SIMPLEOPD: SIMPLE TOKENIZER-AGNOSTIC ON-POLICY DISTILLATION FOR LONG-CONTEXT REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) offers a promising way to transfer reasoning capabilities from stronger teacher models, but applying it to long-context reasoning teachers and short-context students introduces practical challenges, including tokenizer mismatch, teacher-student distribution mismatch, response length explosion, and training instability. In this work, we study this setting by transferring proof-reasoning capabilities from the long-context reasoning model to short-context student models. To handle tokenizer differences, we perform OPD in a shared text space and align only tokens that occupy identical text spans under the student and teacher tokenizers. To mitigate the problem of excessive generation length and frequent truncation, we introduce a student reference KL loss and mask the advantages of special termination tokens. This strategy constrains the student from drifting excessively from its initial policy, thereby mitigating the teacher-student distribution mismatch problem and fostering steady length growth. Experiments on both same-family and different-family student models, including Qwen3, Qwen3.5, Intern-S2, GLM-4.7, Gemma-4, show consistent gains in mathematical reasoning, especially natural-language math proving. Notably, Intern-S2-Preview improves by 21.2 points on ProofBench, reaching 55.2 and surpassing Gemini-2.5-Pro. It also improves on science benchmarks such as HLE and HiPhO, suggesting that OPD transfers reasoning capabilities that generalize beyond the mathematical training domain.

1 INTRODUCTION

On-policy distillation (OPD) (Gu et al., 2024; Agarwal et al., 2024) presents a compelling paradigm in the post-training landscape of LLMs for tasks such as strong-to-weak distillation (Lu & Lab, 2025), multi-teacher distillation (Ma et al., 2026), and self-distillation (Hübotter et al., 2026). Unlike off-policy approaches that distill knowledge from fixed teacher-generated data, OPD actively queries the teacher to evaluate tokens in student-generated trajectories, thereby providing dense token-level supervision grounded in the student's own policy (Gu et al., 2024; Agarwal et al., 2024). It enables efficient distillation of complex reasoning behaviors, as it incurs less forgetting and achieves better generalization than off-policy settings. However, prior work has primarily studied teacher–student pairs that share the same model family and vocabulary. Distilling reasoning capabilities from long-context teachers into short-context students, especially across different model families, remains largely underexplored (Sun et al., 2026).

Applying OPD to distilling long-context reasoning teachers to short-context reasoning students presents several challenges. First, teacher and student models may use different tokenizers, making direct alignment between their token-level distributions infeasible (Zhang et al., 2024; Chen et al., 2025b; Singh et al., 2026). Second, differences in model capacity and context length introduce a substantial teacher–student distribution mismatch (Li et al., 2026b). In particular, long-context reasoning teachers often favor responses that exceed the student's distillation budget. Through experiments, we observe that direct OPD can cause rapid response-length growth, frequent truncation, and incomplete reasoning trajectories, ultimately destabilizing training. This issue is especially pronounced for termination tokens such as $<\!/\mathrm{think}\!>$ and $<\!|\mathrm{im_end}|\!>$, as teacher supervision may repeatedly discourage the student from terminating its thinking process or final response.

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.

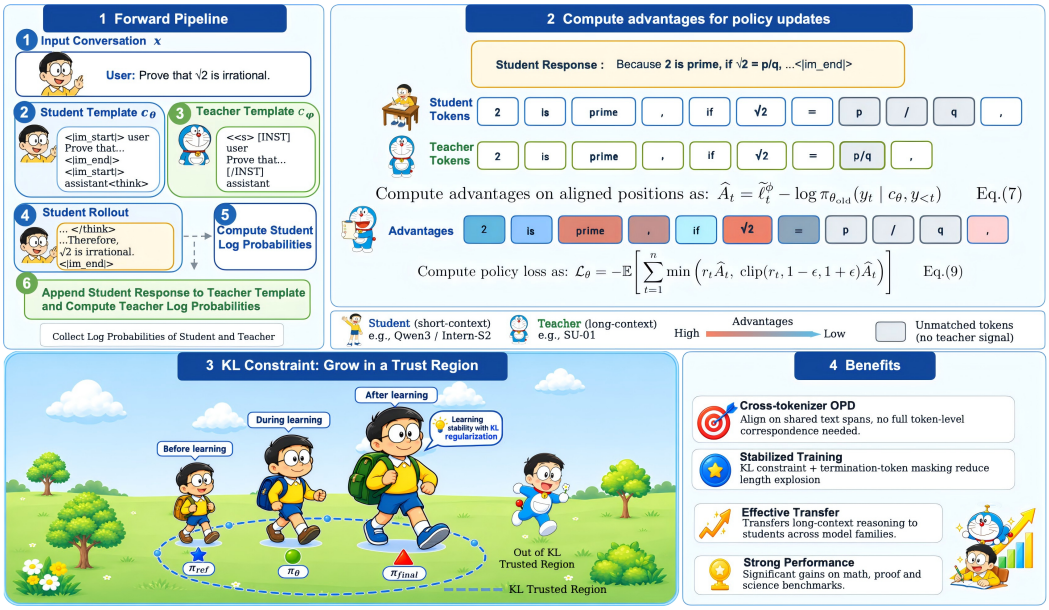

Figure 1: Overview of SimpleOPD. The student generates responses under its own tokenizer, while the teacher evaluates the same response under its native tokenizer. We align tokens with identical text spans, use teacher supervision for matched tokens, and stabilize training with KL regularization and termination-token masking.

In this work, we focus on the transfer of reasoning capabilities from the long-context reasoning model SU-01 (Li et al., 2026a) to short-context student models. As shown in Figure 1, to enable distillation across different model families, we perform OPD in a shared text space and align teacher and student tokens only when they occupy identical text spans. This provides reliable token-level supervision without requiring an artificial correspondence between incompatible tokenizations. To mitigate excessive response-length growth and stabilize optimization, we introduce a reference KL-divergence loss between the student and its initial policy. We also mask the advantages of special termination tokens, preventing teacher supervision from directly suppressing termination. Together, these techniques constrain excessive policy drift, alleviate teacher–student distribution mismatch, and promote steady response-length growth within the student's context budget.

We evaluate SimpleOPD across both same-family and cross-family distillation settings, using student models from Qwen3, Qwen3.5, Intern-S2-Preview, GLM-4.7, and Gemma-4 (Yang et al., 2025; Qwen Team, 2026; InternLM Team, 2026; Z.ai, 2026; Gemma Team, 2026). Across model families, SimpleOPD consistently improves mathematical reasoning performance, with particularly strong gains on natural-language mathematical proof tasks. Most notably, Intern-S2-Preview improves by 21.2 points on ProofBench (Luong et al., 2025), from 34.0 to 55.2, surpassing Gemini-2.5-Pro (Figure 7). The distilled students also achieve improvements on science-oriented benchmarks, including HLE and HiPhO (Phan et al., 2025; Yu et al., 2025), despite being trained exclusively on mathematical reasoning data. These results demonstrate that stable cross-tokenizer OPD can transfer reasoning capabilities from long-context teachers, and that the acquired capabilities generalize beyond the training domain. In summary, our observations and contributions are:

- We introduce SIMPLEOPD, a cross-tokenizer on-policy distillation framework that operates in a shared text space, i.e., aligns only teacher–student tokens with identical text spans, avoiding the need for vocabulary-level token alignment.
- We develop a stabilization strategy for distilling long-context reasoning into short-context students, including student-reference KL regularization and termination-token advantage masking to mitigate response-length explosion, truncation, and training instability.
- We demonstrate that SIMPLEOPD consistently improves mathematical reasoning across both same-family and cross-family students, such as Qwen3, Qwen3.5, Intern-S2, GLM-4.7, and Gemma-4. The resulting gains are particularly strong on natural-language proof tasks and transfer beyond mathematics to scientific reasoning benchmarks.

2 METHOD

Let $\mathbf{x}$ denote the input conversation represented as a list of messages. The student and teacher with different tokenizers may use different chat templates, denoted by $\mathcal{C}_\theta$ and $\mathcal{C}_\phi$, respectively. We first construct the student input context text $c_\theta = \mathcal{C}_\theta(\mathbf{x})$ and sample a response-token sequence with the student policy executed by the rollout engine $\pi_{\theta_{\text{roll}}}$:

$$y_{1:n} = (y_1, \ldots, y_n) \sim \pi_{\theta_{\text{roll}}}(\cdot \mid c_\theta). \tag{1}$$

Decoding $y_{1:n}$ via the tokenizer decoder of the student gives the response surface string $s = \mathcal{D}_\theta(y_{1:n})$. Instead of passing the student context to the teacher, we reconstruct the teacher context text c_ϕ using its own chat template and append the student response:

$$c_\phi = \mathcal{C}_\phi(\mathbf{x}), \quad u_\phi = c_\phi \oplus s,$$

where $\oplus$ denotes string concatenation. The complete text u_ϕ is then provided to the teacher, which tokenizes it using the tokenizer encoder $\mathcal{E}_\phi$of the teacher. Let $\mathcal{E}_\phi(s|c_\phi) = (z_1, \ldots, z_m)$ denote token list obtained by encoding the student response given the input context $c_\phi$using the teacher tokenizer. This procedure allows the teacher to evaluate the student response under its native tokenizer and chat-template. Although c_θ and c_ϕ may differ, the response text being evaluated is identical on both sides. For token alignment, let $\tau_\theta(y_t)$ and $\tau_\phi(z_i)$ denote the incremental text spans contributed by the corresponding response tokens. They satisfy

$$\bigoplus_{t=1}^{n} \tau_\theta(y_t) = \bigoplus_{i=1}^{m} \tau_\phi(z_i) = s.$$

We perform no additional cleanup or normalization, so these spans define consistent offsets in the shared response string. The teacher log-probability associated with token z_i is $\log \pi_\phi(z_i \mid c_\phi, z_{<i})$.

2.1 CROSS-TOKENIZER ALIGNMENT

Define the cumulative response text preceding each student and teacher token as

$$P_\theta(t) = \bigoplus_{k=1}^{t-1} \tau_\theta(y_k), \qquad P_\phi(i) = \bigoplus_{k=1}^{i-1} \tau_\phi(z_k),$$

with $P_\theta(1) = P_\phi(1) = \varepsilon$, where ε denotes the empty string. A teacher position i is aligned with a student position t if both tokenizations have consumed the same response prefix and the current tokens contribute the same text span:

$$\mathcal{M} = \{(i, t) : P_\phi(i) = P_\theta(t) \wedge \tau_\phi(z_i) = \tau_\theta(y_t)\}. \tag{2}$$

Equivalently, an aligned teacher-student pair covers the same start and end offsets in the shared response string s. Tokens that overlap only partially are not aligned because the log-probability of one teacher token cannot be uniquely assigned to multiple student tokens, nor can the log-probabilities of multiple teacher tokens be uniquely merged into a single student token. Since y and z are ordered segmentations of the same response string, $\mathcal{M}$ is a partial one-to-one mapping: each token on either side can match at most one token on the other side. We enumerate $\mathcal{M}$ using a linear two-pointer scan. The scan maintains the accumulated response prefixes on both sides. If the prefixes and current token spans agree, the two tokens are aligned and both pointers advance. Otherwise, the side that has consumed less response text advances until the two prefixes meet again. When both sides have consumed the same amount of text but the current token spans differ, both pointers advance.

For each student position t, define the alignment indicator

$$a_t = \mathbf{1}[\exists i \text{ such that } (i, t) \in \mathcal{M}]. \tag{3}$$

Let $\log \pi_\theta(y_t \mid c_\theta, y_{<t})$ and $\log \pi_\phi(z_i \mid c_\phi, z_{<i})$ denote the log-probabilities of student policy and teacher policy, respectively. We construct a student-length teacher target as

$$\widetilde{\ell}_t^{\phi} = \begin{cases} \log \pi_\phi(z_i \mid c_\phi, z_{<i}), & a_t = 1, \\ \log \pi_\theta(y_t \mid c_\theta, y_{<t}), & a_t = 0. \end{cases} \tag{4}$$

Thus, aligned positions inherit the corresponding teacher log-probability, while unmatched positions fall back to the student's log-probability. We report the lexical overlap ratio $\rho = \frac{|\mathcal{M}|}{n}$, which measures the fraction of student response tokens that receive teacher supervision. The complete alignment procedure is provided in Algorithm 1 in Appendix B.

2.2 ON-POLICY DISTILLATION OBJECTIVE

The cross-tokenizer distillation objective is defined over aligned response positions:

$$\mathcal{L}_{\text{Distill}}(\theta) = \mathbb{E}_{y\sim\pi_\theta}\left[\sum_{t=1}^{n} \log \pi_\theta(y_t \mid c_\theta, y_{<t}) - \widetilde{\ell}_t^{\phi}\right]. \tag{5}$$

This objective is a token-aligned surrogate for the reverse KL divergence. It compares the teacher and student probabilities only at positions where the two tokenizers induce the same local segmentation of the response string. When the tokenizers are identical, every student token is aligned with the corresponding teacher token, so $a_t = 1$ and $z_t = y_t$ for all t. The objective then reduces to

$$\mathcal{L}_{\text{Distill}}(\theta) = \mathbb{E}_{y\sim\pi_\theta}\left[\log \frac{\pi_\theta(y \mid c_\theta)}{\pi_\phi(y \mid c_\phi)}\right] = D_{\text{KL}}\left(\pi_\theta(\cdot \mid c_\theta) \,\|\, \pi_\phi(\cdot \mid c_\phi)\right). \tag{6}$$

When the tokenizers differ, the same response string is factorized into different token sequences, so the exact token-level KL is not directly computable. The proposed objective instead applies teacher supervision only where the two tokenizations agree on both the token boundary and textual span. To enable multiple policy updates on the same rollout batch, we substitute the online log-probability $\log \pi_\theta(y_t \mid c_\theta, y_{<t})$ in $\widetilde{\ell}_t^{\phi}$ with its pre-update counterpart $\log \pi_{\theta_{\text{old}}}(y_t \mid c_\theta, y_{<t})$ for all unmatched positions ($a_t = 0$) and define the fixed policy advantages as $\widehat{A}_t = \widetilde{\ell}_t^{\phi} - \log \pi_{\theta_{\text{old}}}(y_t \mid c_\theta, y_{<t})$, using the PPO clipped policy loss. The importance-sampling ratio is

$$r_t = \frac{\pi_\theta(y_t \mid c_\theta, y_{<t})}{\pi_{\theta_{\text{old}}}(y_t \mid c_\theta, y_{<t})}. \tag{7}$$

The resulting objective is

$$\mathcal{L}_\theta = -\mathbb{E}\left[\sum_{t=1}^{n} \min\left(r_t \widehat{A}_t,\ \text{clip}(r_t, 1-\epsilon, 1+\epsilon)\widehat{A}_t\right)\right]. \tag{8}$$

3 EXPERIMENT

3.1 EXPERIMENT SETTING

We adopt SU-01 (Li et al., 2026a) serving as the teacher model, a 30B-A3B reasoning model based on Qwen3-30B-A3B (Yang et al., 2025) with strong mathematical proof capabilities. It achieves gold-medal-level performance on olympiad-level evaluations and demonstrates particularly competitive results on proof-oriented benchmarks such as ProofBench (Luong & Lockhart, 2025). It is capable of sustaining over 100K tokens of natural-language reasoning for difficult Olympiad problems, making it a powerful yet highly long-context teacher. Our goal is to transfer its mathematical reasoning and proof-generation capabilities to a diverse set of student models through on-policy distillation.

We consider both same-vocabulary and cross-tokenizer settings. For the same-tokenizer setting, we distill SU-01 into models from the Qwen3 family, including Qwen3-4B-thinking-2507 and Qwen3-30B-A3B-Thinking-2507 (abbreviated as Qwen3-4B and Qwen3-30B-A3B, respectively). For the cross-tokenizer setting, we further evaluate models with different tokenizer designs, including Qwen3.5-series (Qwen Team, 2026), Intern-S2-Preview (InternLM Team, 2026), GLM-4.7-Flash (Z.ai, 2026), and Gemma-4-26B-A4B-it (Gemma Team, 2026) (abbreviated as Gemma-4-26B-A4B). This setting allows us to examine whether cross-tokenizer OPD can transfer mathematical proof capabilities across different model architectures and tokenization schemes.

3.2 TRAINING DATA

All training examples are mathematical proof problems. The collection combines curated proof corpora, community-contributed problems, and olympiad training materials to provide a varied set of proof-oriented prompts. Our training set consists of: (1) **OPC**: 63 problems from the Open Proof Corpus; (2) **AoPS**: 2,948 problems collected from the Art of Problem Solving community; (3) **Books**: 900 problems collected from online mathematical competition training books; (4) **Shuzhimi**: 617 problems sourced from the Shuzhimi Forum, an online Chinese mathematical problem-solving community, and Evan Chen's olympiad materials.

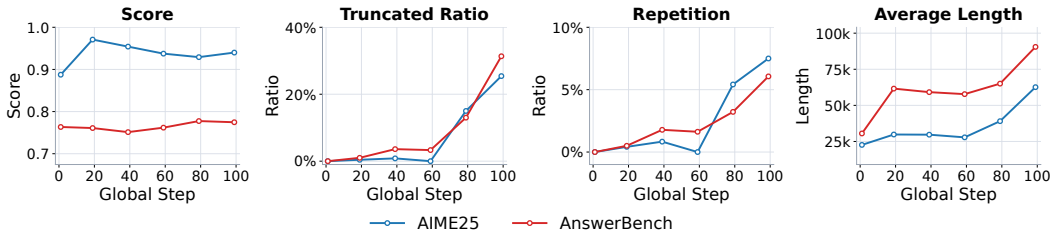

Figure 2: Training dynamics of Intern-S2-Preview during direct OPD training.

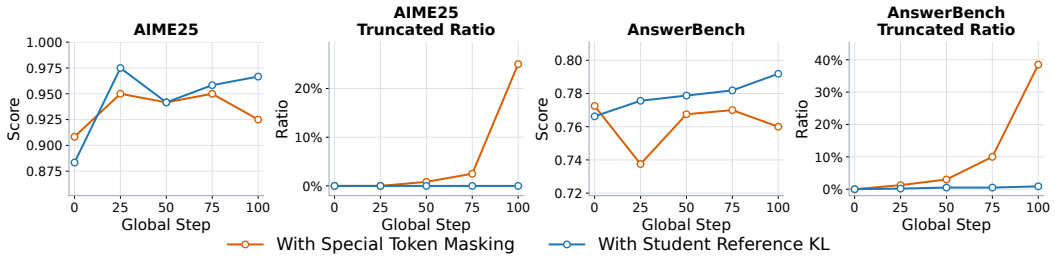

Figure 3: Intern-S2-Preview OPD training dynamics with masking the special tokens such as $<\!/\text{think}\!>$ and $<\!|\text{im_end}|\!>$, and with student reference KL loss.

3.3 Implementation Details

We use Slime (Zhu et al., 2025) as the underlying framework for OPD and SGLang (Zheng et al., 2024) for evaluation rollout. The models are trained for 100 rollout iterations with a constant learning rate of 1e-6, a rollout batch size of 64, 4 responses sampled per prompt, and a maximum rollout length of 32K tokens for Qwen-Series, 6k for GLM-4.7 and Gemma-4. The policy objective uses a PPO clipping coefficient of 0.2. The policy is updated four times per rollout step.

In our evaluation, we primarily evaluate the non-verifiable ProofBench and the verifiable AnswerBench from the IMO-Bench suite (Luong et al., 2025), together with the verifiable benchmarks AIME25 (Mathematical Association of America, 2025) and AMOBench (An et al., 2025). For ProofBench, we use DeepSeek-V4-Flash as the judge for cost-efficient evaluation. To mitigate evaluation instability, we evaluate 4 times for the same proof rollout and choose the average result. For the latter three, we first use a rule-based verifier for testing; if the answer is correct, it is counted as correct, and if it is wrong, we then use GPT-OSS-120B (OpenAI et al., 2025) for evaluation. The reported results are the averages over 4 rollouts for ProofBench and 8 for the other three. We use the following evaluation settings: temperature 1.0, top-p 0.95, repetition penalty 1.0, and a maximum response length of 160,000 tokens. The best checkpoint is selected based on the average score of AIME@4 and AnswerBench@1.

4 Experimental Results

4.1 Training Instability in On-Policy Distillation

Initial Observation: Length Explosion and Truncation. During OPD training, the student Intern-S2-Preview shows modest improvements in task performance, but these gains are accompanied by clear signs of degeneration (Figure 2). Specifically, the truncation rate and repetition rate both increase substantially as training progresses, while the average response length also grows sharply. This phenomenon is even more pronounced for Qwen3.5-35B-A3B (Figure 6), as its task performance deteriorates more noticeably, and the degradation is accompanied by a persistently high truncation rate. These results suggest that apparent performance gains can come at the cost of increasingly verbose and unstable generation behavior, revealing a trade-off between task performance and output quality. Case studies could be found in Appendix D.

Table 1: Evaluation results of distilling SU-01 to students on ProofBench, AnswerBench, AIME25, and AMOBench. Green values indicate improvements over the corresponding base model.

Model	ProofBench@4	AnswerBench@8	AIME25@8	AMOBench@8
SU-01	45.00	77.50	94.60	61.75
Same-Tokenizer Models				
Qwen3-4B	11.42	47.50	71.25	23.00
Qwen3-4B-OPD	**23.72** (+Δ 12.30)	**64.50** (+Δ 17.00)	**90.83** (+Δ 19.58)	**35.00** (+Δ 12.00)
Qwen3-30B-A3B	13.80	59.13	88.33	36.50
Qwen3-30B-A3B-OPD	**36.47** (+Δ 22.67)	**74.46** (+Δ 15.33)	**93.75** (+Δ 5.42)	**52.75** (+Δ 16.25)
Cross-tokenizer Models				
Qwen3.5-4B	15.90	60.94	86.67	32.00
Qwen3.5-4B-OPD	**28.61** (+Δ 12.71)	**67.84** (+Δ 6.90)	**91.67** (+Δ 5.00)	**51.25** (+Δ 19.25)
Qwen3.5-35B-A3B	26.78	73.16	94.60	57.25
Qwen3.5-35B-A3B-OPD	**42.39** (+Δ 15.61)	**80.15** (+Δ 6.99)	**96.66** (+Δ 2.06)	**61.25** (+Δ 4.00)
Intern-S2-Preview	21.70	76.03	88.33	58.00
Intern-S2-OPD	**44.50** (+Δ 22.80)	**80.10** (+Δ 4.07)	**95.00** (+Δ 6.67)	**59.50** (+Δ 1.50)

Special Token Masking. During training, we observe that the student model's output length keeps increasing, and many generated responses fail to emit termination tokens such as $<|\text{im_end}|>$. This suggests that the student is strongly affected by the long outputs from the teacher and gradually loses the ability to terminate properly. To mitigate this issue, we mask the OPD loss on the structural tokens $</\text{think}>$ and $<|\text{im_end}|>$, as these tokens primarily control the output format and termination behavior and do not need to match the teacher distribution exactly. The training curve is shown in Figure 3 (orange). The results suggest that the masking strategy helps mitigate length-related instability. However, in the later stage of training, the model's length truncation rate still continues to increase sharply, indicating that special-token masking alone cannot fully resolve the length expansion problem.

Student-Reference KL Loss. We then attempt to introduce a student reference KL loss to prevent the student policy from deviating excessively from its initial distribution, thereby preserving the student model's general capabilities during distillation. In this experiment, we set the student-reference kl-loss coefficient to 0.5. The training curve is shown in Figure 3 (blue). As observed, the truncation rate is effectively reduced to nearly zero, while performance on AIME25 and AnswerBench is consistently improved. This indicates that the student reference KL loss can effectively mitigate the dramatic increase in response length during OPD training. The evaluation results are summarized in Appendix Table 5. Specifically, OPD + Ref KL improves ProofBench@4 from 21.70 to 38.50, and further raises AnswerBench@8 and AIME25@8 to 79.10 and 95.80, respectively. The results demonstrate that adding the reference KL loss substantially improves OPD training while preserving the student model's original capabilities.

4.2 Results on Same-family Models

Finally, we combine special token masking with the student reference KL loss, which yields larger performance improvements. We set kl-loss-coef to 0.5 for the Qwen-series models and Intern-S2-Preview. The experimental results are shown in Table 1. Overall, SimpleOPD consistently improves performance across model sizes, tokenizer settings, and model families. Among the same-tokenizer models, Qwen3-4B-OPD gains 12.30 points on ProofBench, 17.00 points on AnswerBench, and 19.58 points on AIME25. Scaling the student to Qwen3-30B-A3B produces even greater proof-reasoning improvements; its ProofBench score increases by 22.67 points, from 13.80 to 36.47, while its AnswerBench and AIME25 scores improve by 15.33 and 5.42 points, respectively.

The improvements also generalize to cross-tokenizer distillation. Qwen3.5-35B-A3B-OPD reaches 42.39 on ProofBench, improving over its base model by 15.61 points, and achieves additional gains of 6.99 and 2.06 points on AnswerBench and AIME25. Intern-S2-Preview exhibits the largest Proof-

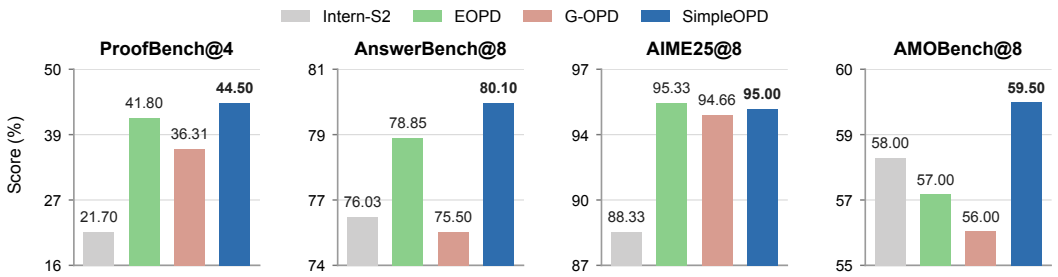

Figure 5: Comparison of the Intern-S2-Preview model distilled from SU-01 with EOPD, G-OPD, and SimpleOPD across four reasoning benchmarks.

Bench gain, improving by 22.80 points from 21.70 to 44.50, nearly matching the teacher's score of 45.00. It also reaches 80.10 on AnswerBench and 95.00 on AIME25, surpassing SU-01 on both benchmarks. We also evaluate ProofBench using Gemini-2.5-Pro as the judge, following the same evaluation setting as SU-01 (Figure 7). Specifically, Intern-S2-OPD improves from 34.0 to 55.2, achieving a gain of 21.2, surpassing Gemini-2.5-Pro. These results demonstrate that our approach effectively transfers proof-reasoning capabilities to diverse students without requiring shared tokenization or supervised fine-tuning on teacher-generated trajectories.

4.3 Results on Cross-family Models

Figure 4 presents additional cross-family distillation results on GLM-4.7-Flash and Gemma-4-26B-A4B. GLM-4.7-OPD improves consistently on both benchmarks, increasing from 30.8 to 39.7 on ProofBench and from 69.6 to 72.0 on AnswerBench. Gemma-4-26B-A4B-OPD also achieves a substantial ProofBench gain, rising from 25.5 to 34.2, although its AnswerBench score decreases from 68.8 to 67.5.

These results demonstrate the effectiveness of our method for cross-family distillation, while also highlighting the challenge posed by tokenizer mismatch. SU-01 is based on the Qwen model family and uses a byte-level BPE tokenizer. GLM-4.7-Flash also adopts a BPE-based tokenizer, although with a different vocabulary, whereas Gemma uses a SentencePiece-based tokenizer with substantially different vocabulary and segmentation behavior. Accordingly, GLM benefits consistently across both benchmarks, while the more tokenizer-dissimilar Gemma merely improves on ProofBench. Nevertheless, the clear ProofBench improvements for both models demonstrate that our approach can transfer proof-reasoning capabilities across model families and tokenizer boundaries, although larger tokenizer discrepancies may make such transfer more challenging.

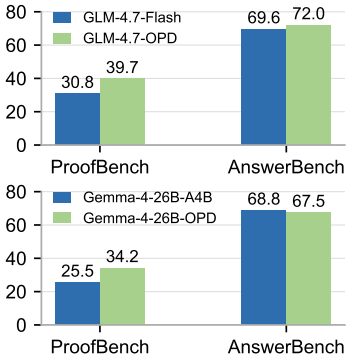

Figure 4: Cross-family distillation results for GLM-4.7-Flash and Gemma-4-26B-A4B from SU-01.

5 Analysis

5.1 Comparison with OPD Baselines

We compare SimpleOPD with two model: Intern-S2-Preview and Qwen3.5-35B-A3B and two recent OPD variants: EOPD, which supplements reverse-KL OPD with forward KL at positions where the teacher has high token-level entropy (Jin et al., 2026), and G-OPD, which generalizes OPD through a flexible reference model and reward scaling (Yang et al., 2026) (the results on Qwen3.5-35B-A3B are shown in Appendix A.7). As shown in Figure 5, SimpleOPD obtains 44.50 on ProofBench, 80.10 on AnswerBench, and 59.50 on AMOBench, outperforming both EOPD and G-OPD by a large margin. On AIME25, SimpleOPD reaches 95.00, which is 0.33 points below EOPD and 0.34 points above G-OPD. Overall, SimpleOPD achieves the best result on three of the four bench-

Table 2: Evaluations of Intern-S2-OPD distilled from SU-01 on scientific reasoning benchmarks.

Model	FrontierScience Olympiad	FrontierScience Research	HLE (text-only)	HiPhO
SU-01	61.5	11.7	20.7	35.0
Intern-S2-Preview	60.6	1.7	19.6	38.6
Intern-S2-OPD	**60.9** (+Δ 0.3)	**5.0** (+Δ 3.3)	**20.5** (+Δ 0.9)	**41.1** (+Δ 2.5)

Table 3: Results of different distillation lengths on Intern-S2-Preview and Qwen3.5-35B-A3B distilled from the teacher SU-01.

Model	Length	ProofBench@4	AnswerBench@8	AIME25@8
SU-01	–	45.00	77.50	94.60
Intern-S2-Preview	–	21.70	76.03	88.33
Qwen3.5-35B-A3B	–	26.78	73.16	94.60
Intern-S2-OPD	6k	38.80 (+Δ 17.10)	77.25 (+Δ 1.22)	95.00 (+Δ 6.67)
	32k	**44.50** (+Δ 22.80)	**80.10** (+Δ 4.07)	**95.00** (+Δ 6.67)
Qwen3.5-35B-A3B-OPD	6k	40.07 (+Δ 13.29)	77.97 (+Δ 4.81)	96.25 (+Δ 1.65)
	32k	**42.39** (+Δ 15.61)	**80.16** (+Δ 7.00)	**96.67** (+Δ 2.07)

marks, with its largest advantage appearing on the proof-oriented ProofBench, while remaining competitive on AIME25.

5.2 Out-of-domain Generalization

Although OPD training is conducted only on mathematical reasoning data, the resulting model also shows consistent gains on out-of-domain scientific reasoning benchmarks. The results are shown in Table 2. Compared with Intern-S2-Preview, Intern-S2-OPD achieves a slight improvement on FrontierScience-Olympiad and HLE (Wang et al., 2026; Phan et al., 2025), and a clearer gain on HiPhO (Yu et al., 2025), improving from 38.6 to 41.1. On FrontierScience Research (Wang et al., 2026), the score also improves from 1.7 to 5.0. This suggests that the OPD-trained model does not lose its broader scientific reasoning ability despite being trained only on math data, and may further benefit physics-oriented reasoning. In particular, Intern-S2-OPD outperforms SU-01 on HiPhO, indicating that the student model retains its own strengths while gaining additional reasoning capability through SimpleOPD.

5.3 Effect of OPD Length

We also analyze the effect of the distillation length(Table 3). By comparison, we find that for long-context reasoning tasks, a 6k distillation length is insufficient to fully capture the teacher model's reasoning patterns. Specifically, increasing the distillation length from 6k to 32k improves the student model consistently across all evaluated benchmarks when the distillation is stable. In Qwen3.5-OPD, ProofBench@4 increases from 40.07 to 42.39, AnswerBench@8 from 77.97 to 80.16, and AIME25@8 from 96.25 to 96.67. The improvement is most pronounced on ProofBench, suggesting that longer distillation sequences are particularly important for transferring complex, multi-step proof reasoning behavior. These results indicate that preserving longer reasoning traces during distillation helps the student model better approximate the teacher's long-context reasoning capability.

5.4 Distillation from DeepSeek-V4-Flash

Table 4 reports the results of distilling DeepSeek-V4-Flash (DeepSeek-AI, 2026) into Intern-S2-Preview using a 6K distillation length for computational efficiency. Despite the shorter distillation length, Intern-S2-DS-OPD consistently outperforms its base model across all three benchmarks. Its ProofBench@4 score increases from 21.70 to 39.71, corresponding to a gain of 18.01 points. AnswerBench@8 improves from 76.03 to 77.94, while AIME25@8 increases from 88.33 to 97.50, yielding gains of 1.91 and 9.17 points, respectively. Moreover, its ProofBench@4 score of 39.71 is

Table 4: SimpleOPD results from DeepSeek-V4-Flash to Intern-S2-Preview. The distillation length is 6k for computational efficiency.

Model	ProofBench@4	AnswerBench@8	AIME25@8
DeepSeek-V4-Flash	52.18	77.00	99.17
Intern-S2-Preview	21.70	76.03	88.33
Intern-S2-DS-OPD	**39.71** (+Δ 18.01)	**77.94** (+Δ 1.91)	**97.50** (+Δ 9.17)

0.91 points higher than the 38.80 achieved using SU-01 as the teacher with the same 6K distillation length in Table 3, suggesting that a stronger teacher can lead to greater improvements in the student. Overall, these results demonstrate that SimpleOPD can effectively transfer reasoning capabilities across distinct model families and from a substantially larger teacher (158B) to a smaller student (30B), even with a relatively short distillation length.

6 Related Work

Olympiad-Level Long-Context Reasoning Early works such as OpenAI o1 and DeepSeek-R1 established long chain-of-thought reasoning as a central test-time scaling paradigm (OpenAI, 2024; DeepSeek-AI et al., 2025). Recent studies have further advanced reasoning capabilities in mathematics and science. Gemini Deep Think was among the first systems to achieve gold-medal-level performance at the IMO (Luong & Lockhart, 2025). Nemotron-Math-v2 and MiniMax subsequently reported similarly strong results on mathematical Olympiad benchmarks (Du et al., 2025; Chen et al., 2026). Beyond mathematics, P1 and P1-VL achieved gold-medal-level performance across several physics Olympiads, while SU-01 introduced a unified recipe for eliciting Olympiad-level proof behavior from a 30B model (Chen et al., 2025a; Luo et al., 2026; Li et al., 2026a). However, these capabilities typically rely on costly SFT and RL pipelines, including long-trajectory curation, repeated data-selection rollouts, and RL training, whose cost grows substantially with reasoning length. This motivates our exploration of on-policy distillation for efficiently transferring Olympiad-level proof capabilities across diverse models.

On-policy Distillation. Recently, OPD, introduced by MiniLLM and GKD (Gu et al., 2024; Agarwal et al., 2024), has emerged as an important component of LLM post-training and has been adopted by flagship models (GLM-5 Team, 2026; DeepSeek-AI, 2026; Qwen Team, 2026). By optimizing the reverse KL divergence between teacher and student distributions over student-generated rollouts, OPD combines on-policy exploration with dense token-level supervision, offering a stable and generalizable alternative to conventional SFT and RL (Gu et al., 2024; Agarwal et al., 2024). However, its effectiveness can be limited by mismatched reasoning patterns between teacher and student and unreliable supervision over long sequences (Agarwal et al., 2024; Lei et al., 2026). Moreover, most existing studies assume a shared vocabulary, typically distilling models within the same family or domain-specific RL teachers (Gu et al., 2024; Agarwal et al., 2024; DeepSeek-AI, 2026), while cross-tokenizer alignment has been explored in other settings (Zhang et al., 2025; 2026). However, OPD between heterogeneous models with different architectures and vocabularies in long-context reasoning remains relatively underexplored. In this work, we investigate heterogeneous OPD for transferring long-context reasoning capabilities across diverse models.

7 Conclusion

In this work, we studied on-policy distillation from a long-context reasoning teacher to short-context students. To handle tokenizer differences, we performed OPD in a shared text space, aligning only tokens with identical text spans. To stabilize training, we introduced termination-token masking and a student-reference KL loss, which together substantially reduced length explosion and truncation while improving performance. Experiments on both same-family and cross-family students showed consistent gains in mathematical reasoning and out-of-domain evaluations, demonstrating that SimpleOPD is an effective and generalizable way to transfer long-context reasoning capabilities across tokenizers and model families.

8 AI DISCLOSURE

In this work, we used generative AI tools for implementing methods (specifically, assisting in the development of algorithm code). We have not used generative AI tools for any other tasks requiring disclosure, and the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for editing the manuscript to improve readability and for creating/editing software code for paper figures. We have reviewed all AI-assisted work. For example, AI-assisted manuscript edits were reviewed and revised by the authors; AI-assisted algorithm code was verified and tested for correctness by the authors; and AI-assisted figure-generation code was executed and checked by the authors to ensure that the figures accurately represent the results. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Shengnan An, Xunliang Cai, Xuezhi Cao, Xiaoyu Li, Yehao Lin, Junlin Liu, Xinxuan Lv, Dan Ma, Xuanlin Wang, Ziwen Wang, and Shuang Zhou. AMO-Bench: Large language models still struggle in high school math competitions. *arXiv preprint arXiv:2510.26768*, 2025. URL https://arxiv.org/abs/2510.26768.

Jiacheng Chen, Qianjia Cheng, Fangchen Yu, Haiyuan Wan, Yuchen Zhang, Shenghe Zheng, Junchi Yao, Qingyang Zhang, Haonan He, Yun Luo, et al. P1: Mastering physics olympiads with reinforcement learning. *arXiv preprint arXiv:2511.13612*, 2025a. URL https://arxiv.org/abs/2511.13612.

Jiacheng Chen, Xinyu Zhang, Shunkai Zhang, Yanmohan Wang, Lin Li, Tiancheng Qin, Qin Wang, Zhengmao Zhu, Tianle Li, Jingyang Li, et al. MaxProof: Scaling mathematical proof with generative-verifier RL and population-level test-time scaling. *arXiv preprint arXiv:2606.13473*, 2026. URL https://arxiv.org/abs/2606.13473.

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, Vienna, Austria, 2025b. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.419. URL https://aclanthology.org/2025.findings-acl.419/.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/blob/main/DeepSeek_V4.pdf. Technical report.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Nature*, 645:633–638, 2025. URL https://arxiv.org/abs/2501.12948.

Wei Du, Shubham Toshniwal, Branislav Kisacanin, Sadegh Mahdavi, Ivan Moshkov, George Armstrong, Stephen Ge, Edgar Minasyan, Feng Chen, and Igor Gitman. Nemotron-Math: Efficient long-context distillation of mathematical reasoning from multi-mode supervision. *arXiv preprint arXiv:2512.15489*, 2025. URL https://arxiv.org/abs/2512.15489.

Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

GLM-5 Team. GLM-5: From vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026. URL https://arxiv.org/abs/2602.15763.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation, 2026. URL https://arxiv.org/abs/2601.20802.

InternLM Team. Intern-S2-Preview. Hugging Face model card, 2026. URL https://huggingface.co/internlm/Intern-S2-Preview.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026. URL https://arxiv.org/abs/2603.07079.

Haodi Lei, Yafu Li, Haoran Zhang, Shunkai Zhang, Qianjia Cheng, et al. Draft-opd: On-policy distillation for speculative draft models, 2026. URL https://arxiv.org/abs/2605.29343.

Yafu Li, Runzhe Zhan, Haoran Zhang, Shunkai Zhang, Yizhuo Li, Zhilin Wang, Jiacheng Chen, Futing Wang, Xuyang Hu, Yuchen Fan, et al. Achieving gold-medal-level olympiad reasoning via simple and unified scaling. *arXiv preprint arXiv:2605.13301*, 2026a. URL https://arxiv.org/abs/2605.13301.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026b. URL https://arxiv.org/abs/2604.13016.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Yun Luo, Futing Wang, Qianjia Cheng, Fangchen Yu, Haodi Lei, Jianhao Yan, Chenxi Li, Jiacheng Chen, Yufeng Zhao, Haiyuan Wan, et al. P1-VL: Bridging visual perception and scientific reasoning in physics olympiads. *arXiv preprint arXiv:2602.09443*, 2026. URL https://arxiv.org/abs/2602.09443.

Thang Luong and Edward Lockhart. Advanced version of Gemini with Deep Think officially achieves gold-medal standard at the international mathematical olympiad. Google DeepMind Blog, jul 2025. URL https://deepmind.google/blog/advanced-version-of-gemini-with-deep-think-officially-achieves-gold-medal-standard-at-the-international-mathematical-olympiad/.

Thang Luong, Dawsen Hwang, Hoang H. Nguyen, Golnaz Ghiasi, Yuri Chervonyi, Insuk Seo, Junsu Kim, Garrett Bingham, Jonathan Lee, Swaroop Mishra, Alex Zhai, Huiyi Hu, Henryk Michalewski, Jimin Kim, Jeonghyun Ahn, Junhwi Bae, Xingyou Song, Trieu Hoang Trinh, Quoc V. Le, and Junehyuk Jung. Towards robust mathematical reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 35418–35442, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.1794. URL https://aclanthology.org/2025.emnlp-main.1794/.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, Jinhao Dong, Zhifang Sui, and Fuli Luo. Mopd: Multi-teacher on-policy distillation for capability integration in llm post-training, 2026. URL https://arxiv.org/abs/2606.30406.

Mathematical Association of America. American invitational mathematics examination (aime) 2025, 2025. URL https://maa.org/math-competitions/american-invitational-mathematics-examination-aime.

OpenAI. OpenAI o1 system card, 2024. URL https://openai.com/index/openai-o1-system-card/.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives, Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily, Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath, Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song, Dane Stuckey, Zhiqing Sun, Philippe Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy Wendling, Kevin Whinnery, Cedric Whitney, Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, Kristen Ying, Wojciech Zaremba, Wenting Zhan, Cyril Zhang, Brian Zhang, Eddie Zhang, and Shengjia Zhao. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025. URL https://arxiv.org/abs/2501.14249.

Qwen Team. Qwen3.5. Model release, 2026. URL https://qwen.ai/blog?id=qwen3.5.

Avyav Kumar Singh, Yen-Chen Wu, Alexandru Cioba, Alberto Bernacchia, and Davide Buffelli. Cross-tokenizer LLM distillation through a byte-level interface. In *Proceedings of the Second Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual*, pp. 84–96, San Diego, California, USA, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.customnlp4u-1.9. URL https://aclanthology.org/2026.customnlp4u-1.9/.

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, 2026. URL https://arxiv.org/abs/2605.07711.

Miles Wang, Robi Lin, Kat Hu, Joy Jiao, Neil Chowdhury, Ethan Chang, and Tejal Patwardhan. FrontierScience: Evaluating AI's ability to perform expert-level scientific tasks. *arXiv preprint arXiv:2601.21165*, 2026. URL https://arxiv.org/abs/2601.21165.

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*, 2025. URL https://arxiv.org/abs/2505.09388.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026. URL https://arxiv.org/abs/2602.12125.

Fangchen Yu, Haiyuan Wan, Qianjia Cheng, Yuchen Zhang, Jiacheng Chen, Fujun Han, Yulun Wu, Junchi Yao, Ruilizhen Hu, Ning Ding, Yu Cheng, Tao Chen, Lei Bai, Dongzhan Zhou, Yun Luo, Ganqu Cui, and Peng Ye. HiPhO: How far are (M)LLMs from humans in the latest high school physics olympiad benchmark? *arXiv preprint arXiv:2509.07894*, 2025. URL https://arxiv.org/abs/2509.07894.

Z.ai. GLM-4.7-Flash. Hugging Face model card, 2026. URL https://huggingface.co/zai-org/GLM-4.7-Flash.

Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen, and Jinan Xu. Dual-space knowledge distillation for large language models. In *Proceedings of the 2024 Conference on Empirical*

Methods in Natural Language Processing, pp. 18164–18181, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1010. URL https://aclanthology.org/2024.emnlp-main.1010/.

Songming Zhang, Xue Zhang, Tong Zhang, Bojie Hu, Yufeng Chen, and Jinan Xu. KDFlow: A user-friendly and efficient knowledge distillation framework for large language models. *arXiv preprint arXiv:2603.01875*, 2026. URL https://arxiv.org/abs/2603.01875. Accepted to the EMNLP 2026 System Demonstrations Track.

Xue Zhang, Songming Zhang, Yunlong Liang, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. A dual-space framework for general knowledge distillation of large language models. *arXiv preprint arXiv:2504.11426*, 2025. URL https://arxiv.org/abs/2504.11426.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: Efficient execution of structured language model programs. In *Advances in Neural Information Processing Systems*, 2024. doi: 10.48550/arXiv.2312.07104. URL https://arxiv.org/abs/2312.07104.

Zilin Zhu, Chengxing Xie, Xin Lv, and slime Contributors. slime: An LLM post-training framework for RL scaling. https://github.com/THUDM/slime, 2025. GitHub repository.

A Additional Experimental Results

A.1 Training Dynamics of Direct OPD

We further visualize the training dynamics of Qwen3.5-35B-A3B under direct OPD from SU-01, i.e., without termination-token masking or the student-reference KL loss. As shown in Figure 6, the task performance deteriorates noticeably as training progresses, and the degradation is accompanied by a persistently high truncation rate and a rapidly increasing response length, consistent with the instability phenomenon discussed in Section 4. This observation motivates the stabilization strategies adopted in the main paper.

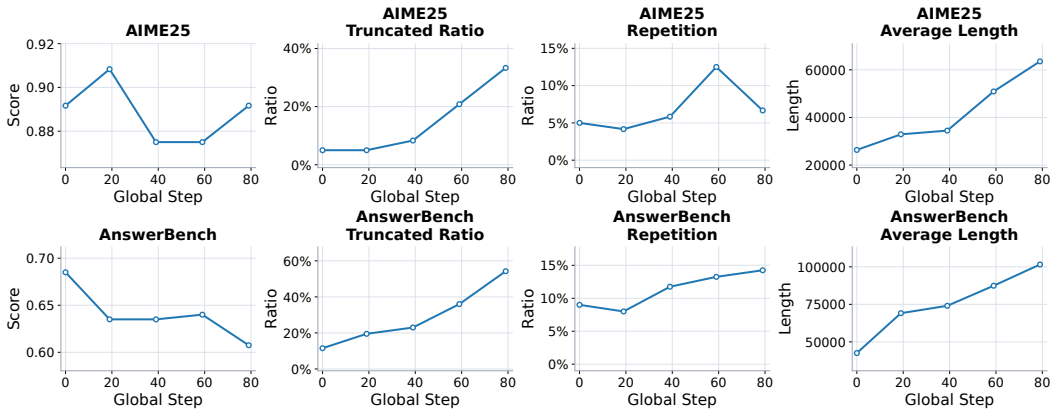

Figure 6: Training dynamics of Qwen3.5-35B-A3B during direct OPD training.

A.2 Termination-Token Masking and Reference-KL Results

Table 5 summarizes the results of distilling Intern-S2-Preview from SU-01 with the two stabilization components applied separately. Masking the advantages of special termination tokens improves ProofBench@4 from 21.70 to 38.10 over the base model, with consistent gains on AnswerBench@8 and AIME25@8. Adding the student-reference KL loss performs slightly better on all three benchmarks, reaching 38.50 on ProofBench@4, 79.10 on AnswerBench@8, and 95.80 on AIME25@8. These results confirm that both components not only stabilize training but also preserve the transfer of proof-reasoning capability from the teacher.

Table 5: OPD results from SU-01 to Intern-S2-Preview with masking termination tokens and adding student reference KL loss.

Model	ProofBench@4	AnswerBench@8	AIME25@8
SU-01	45.00	77.50	94.60
Intern-S2-Preview	21.70	76.03	88.33
OPD + Spec Mask	38.10 (+Δ 16.40)	77.60 (+Δ 1.57)	95.00 (+Δ 6.67)
OPD + Ref KL	38.50 (+Δ 16.80)	79.10 (+Δ 3.07)	95.80 (+Δ 7.47)

A.3 ProofBench Evaluated by Gemini-2.5-Pro

We also evaluate ProofBench using Gemini-2.5-Pro as the judge, following the same evaluation setting as SU-01. The results are shown in Figure 7. The results show that OPD training brings a substantial improvement to Intern-S2-Preview on ProofBench. Specifically, Intern-S2-OPD improves from 34.0 to 55.2, achieving a gain of 21.2 points over the original Intern-S2-Preview model. This demonstrates that OPD can effectively transfer proof-reasoning capabilities from the long-context teacher model to the student model. Notably, Intern-S2-OPD also surpasses several strong frontier models, including Gemini-2.5-Pro and GPT-5. Although it still trails SU-01 and DeepSeek-V3.2-Speciale, the gap is significantly narrowed after OPD training. These results further confirm the effectiveness of OPD for improving natural-language mathematical proof reasoning, especially in cross-model capability transfer settings.

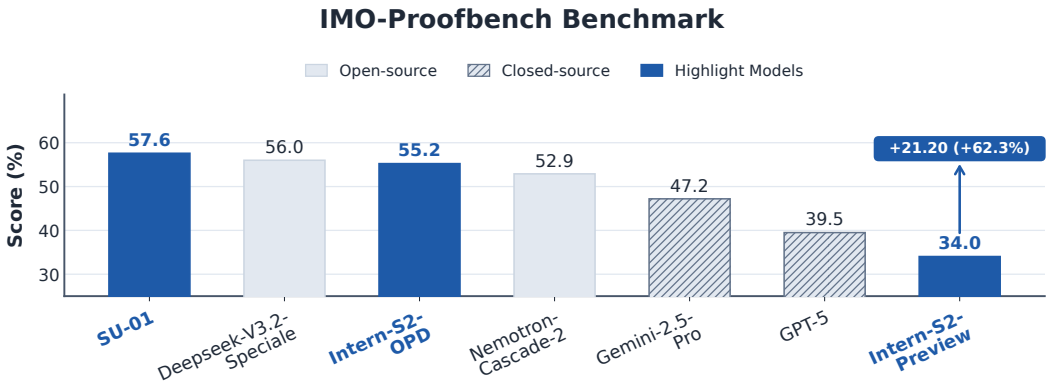

Figure 7: Evaluation performance on ProofBench using Gemini-2.5-Pro as the judge, under the same evaluation setting as SU-01. With SimpleOPD, Intern-S2-OPD achieves performance comparable to DeepSeek-V3.2-Speciale, demonstrating the effectiveness of our OPD strategy.

A.4 Lexical Overlap

The lexical-overlap curves provide an estimate of how much token-level supervision can be retained under cross-tokenizer alignment. As shown in Figure 8, the models already exhibit a high aligned-token ratio at the beginning of training, and the ratio increases further over time. This indicates that, despite tokenizer differences, a large portion of the student-generated text can be matched to teacher tokens through shared surface spans. These results suggest that the partial alignment used in cross-tokenizer OPD preserves a substantial amount of usable training signal in practice, while avoiding the requirement of full tokenizer compatibility.

A.5 Effect of Training Data

We further analyze the effect of data composition in distillation. Specifically, we include the verifiable math data during the OPD process in addition to the proof data.

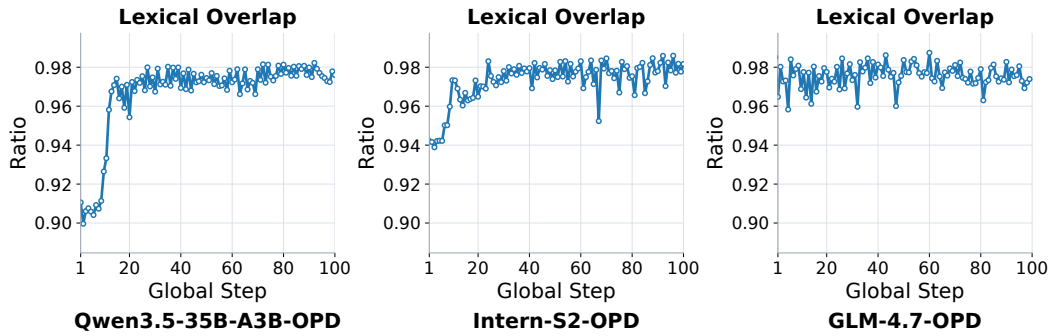

Figure 8: The lexical-overlap curves of Qwen3.5-35B-A3B, Intern-S2-Preview and GLM-4.7-Flash during OPD training from the teacher SU-01.

Table 6: OPD results with proof data and proof+verifiable data.

Model	ProofBench@4	AnswerBench@8	AIME25@8
SU-01	45.00	77.50	94.60
Intern-S2-Preview	21.70	76.03	88.33
OPD proof+verifiable data	38.50 (+Δ 16.80)	**81.10** (+Δ 5.07)	**95.00** (+Δ 6.67)
OPD proof data	**44.50** (+Δ 22.80)	80.10 (+Δ 4.07)	**95.00** (+Δ 6.67)

The results (Table 6) show that both data compositions substantially improve Intern-S2-Preview over the base model, confirming the effectiveness of OPD distillation. Using only proof data achieves the best performance on ProofBench@4, improving Intern-S2 from 21.70 to 44.50 and approaching the teacher model SU-01. This suggests that proof-focused data is more beneficial for transferring natural-language proof-reasoning ability. Adding verifiable math data brings only a marginal improvement on AnswerBench@8, increasing the score from 80.10 to 81.10, while AIME25@8 remains unchanged at 95.00. However, this mixed-data setting significantly underperforms proof-only data on ProofBench@4, dropping from 44.50 to 38.50. These results indicate that adding verifiable data provides limited additional benefit for general mathematical reasoning, while potentially weakening the transfer of proof-reasoning ability.

A.6 Effect of Student-Reference KL Loss

We further investigate how the student-reference KL coefficient affects GLM distillation. This coefficient controls the extent to which the student is regularized toward its reference policy and therefore influences the balance between preserving the base model's behavior and learning from the teacher. We evaluate coefficients of 0.5, 1.0, and 1.2 to study this trade-off.

Table 7: Ablation study on the student-reference KL coefficient in the setting from SU-01 to GLM-4.7-Flash. Green values indicate improvements over the GLM-4.7-Flash base model.

Model	Student KL Coefficient	ProofBench@4	AnswerBench@8	AIME25@8
SU-01	–	45.00	77.50	94.60
GLM-4.7-Flash	–	30.75	69.59	92.08
GLM-4.7-Flash-OPD	0.5	37.15 (+Δ 6.40)	71.28 (+Δ 1.69)	**94.17** (+Δ 2.09)
	1.0	**39.71** (+Δ 8.96)	**71.97** (+Δ 2.38)	92.91 (+Δ 0.83)
	1.2	37.96 (+Δ 7.21)	71.19 (+Δ 1.60)	**94.17** (+Δ 2.09)

As shown in Table 7, GLM-4.7-Flash-OPD consistently outperforms the base model under all three settings. Among them, a coefficient of 1.0 achieves the best overall performance, improving ProofBench@4 from 30.75 to 39.71 and AnswerBench@8 from 69.59 to 71.97, with gains of 8.96 and 2.38 points, respectively. A smaller coefficient of 0.5 provides insufficient regularization and yields

lower ProofBench and AnswerBench scores. Conversely, increasing the coefficient to 1.2 imposes a stronger constraint on the student policy, which may limit its ability to absorb the teacher's reasoning capabilities. Although coefficients of 0.5 and 1.2 achieve the highest AIME25@8 score of 94.17, their overall performance is less balanced than that obtained with a coefficient of 1.0. These results suggest that the student-reference KL coefficient should be neither too small nor too large, and that a moderate value provides the best balance between retaining the student's original capabilities and acquiring knowledge from the teacher.

A.7 Comparison with OPD Baselines on Qwen3.5-35B-A3B

Figure 9 extends the comparison with EOPD (Jin et al., 2026) and G-OPD (Yang et al., 2026) in Figure 5 from Intern-S2-Preview to Qwen3.5-35B-A3B. With Qwen3.5-35B-A3B as the student base model, SimpleOPD achieves 42.39 on ProofBench@4, 80.16 on AnswerBench@8, 96.67 on AIME25@8, and 61.25 on AMOBench@8, obtaining the best result on all four benchmarks. Compared with the stronger of the two OPD baselines on each benchmark, SimpleOPD improves the four scores by 3.67, 3.41, 2.17, and 7.25 points, respectively. Together with the Intern-S2-Preview results in Figure 5, this comparison shows that the overall advantage of SimpleOPD extends across different student base models.

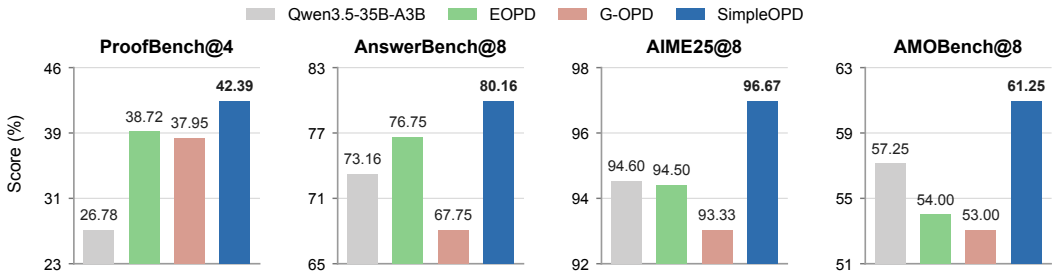

Figure 9: Comparison of EOPD, G-OPD, and SimpleOPD using Qwen3.5-35B-A3B as the student base model across four reasoning benchmarks. All scores are reported as percentages.

B Cross-Tokenizer Scoring and Alignment

The complete procedure of cross-tokenizer scoring and alignment is shown in Algorithm 1.

As shown in Figure 10, which plots an example of cross-tokenizer scoring in the distillation from SU-01 to Intern-S2-Preview, the visualization confirms that our alignment process allows most semantically informative tokens to receive supervision from the teacher.

C Model Solutions for ProofBench

This section presents two ProofBench examples to provide a qualitative view of the mathematical proof-generation behavior learned through SimpleOPD. Both ProofBench-Basic-001 and ProofBench-Advanced-028 are outputs generated by Intern-S2-OPD, the Intern-S2-Preview student distilled from SU-01; each case includes the original problem statement, a concise solution overview, and the complete model-generated proof.

C.1 ProofBench-Basic-001

Question

Determine all functions $f\colon \mathbb{Z} \to \mathbb{Z}$ such that, for all $x, y \in \mathbb{Z}$,

$$f(2x) + 2f(y) = f(f(x+y)).$$

Algorithm 1 Cross-tokenizer teacher scoring and greedy token-piece alignment

Inputs : message sequence $\mathbf{x}$; student response tokens $y_{1:n}$; student log-probabilities $\ell^{\theta}_{1:n}$.
Outputs: aligned teacher log-probabilities $\widetilde{\ell}^{\phi}_{1:n}$; matched set $\mathcal{M}$.

▷ **Token-out, text-in teacher scoring**

$s \leftarrow \mathcal{D}_\theta(y_{1:n}), \quad c_\phi \leftarrow \mathcal{C}_\phi(\mathbf{x}), \quad u_\phi \leftarrow c_\phi \oplus s$

$z = (z_1, \ldots, z_m) \leftarrow \mathcal{E}_\phi(s \mid c_\phi)$

$\ell^{\phi}_i \leftarrow \log \pi_\phi(z_i \mid c_\phi, z_{<i}), \quad \forall i \in \{1, \ldots, m\}$

▷ **Initialize all positions with student log-probabilities**

$\widetilde{\ell}^{\phi}_{1:n} \leftarrow \ell^{\theta}_{1:n}, \quad \mathcal{M} \leftarrow \emptyset$

$i \leftarrow 1, \quad j \leftarrow 1, \quad H_\phi \leftarrow \varepsilon, \quad H_\theta \leftarrow \varepsilon$

▷ τ_ϕ **and** τ_θ **return tokenizer-native token pieces**

while $i \le m \wedge j \le n$ **do**
  $p_\phi \leftarrow \tau_\phi(z_i), \quad p_\theta \leftarrow \tau_\theta(y_j)$
  if $H_\phi = H_\theta \wedge p_\phi = p_\theta$ **then**
    $\widetilde{\ell}^{\phi}_j \leftarrow \ell^{\phi}_i, \quad \mathcal{M} \leftarrow \mathcal{M} \cup \{(i, j)\}$
    $H_\phi \leftarrow H_\phi \oplus p_\phi, \quad H_\theta \leftarrow H_\theta \oplus p_\theta, \quad i \leftarrow i + 1, \quad j \leftarrow j + 1$
  else if $|H_\phi| > |H_\theta|$ **then**
    $H_\theta \leftarrow H_\theta \oplus p_\theta, \quad j \leftarrow j + 1$
  else if $|H_\phi| < |H_\theta|$ **then**
    $H_\phi \leftarrow H_\phi \oplus p_\phi, \quad i \leftarrow i + 1$
  else
    $H_\phi \leftarrow H_\phi \oplus p_\phi, \quad H_\theta \leftarrow H_\theta \oplus p_\theta, \quad i \leftarrow i + 1, \quad j \leftarrow j + 1$
  end if
end while

return $\widetilde{\ell}^{\phi}_{1:n}, \mathcal{M}$

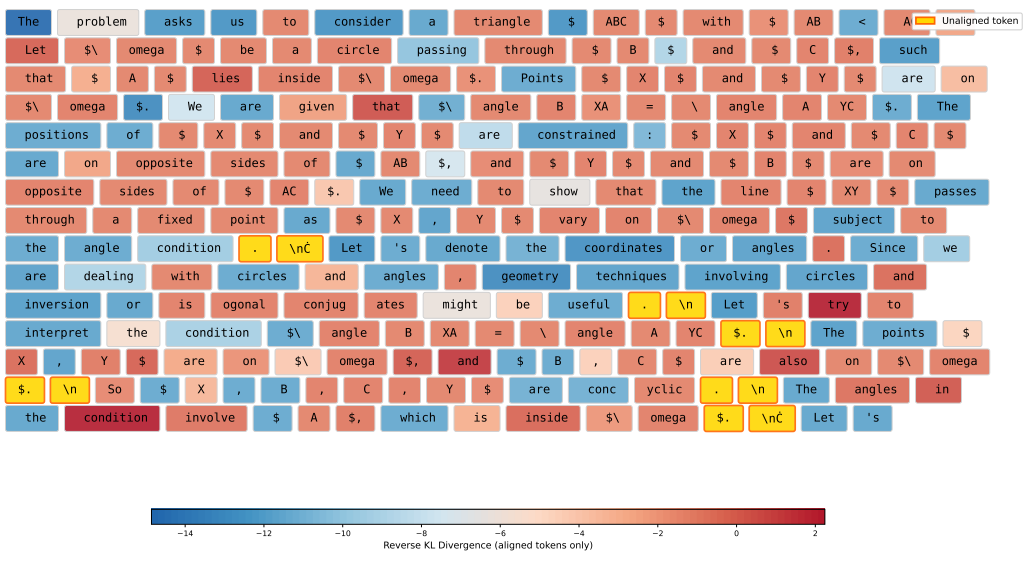

Figure 10: Per-token reverse KL divergence in cross-vocabulary distillation. Colored boxes show aligned tokens with KL divergence values, while gold boxes indicate tokens that could not be aligned between teacher and student models.

Solution Overview

The solution first evaluates the equation at carefully chosen arguments to derive a doubling identity and a translated Cauchy relation. Subtracting the constant $f(0)$ then reduces the problem to an additive function on $\mathbb{Z}$, which must be linear. Comparing coefficients and directly verifying the resulting candidates completes the classification.

Intern-S2-OPD Solution

Solution.

For all $x, y \in \mathbb{Z}$, let $P(x, y)$ denote the assertion

$$f(2x) + 2f(y) = f(f(x + y)) \tag{C.1}$$

Step 1. Two useful identities.

For every $t \in \mathbb{Z}$, $P(0, t)$ and $P(t, 0)$ give, respectively,

$$f(f(t)) = f(0) + 2f(t), \tag{C.2}$$
$$f(f(t)) = f(2t) + 2f(0). \tag{C.3}$$

Equating these expressions gives, for every $t \in \mathbb{Z}$,

$$f(2t) = 2f(t) - f(0). \tag{C.4}$$

Step 2. Deriving a Cauchy-type relation.

Substituting (C.4) into (C.1) yields

$$f(f(x + y)) = 2f(x) + 2f(y) - f(0). \tag{C.5}$$

On the other hand, applying (C.2) with t replaced by $x + y$ gives

$$f(f(x + y)) = f(0) + 2f(x + y). \tag{C.6}$$

Equating (C.5) and (C.6), we obtain, for all $x, y \in \mathbb{Z}$,

$$f(0) + 2f(x + y) = 2f(x) + 2f(y) - f(0),$$

and hence

$$f(x + y) = f(x) + f(y) - f(0) \tag{C.7}$$

Step 3. Additivity after subtracting the constant.

Define $g(x) = f(x) - f(0)$. Then $g(0) = 0$, and, for all $x, y \in \mathbb{Z}$, (C.7) becomes

$$g(x + y) = g(x) + g(y).$$

Thus g is additive on $\mathbb{Z}$. Since $\mathbb{Z}$ is cyclic, any additive map $\mathbb{Z} \to \mathbb{Z}$ is determined by its value at 1. In particular, for every $n \in \mathbb{Z}$,

$$g(n) = ng(1).$$

Let $a = g(1) \in \mathbb{Z}$ and $b = f(0)$. Consequently, for every $x \in \mathbb{Z}$,

$$f(x) = ax + b.$$

Step 4. Determining the parameters a and b.

Substituting $f(x) = ax + b$ into (C.1), the left-hand side is

$$\begin{aligned} f(2x) + 2f(y) &= a(2x) + b + 2(ay + b) \\ &= 2a(x + y) + 3b, \end{aligned}$$

while the right-hand side is

$$\begin{aligned} f(f(x + y)) &= f(a(x + y) + b) \\ &= a^2(x + y) + ab + b. \end{aligned}$$

Since $t = x + y$ ranges over $\mathbb{Z}$, for every $t \in \mathbb{Z}$ we have

$$2at + 3b = a^2t + ab + b.$$

Intern-S2-OPD Solution (continued)

Therefore,

$$\begin{aligned} 2a - a^2 = 0, \quad 2b - ab = 0, \\ a(2-a) = 0, \quad b(2-a) = 0. \end{aligned} \tag{C.8}$$

It follows that $a = 0$ or $a = 2$. If $a = 0$, then $2b = 0$, so $b = 0$ and $f \equiv 0$. If $a = 2$, the second condition is automatically satisfied for every $b \in \mathbb{Z}$, giving $f(x) = 2x + b$.

Step 5. Verification.

For $f \equiv 0$, both sides of (C.1) are zero. For $f(x) = 2x + b$,

$$\begin{aligned} f(2x) + 2f(y) &= (4x + b) + 2(2y + b) \\ &= 4x + 4y + 3b, \end{aligned}$$

and

$$\begin{aligned} f(f(x + y)) &= f(2(x + y) + b) \\ &= 4x + 4y + 3b. \end{aligned}$$

Hence both families satisfy the equation, and the complete set of solutions is

$$\boxed{\begin{aligned} &f \equiv 0, \quad \text{or} \\ &f(x) = 2x + c, \qquad c \in \mathbb{Z}. \end{aligned}}$$

C.2 PROOFBENCH-ADVANCED-028

Question

Let H be the orthocenter of acute triangle ABC, let F be the foot of the altitude from C to AB, and let P be the reflection of H across BC. Suppose that the circumcircle of triangle AFP intersects line BC at two distinct points X and Y. Prove that C is the midpoint of XY.

Solution Overview

The solution places BC on the horizontal axis and computes the coordinates of F, H, and P. It then shows that the circumcenter of $\triangle AFP$ lies on the vertical line through C. Since this line is perpendicular to the chord $XY \subset BC$, the standard chord-bisection property of a circle implies that C is the midpoint of XY.

Intern-S2-OPD Solution

Solution.

Step 1. Coordinate setup.

We place the triangle in a convenient coordinate system. Since the statement involves only midpoints, collinearity, and perpendicularity, we may apply a similarity transformation without loss of generality. Thus, let

$$B = (0, 0), \qquad C = (1, 0), \qquad A = (p, q), \tag{C.2.1}$$

where $q > 0$. Since the triangle is acute, $0 < p < 1$, although these inequalities are not essential for the algebra below. Define

$$D = p^2 + q^2.$$

Step 2. Coordinates of the auxiliary points.

The line AB passes through the origin and has direction (p, q). Therefore, the projection of $C = (1, 0)$ onto AB is

$$F = \frac{C \cdot A}{A \cdot A} A = \frac{p}{p^2 + q^2}(p, q) = \left(\frac{p^2}{D}, \frac{pq}{D}\right). \tag{C.2.2}$$

Intern-S2-OPD Solution (continued)

Because BC is horizontal, the altitude from A is the vertical line $x = p$. The altitude from B is perpendicular to AC. Since $AC = (1-p, -q)$, a perpendicular direction is $(q, 1-p)$, and hence this altitude has equation

$$y = \frac{1-p}{q}x.$$

Intersecting it with $x = p$ gives

$$H = \left(p, \frac{p(1-p)}{q}\right). \tag{C.2.3}$$

Reflecting H across BC, which is the x-axis, yields

$$P = \left(p, -\frac{p(1-p)}{q}\right). \tag{C.2.4}$$

Step 3. A candidate for the circumcenter of $\triangle AFP$.

The points A and P have the same x-coordinate, so AP is vertical. Its perpendicular bisector is the horizontal line through the midpoint

$$\begin{aligned} M_{AP} &= \left(p, \frac{q - \frac{p(1-p)}{q}}{2}\right) \\ &= \left(p, \frac{q^2 - p(1-p)}{2q}\right). \end{aligned}$$

Since

$$q^2 - p(1-p) = p^2 + q^2 - p = D - p,$$

define

$$y_0 = \frac{D-p}{2q}. \tag{C.2.5}$$

The perpendicular bisector of AP is therefore $y = y_0$. We claim that

$$O = (1, y_0) \tag{C.2.6}$$

is the circumcenter of $\triangle AFP$. By construction, O lies on the perpendicular bisector of AP. It remains to show that O is equidistant from A and F.

Step 4. Verifying that $OA = OF$.

From equations (C.2.1), (C.2.2), and (C.2.6),

$$\begin{aligned} OA^2 &= (1-p)^2 + (y_0 - q)^2, \\ OF^2 &= \left(1 - \frac{p^2}{D}\right)^2 + \left(y_0 - \frac{pq}{D}\right)^2. \end{aligned} \tag{C.2.7}$$

First observe that

$$1 - \frac{p^2}{D} = \frac{D - p^2}{D} = \frac{q^2}{D}. \tag{C.2.8}$$

Consider the difference

$$\begin{aligned} \Delta &= OA^2 - OF^2 \\ &= \left[(1-p)^2 - \left(1 - \frac{p^2}{D}\right)^2\right] + \left[(y_0 - q)^2 - \left(y_0 - \frac{pq}{D}\right)^2\right]. \end{aligned} \tag{C.2.9}$$

Factoring the first bracket as a difference of squares gives

$$\begin{aligned} &(1-p)^2 - \left(1 - \frac{p^2}{D}\right)^2 \\ &= \left(\frac{p^2}{D} - p\right)\left(2 - p - \frac{p^2}{D}\right) \\ &= -\frac{p(D-p)}{D}\left(2 - p - \frac{p^2}{D}\right). \end{aligned} \tag{C.2.10}$$

Intern-S2-OPD Solution (continued)

Similarly,

$$\begin{aligned}&(y_0-q)^2-\left(y_0-\frac{pq}{D}\right)^2\\&=\left(\frac{pq}{D}-q\right)\left(2y_0-q-\frac{pq}{D}\right)\\&=-\frac{q(D-p)}{D}\left(2y_0-q-\frac{pq}{D}\right).\end{aligned}\tag{C.2.11}$$

Substituting equations (C.2.10) and (C.2.11) into equation (C.2.9), we obtain

$$\begin{aligned}\Delta=-\frac{D-p}{D}\Bigg[&p\left(2-p-\frac{p^2}{D}\right)\\&+q\left(2y_0-q-\frac{pq}{D}\right)\Bigg].\end{aligned}\tag{C.2.12}$$

Using equation (C.2.5),

$$\begin{aligned}q\left(2y_0-q-\frac{pq}{D}\right)&=q\left(\frac{D-p}{q}-q-\frac{pq}{D}\right)\\&=(D-p)-q^2-\frac{pq^2}{D}.\end{aligned}\tag{C.2.13}$$

Consequently, the bracket in equation (C.2.12) equals

$$\begin{aligned}&p\left(2-p-\frac{p^2}{D}\right)+(D-p)-q^2-\frac{pq^2}{D}\\&=2p-p^2-\frac{p^3}{D}+D-p-q^2-\frac{pq^2}{D}\\&=p+\left(D-p^2-q^2\right)-\frac{p(p^2+q^2)}{D}\\&=p+0-p=0.\end{aligned}$$

It follows that $\Delta=0$, and hence $OA^2=OF^2$. Therefore, O lies on the perpendicular bisector of AF as well as that of AP, so O is the circumcenter of $\triangle AFP$.

Step 5. Bisecting the chord XY.

The circumcenter $O=(1,y_0)$ and the point $C=(1,0)$ have the same x-coordinate. Since BC is the x-axis, the line OC is perpendicular to BC. The chord XY lies on BC, and the perpendicular from the center of a circle to a chord bisects that chord. Thus, C is the midpoint of XY.

$$\boxed{C \text{ is the midpoint of } XY.}$$

D Case Study

This appendix presents two representative instances of direct OPD when the teacher model SU-01 is distilled to Qwen3.5-35B-A3B. In both cases, the student degenerates into an endless repetition—either a fixed multi-sentence self-check block or a single-token loop. The rollouts terminate only when hitting the 160k-token generation limit.

D.1 Case A: correct answer, then 972 exact self-check repeats (AIME25)

The model solves an AIME 2025 divisibility problem, reaches the correct numerical answer $\boxed{279}$, but then appends the same ten-sentence "verification" unit verbatim 972 times. This loop occupies nearly the entire remaining generation budget and is truncated mid-unit.

Question

Determine the difference between N and 2025, where N is the number of eight-digit positive integers using each of the digits $1, 2, \ldots, 8$ exactly once that are divisible by 22.

Model Output (excerpt)

Beginning of output.

We need to solve: there are $8! = 40320$ eight-digit positive integers that use each of the digits $1, 2, 3, 4, 5, 6, 7, 8$ exactly once. Let N be the number of these integers that are divisible by 22. Find the difference between N and 2025. So we need to compute N, then compute $N - 2025$. Since $22 = 2 \cdot 11$, we need divisibility by 2 and by 11.

At the failure onset (response chars 75,000–77,150).

Consequently, $N = 144 \times 16 = 2304$. Finally, $N - 2025 = 2304 - 2025 = 279$. Thus the answer is $\boxed{279}$. I'll also ensure that the solution is not missing any step.

Repeating pattern — exact 10-sentence self-check unit (542 characters), repeated 972 times.

> "I'll also ensure that the solution is not missing any step."
> "I'll also ensure that the solution is clear."
> "I'll also ensure that the solution is correct."
> "I'll also ensure that the solution is comprehensive."
> "I'll also ensure that the solution is easy to follow."
> "I'll also ensure that the solution is in the required format."
> "I'll also ensure that the solution is not too long."
> "I'll also ensure that the solution is self-contained."
> "I'll also ensure that the solution is not missing any justification."
> "I'll also ensure that the solution is correct."

The model reaches the correct answer 279, then repeats the same ten-sentence self-check unit exactly 972 times. The rollout hits the 160k-token generation limit inside this unit and is recorded as truncated; the response ends mid-unit with "… not missing any justification. I'll also ensure that the".

D.2 Case B: reasoning collapses into a one-token loop (AnswerBench)

In this case, the model engages with a complex number-theoretic definition ("exotic integers"). After some initial reasoning, it gets stuck on a sub-problem involving multiplicative orders and begins emitting the token sequence "$2^?$" indefinitely. The repetition is atomic (one token) and repeats over 34,000 times with no progress or termination.

Question

For a positive integer n, call $g\colon \mathbb{Z} \to \mathbb{Z}$ an n-good function if $g(1) = 1$ and, for distinct integers a and b, $g(a) - g(b)$ divides $a^n - b^n$. Call n exotic if the number of n-good functions is twice an odd integer. Find the 132nd exotic integer.

Model Output (excerpt)

Beginning of output.

We need to solve: for positive integer n, define $g\colon \mathbb{Z} \to \mathbb{Z}$ such that $g(1) = 1$ and for any two distinct integers a and b, $g(a) - g(b)$ divides $a^n - b^n$. We call n an exotic integer if the number of n-good functions is twice an odd integer. Find the 132nd exotic integer. We need to understand the condition and count the number of such functions for a given n.

At the failure onset (response tokens 840–919).

For $m = 8$: $2^8 - 1 = 255$, order of 2 mod 255? $2^8 = 256 \equiv 1 \pmod{255}$, but $2^4 = 16 \pmod{255}$, not 1; … but could be smaller? $2^?$ Actually … $2^? \; 2^? \; 2^?$

Repeating pattern — exact one-token unit, repeated many times.

$$2^{?}\ \ 2^{?}\ \ 2^{?}\ \ 2^{?}\ \ 2^{?}\ \ 2^{?}\ \ 2^{?}\ \ 2^{?}\ \ 2^{?}\ \ 2^{?}\quad \cdots$$

The identical token occupies one uninterrupted run; no new claim, calculation, or answer is produced after the loop begins. The rollout reaches the 160k-token generation limit inside this run and is recorded as truncated; the response ends with "$\ldots\ 2^{?}\ 2^{?}\ 2^{?}\ 2^{?}\ 2^{?}$".