pairs with identical decoded content after NFC normalization. Coverage measures the percentage of student tokens in text-matched chunks. Results are grouped by English mathematical text and mixed mathematical text with CJK characters.

	English math		Mixed (math + CJK)	
Method	Text match	Coverage	Text match	Coverage
Byte-prefix	78.4	84.1	21.7	34.5
DVCA (ours)	**100.0**	**99.7**	**100.0**	**99.8**

Table 2 groups the variants by scoring, alignment, and annealing.

CRDS components. The raw-gap ablation replaces rank-based direction with the signed log-probability gap, keeping alignment, entropy weighting, and uniform broadcasting fixed. This reduces AIME24 accuracy by 3.95 points; removing log-probability magnitude reduces it by 4.44 points. Both variants also underperform on AIME25 and MATH500. Removing entropy weighting has the largest effect among the scoring ablations: AIME24 falls from 29.16% to 19.79%, below the outcome-only baseline. These comparisons support the combined use of directional, magnitude, and confidence information in the teacher signal.

Alignment algorithm. Replacing DVCA with the tested byte-prefix implementation reduces accuracy to 1.04%, 0.63%, and 3.27%. Appendix B illustrates an isolated-decoding failure mode in which corrupted cumulative byte counts disrupt subsequent correspondence.

Annealing strategy. The linear schedule achieves the highest accuracy on all three benchmarks. A fixed coefficient of 0.5 is the closest alternative on AIME24 and AIME25, trailing by 0.44 and 1.01 points, respectively. On MATH500, a constant coefficient of 1 is closer, trailing by 0.24 points. Both fixed-weight variants improve over the no-CRDS baseline, and linear annealing further improves accuracy by gradually reducing teacher guidance during training. The outcome-reward coefficient remains one for every schedule.

4.5 Analysis

4.5.1 Alignment Quality

Table 3 compares decoded-text consistency and token coverage. DVCA's acceptance test enforces matching decoded content, yielding a 100% text-match rate while retaining 99.7–99.8% of student tokens. The byte-prefix baseline reaches 78.4% text match on English math and 21.7% on mixed-language traces, with coverage of 84.1% and 34.5%, respectively.

Table 4: Sensitivity of CT-OPD to the CRDS annealing horizon $T_{\max}$ (policy-update steps) and entropy-weighting coefficient α. AIME24 scores are reported as mean@16 (%); the best result in each sweep is bold.

$T_{\max}$	30	50	80	120
AIME24 (%)	24.58	**29.16**	28.75	27.50

α	0	0.1	0.3	0.5	1.0	2.0
AIME24 (%)	19.79	23.95	25.83	**29.16**	27.29	24.37

4.5.2 Hyperparameter Sensitivity

Table 4 examines the annealing horizon and entropy weighting in separate sweeps. For $T_{\max}$, accuracy increases from 24.58% at 30 steps to 29.16% at 50 steps, then decreases to 28.75% and 27.50% at 80 and 120 steps. The 50-step horizon gives the best result, followed by 80 steps with a gap of 0.41 points. For entropy weighting, every tested positive α outperforms $\alpha = 0$, with accuracy peaking at $\alpha = 0.5$. Increasing α to 1.0 or 2.0 reduces accuracy to 27.29% and 24.37%, respectively. These sweeps show a common pattern: an intermediate duration and strength of teacher guidance yield higher accuracy than either shorter or longer horizons and weaker or stronger attenuation.

5 Limitations

CT-OPD requires teacher scoring during training, adding computation relative to outcome-only RL. Accuracy-versus-step curves do not quantify this overhead. Evaluation is limited to mathematical reasoning, two student families, and one teacher; repeated-run uncertainty and broader task coverage are needed to assess generality. CRDS uses tokenization-dependent mean log-probabilities, ranks, and entropy rather than calibrated semantic likelihoods, and its annealing schedule remains a heuristic. A confident but incorrect teacher can still produce misleading guidance.

The template-aware extension in Appendix C requires validated extraction, canonicalization, and original-token provenance for each serialization. Arbitrary field reorderings and incomplete outputs may not admit a faithful causal correspondence.

6 Conclusion

We presented **Cross-Tokenizer On-Policy Distillation (CT-OPD)**, a framework for transferring reasoning through teacher feedback on student-generated trajectories without requiring a shared tokenizer. CT-OPD addresses two complementary challenges: DVCA establishes correspondence through jointly decoded text spans, while CRDS converts chunk-level rank, log-probability, and entropy statistics into an auxiliary advantage uniformly assigned to the student's original actions. Unmatched actions retain outcome supervision without receiving a distillation signal. Annealing only the auxiliary advantage keeps outcome feedback active throughout training, allowing early teacher guidance to give way to task-reward optimization.

Across five mathematical reasoning benchmarks, CT-OPD improves over outcome-only RL for Youtu-Llm-2B-Base and Llama-3.1-8B-Instruct, with average accuracy gains of 3.59 and 3.63 percentage points, respectively. It achieves the best student result on nine of ten model–benchmark combinations. Ablations and sensitivity results support combining reliable correspondence with directional, confidence-weighted feedback and controlled guidance duration. These findings suggest that cross-family teachers can guide intermediate decisions without replacing the task reward, even when their tokenizations differ. Future work should examine broader tasks and model pairs, adaptive schedules, and the template-aware extension for heterogeneous tool interactions.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024.

Nicolas Boizard, Kevin El Haddad, Céline Hudelot, and Pierre Colombo. Towards cross-tokenizer distillation: The universal logit distillation loss for LLMs. *arXiv preprint arXiv:2402.12030*, 2024.

Yijie Chen, Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. Enhancing cross-tokenizer knowledge distillation with contextual dynamical mapping. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 8005–8018, 2025.

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

Xiao Cui, Mo Zhu, Yulei Qin, Liang Xie, Wengang Zhou, and Houqiang Li. Multi-level optimal transport for universal cross-tokenizer knowledge distillation on language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23724–23732, 2025.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. ReTool: Reinforcement learning for strategic tool use in LLMs. *arXiv preprint arXiv:2504.11536*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, et al. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

Haonan He, Haodi Lei, Yun Luo, Haoran Zhang, Shunkai Zhang, Yizhuo Li, Shengji Tang, Zhilin Wang, Runzhe Zhan, Lei Bai, Ganqu Cui, Fangchen Yu, Yafu Li, Peng Ye, Ning Ding, and Yu Cheng. SimpleOPD: Simple tokenizer-agnostic on-policy distillation for long-context reasoning. *arXiv preprint arXiv:2608.14277*, 2026.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, et al. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

Junlong Ke, Zichen Wen, Weijia Li, Conghui He, and Linfeng Zhang. Respecting self-uncertainty in on-policy self-distillation for efficient LLM reasoning. *arXiv preprint arXiv:2605.13255*, 2026.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*, 2024.

Junru Lu, Jiarui Qin, Lingfeng Qiao, et al. Youtu-LLM: Unlocking the native agentic potential for lightweight large language models. *arXiv preprint arXiv:2512.24618*, 2025.

Benjamin Minixhofer, Ivan Vulić, and Edoardo Maria Ponti. Universal cross-tokenizer distillation via approximate likelihood matching. *arXiv preprint arXiv:2503.20083*, 2025.

Yifan Niu, Han Xiao, Dongyi Liu, Zelong Wang, Dihong Gong, Yasheng Wang, and Jia Li. Breaking the tokenizer barrier: On-policy distillation across model families. *arXiv preprint arXiv:2606.09456*, 2026.

Carlos Miguel Patiño, Kashif Rasul, Quentin Gallouédec, Ben Burtenshaw, Sergio Paniego, Vaibhav Srivastav, Thibaud Frere, Ed Beeching, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Unlocking on-policy distillation for any model family, 2025. Hugging Face technical article. URL `https://huggingface.co/spaces/HuggingFaceH4/on-policy-distillation`.

Yuan Pu, Shutong Lin, Xi Chen, Seunggeun Kim, Sungyoung Lee, Zhuolun He, David Z. Pan, and Bei Yu. GAKD: Generative adversarial knowledge distillation for large language models, 2026. URL `https://openreview.net/forum?id=9pfWYxYHAn`. Submitted to ICLR 2026.

Yulei Qin, Xiaoyu Tan, Zhengbao He, Gang Li, Haojia Lin, Zongyi Li, Zihan Xu, Yuchen Shi, Siqi Cai, Renting Rui, et al. Learn the ropes, then trust the wins: self-imitation with progressive exploration for agentic reinforcement learning. *arXiv preprint arXiv:2509.22601*, 2025.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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.

Jie Sun, Mao Zheng, Mingyang Song, Qiyong Zhong, Yilin Cheng, Bichuan Feng, Pengfei Liu, Junfeng Fang, and Xiang Wang. SimCT: Recovering lost supervision for cross-tokenizer on-policy distillation. *arXiv preprint arXiv:2605.07711*, 2026.

Hao Wang, Kun Yuan, Wenlin Zhong, Minglei Zhang, Han Xiao, Ming Sun, and Honggang Qi. Cross-tokenizer on-policy distillation via byte-prefix marginalization. *arXiv preprint arXiv:2607.22334*, 2026.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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. *arXiv preprint arXiv:2503.14476*, 2025.

Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen, and Jinan Xu. Dual-space knowledge distillation for large language models. *arXiv preprint arXiv:2406.17328*, 2024.

A RL Training Configuration

Reward function. The outcome reward combines a strict boxed-answer math score with a turn-based tool-use adjustment. Let

$$\begin{aligned} r_{\text{math}} &= \text{MathDAPO}(y, a^\star; \texttt{strict_box_verify} = \textbf{true}), \\ r_{\text{tool}} &= 0.05\,(n_{\text{turn}} - 2), \end{aligned} \tag{17}$$

where y denotes the recorded assistant response used by the outcome checker, n_{turn} is read from rollout metadata, and its default is 2. The final reward is

$$r = \begin{cases} r_{\text{math}} + \mathbf{1}[n_{\text{turn}} > 2], & \text{if the boxed answer is correct,} \\ \min\big(0, r_{\text{math}} + r_{\text{tool}}\big), & \text{if the boxed answer is wrong.} \end{cases} \tag{18}$$

Table 5: RL training configuration. Outcome advantages use group-relative normalization with mean absolute deviation (Eq. 13).

Hyperparameter	Value
Training GPUs	8
Rollout batch size	32
Samples per prompt	16
Steps per rollout	1
Max prompt length	2,048 tokens
Max response length	16,384 tokens
Temperature / top-p	1.0 / 0.9
Outcome advantage	Group-relative (Eq. 13)
Optimizer	AdamW
Learning rate / schedule	1×10^{-6}, constant

Text Bytes	5	8	2	3	1	2	3	1	2	3		1	5	7
Text	Three	numbers	a	₁	,	a	₂	,	a	₃		.	Each	number
Token ID	19641	5109	264	30986	11	264	31807	11	264	15896	225	13	8886	1372
Token Bytes	5	8	2	3	1	2	3	1	2	3	3	1	5	7

Text Bytes ≠Token Bytes

Figure 3: Isolated-decoding byte-count error for subscript three (`U+2083`). The highlighted token pair represents one three-byte character. Independently decoding the tokens and re-encoding their replacement characters yields six bytes, shown in the "Token Bytes" row. Joint decoding recovers the character and its original three-byte length.

Training hyperparameters. Table 5 summarizes the RL training configuration. The reward in Eq. 18 depends on both answer correctness and turn count. A matched outcome-only comparison retains this reward function and sets the auxiliary distillation coefficient to zero.

B Isolated-Decoding Failure in Byte-Prefix Alignment

Inferring token-byte lengths from isolated decodes can corrupt prefix-sum alignment. Figure 3 illustrates the issue using subscript three (`U+2083`), whose UTF-8 encoding is `E2 82 83`. Consider the schematic split `[E2 82] [83]`. With replacement decoding, each incomplete piece produces `U+FFFD`, whose UTF-8 encoding occupies three bytes. Summing the re-encoded isolated decodes therefore yields six bytes for a character that occupies only three. Until an explicit resynchronization, subsequent prefix sums carry this three-byte offset.

Jointly decoding the original pieces recovers the intended character and its true byte length. DVCA verifies correspondence through clean decoded-span equality. Reading faithful underlying token bytes also preserves the byte count in this example.

C Template-Aware Correspondence Extension

This proposed extension separates interaction-level correspondence from local token alignment for native tool serializations in which corresponding functions or arguments may use different syntax.

C.1 Region Identities and Canonicalization

A model-specific adapter $\mathcal{P}_M$ extracts typed regions and their original token spans. Each region has a shared identity

$$q = (u, c, a, \kappa, o), \tag{19}$$

where u is the assistant turn, c the tool-call index, a the argument path, κ the region type, and o the occurrence within the field or turn. Inapplicable fields have null identifiers. Content types include

reasoning, assistant text, function names, argument keys, and argument values. The occurrence index distinguishes repeated strings that would otherwise have identical keys.

Let I_q^S, I_q^T be the original token spans for an eligible region pair, and let Γ_M^κ denote its declared decoding and canonicalization map. Accepted content chunks satisfy

$$\Gamma_S^{\kappa_q}(\mathbf{s}[S_{q,k}]) = \Gamma_T^{\kappa_q}(\mathbf{t}[T_{q,k}]), \qquad S_{q,k} \subseteq I_q^S, \quad T_{q,k} \subseteq I_q^T. \tag{20}$$

Candidate expansion and recovery remain inside each paired region. For ordinary content matching, Γ_M^κ reduces to the span decoder in Eq. 4. Cross-format pairs may require format-specific escape decoding, with a provenance map back to the original token spans. Canonicalization preserves argument values and executable semantics; Unicode normalization is eligible only when it preserves those semantics.

Each model is scored on its full native prefix. Prefix conversion uses only student content observed up to the corresponding point in the interaction. Cross-format field pairing therefore requires both matching identities and compatible causal order. Tokens crossing a region boundary remain unmatched unless a verified ownership rule assigns each complete sampled action to a single region.

C.2 Control Events and Action Provenance

Generated control markers may be paired by explicit event identity, such as reasoning start/end, tool-call start/end, or assistant-turn termination. Eligible pairs represent the same observed transition. Prompt-inserted markers, template separators, and closures synthesized after truncation belong to the conditioning context and are excluded from sampled action targets. Genuine termination events are retained for every completed assistant turn.

The extended correspondence set is $\mathcal{C}_{\text{text}} \cup \mathcal{C}_{\text{event}}$. Content pairs obey Eq. 20; event pairs obey their declared event rules and are tracked separately from exact text matches. Student spans are non-overlapping. Unmatched spans receive zero auxiliary advantage while retaining any valid outcome-reward supervision. Partial final turns retain their observed prefixes without synthetic completion.

C.3 ALGORITHMS

Algorithm 1 specifies the outer dispatcher. Algorithm 2 applies the same clean-span search locally within each content pair, recording rejected residuals in the unmatched set $\mathcal{U}$. Each adapter specifies its search bounds, recovery rules, canonicalization, and supported events.

Algorithm 1 Template-aware region correspondence for CT-OPD

Require: observed trajectory τ, serializers/tokenizers, region adapters
Ensure: valid units $\mathcal{C}$, unmatched spans $\mathcal{U}$, and action masks
Preserve sampled student IDs and prefixes; construct the teacher's native view
Score both sequences with their full causal contexts
Extract typed regions, original token spans, and action provenance
Pair eligible regions by turn, call, argument path, type, and occurrence
$\mathcal{C} \leftarrow \emptyset$; initialize $\mathcal{U}$ with unpaired or ineligible spans
for each eligible pair (I_q^S, I_q^T, κ_q) **do**
    if κ_q is a control-event type **then**
        if event rule, generation provenance, and span ownership are valid **then**
            Add the event span pair to $\mathcal{C}$
        else
            Record the unmatched pair in $\mathcal{U}$
        end if
    else
        $(\mathcal{C}_q, \mathcal{U}_q) \leftarrow$ local DVCA (Algorithm 2)
        Restore full-sequence offsets; add $\mathcal{C}_q$ to $\mathcal{C}$
        Add unmatched residuals $\mathcal{U}_q$ to $\mathcal{U}$
    end if
end for
Check non-overlap, action masks, and score-index consistency
Apply CRDS only to valid units and broadcast to student positions
Leave unmatched CRDS advantages zero; retain the independent RL mask
return $\mathcal{C}, \mathcal{U}$, and masks

Algorithm 2 Decode-Verified Chunk Alignment within one content-region pair

Require: native token slices $\mathbf{s}_q, \mathbf{t}_q$; typed maps $\Gamma_S^\kappa, \Gamma_T^\kappa$
Ensure: verified local chunks $\mathcal{C}_q$ and unmatched spans $\mathcal{U}_q$
$\mathcal{C}_q \leftarrow \emptyset, \mathcal{U}_q \leftarrow \emptyset, c_s \leftarrow 0, c_t \leftarrow 0$
while $c_s < |\mathbf{s}_q|$ and $c_t < |\mathbf{t}_q|$ **do**
    Decode candidates from c_s, c_t; keep every candidate within this pair
    Apply only the declared typed canonicalization with original-span provenance
    Reject candidates with invalid partial decoding or ambiguous correspondence
    Search within fixed bounds for nonempty spans with equal canonical content
    if a verified pair (c_s, i, c_t, j) is found **then**
        Append (c_s, i, c_t, j) to $\mathcal{C}_q$
        $c_s \leftarrow i, c_t \leftarrow j$
    else
        Attempt bounded local recovery inside this same region pair
        if recovery advances to another candidate boundary **then**
            Record skipped spans in $\mathcal{U}_q$ and advance the cursors
        else
            Record remaining spans in $\mathcal{U}_q$; advance both cursors to the ends
        end if
    end if
end while
if either side has an unmatched suffix **then**
    Record that suffix in $\mathcal{U}_q$, not in $\mathcal{C}_q$
end if
return $\mathcal{C}_q, \mathcal{U}_q$

D CASE STUDY

This example shows the local CRDS feedback along a trajectory with a correct final answer. Red and blue highlights denote negative and positive auxiliary scores, respectively, with qualitative color intensity. The outcome reward evaluates final-answer correctness and combines with CRDS through Eq. 14.

CT-OPD Alignment Case Study

Question. For every three-digit natural number n (where the leading digit of n is nonzero), consider the number n_0 obtained by eliminating all zero digits from n. For example, if $n = 207$, then $n_0 = 27$. Determine the number of three-digit positive integers n for which n_0 is a divisor of n and $n_0 \neq n$.

Ground Truth: 93 **Student Answer:** 93 ✓ **Correct:** Yes

Color legend. CRDS < 0: penalty CRDS > 0: reward

Assistant turn 0.

Alright, let's tackle this problem step by step. The goal is to find the number of three-digit natural numbers n such that when we remove all zero digits from n to get n_0, n_0 is a divisor of n, and $n_0 \neq n$.

First, let's understand the problem with an example. If $n = 207$, then $n_0 = 27$. We need to check if 27 divides 207. Since $207 \div 27 = 7.666\ldots$, which is not an integer, 207 is not divisible by 27. So, 207 would not be counted.

To systematically solve this, we'll generate all three-digit numbers n from 100 to 999, compute n_0 by removing all zero digits, and then check if n_0 divides n and $n_0 \neq n$.

Let's use code to automate this process :

Tool Call: code_interpreter

```
count = 0
for n in range(100, 1000):
    s = str(n)
    n0 = ''.join([digit for digit in s if digit != '0'])
    n0 = int(n0) if n0 else 0
    if n0 != n and n % n0 == 0:
        count += 1
print(count)
```

Tool Output

```
93
```

Assistant turn 1.

The code iterates through all three-digit numbers, computes n_0 by removing zeros, checks if n_0 divides n, and ensures $n_0 \neq n$. The result is 93, which matches the manual verification for specific cases and the code's efficiency.

Therefore, the number of such three-digit numbers n is 93.

Answer: \boxed{93}

Independent answer check. Every three-digit number ending in zero satisfies the condition, giving 90 cases. The remaining possibilities have the form $n = 100a + b$ with $a, b \in \{1, \ldots, 9\}$ and $n_0 = 10a + b$. Since $n = 10n_0 - 9b$, divisibility is equivalent to $10a + b \mid 9b$. Checking these digit pairs yields only 105, 108, and 405, for a total of $90 + 3 = 93$.

DISCLOSURE OF AI USE

Generative AI tools were used primarily to assist with drafting, language polishing, and organization of the manuscript.The authors are responsible for verifying all AI-assisted content and take full responsibility for the final manuscript, including its claims, references, and reported results.