arXiv:2608.19181v1 [cs.LG] 19 Aug 2026

Beyond Teacher Likelihood: Group-Calibrated On-Policy Distillation for Long-Context Reasoning

Zhu Zhang[1,3]* **Jixun Wang**[2,3]* **Xiaoang Xu**[2,3] **Xiaorong Wang**[3]
Zihan Zhou[3] **Zhiyuan Wang**[1] **Shuo Wang**[1] **Chaojun Xiao**[1]† **Yuezhi Zhou**[1]†

[1]Tsinghua University [2]Beijing University of Posts and Telecommunications [3]OpenBMB

Abstract

On-policy distillation (OPD) trains a student on its own responses using dense token-level guidance from a stronger teacher. In long-context tasks, however, token-level teacher support can favor locally plausible responses that omit evidence distributed across the input or violate global task constraints. Task-specific verifiers, in contrast, evaluate task completion at the response level and may return graded rewards that reflect partial success. We diagnose this mismatch on fixed responses from two representative long-context evidence-aggregation tasks. Across longer input ranges, trajectory-level OPD scores become progressively less aligned with verifier rewards, indicating teacher–verifier disagreement. Motivated by this observation, we introduce Group-Calibrated On-Policy Distillation (GC-OPD). GC-OPD separately normalizes verifier rewards and trajectory-level OPD scores within each rollout group and uses their difference as a signed teacher–verifier disagreement residual. Relative-advantage-based credit assignment (RACA) distributes this trajectory-level residual across tokens according to their relative OPD advantages while preserving the original OPD signal. Across five long-context benchmarks, post-training with GC-OPD raises the five-benchmark averages of the official Qwen3-4B and Qwen3-8B checkpoints from 29.08 to 40.47 and from 35.12 to 44.65, respectively. Vanilla OPD reaches 39.31 and 43.56 under the same setup. Controlled ablations show that the signed residual is more effective than either an additional OPD-derived term or direct group-normalized verifier reward addition, while RACA further improves over uniform token allocation. Together, these results demonstrate that group-relative residual calibration can incorporate verifier outcomes without discarding dense token-level guidance. Code is available at `https://github.com/SolereZhang/GC-OPD`.

1 Introduction

Knowledge distillation transfers knowledge from a strong teacher to a weaker student and is particularly valuable for compact language models [8, 7, 13]. On-policy distillation (OPD) applies this paradigm to student-generated responses, providing dense teacher guidance on the student's own generation distribution [1]. For each sampled token, OPD derives an advantage from the difference between its teacher and student log-probabilities. This dense signal underlies strong recent OPD methods [31, 16, 34], but measures teacher preference rather than verified task success.

This distinction is particularly important for long-context tasks, where correct responses must integrate evidence scattered across distant positions and satisfy global task constraints [3, 10, 6, 4]. With longer inputs, student models are more likely to produce locally plausible but globally incomplete responses, as they can struggle to locate and use relevant information when its position or lexical

*Equal contribution. Emails: `zhuzhang24@mails.tsinghua.edu.cn`, `wangjixun@bupt.edu.cn`.
†Corresponding authors. Emails: `xcj@tsinghua.edu.cn`, `zhouyz@mail.tsinghua.edu.cn`.

Preprint. Work in progress.

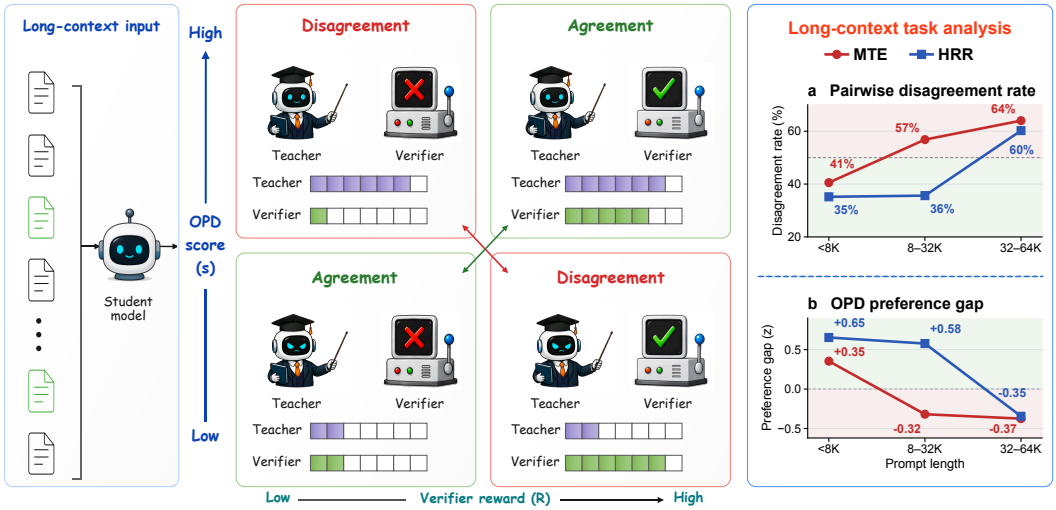

Figure 1: Teacher–verifier disagreement in two evidence-aggregation tasks: Multi-Table Extraction (MTE) and High-Recall Retrieval (HRR). Pairwise disagreement rate captures ranking conflicts, while OPD preference gap captures the direction and magnitude of OPD preference along the verifier ordering. Both indicate weaker alignment over longer input ranges. See Section 3.2 for details.

form changes [18, 21]. A response may therefore receive strong teacher support while omitting required evidence. Task-specific verifiers, in contrast, evaluate task completion at the response level and may return graded rewards that reflect partial success [23]. We call discrepancies between teacher preference and verified task outcomes teacher–verifier disagreement. Figure 1 presents diagnostic evidence from a fixed set of generated responses in two representative tasks that share this evidence-aggregation requirement. In both, trajectory-level OPD scores become progressively less aligned with verifier rewards across the analyzed input-length ranges. This trend motivates an interface that preserves dense teacher guidance while correcting response-level preferences inconsistent with verified outcomes.

Existing verifier-aware OPD methods use outcome feedback through objective routing, teacher-context reconstruction, outcome-class calibration, or token-level gating [36, 32, 9, 29]. These methods establish that verifier feedback complements dense teacher guidance. However, they do not simultaneously preserve graded within-group outcome differences and translate the resulting response-level discrepancy into token-dependent calibration while retaining the original OPD advantage.

To address these limitations, we propose Group-Calibrated On-Policy Distillation (GC-OPD). Following the group-relative comparison in Group Relative Policy Optimization [23], GC-OPD separately normalizes verifier rewards and trajectory-level OPD scores within each rollout group. As Figure 2 shows, their difference forms a signed residual whose direction and magnitude quantify teacher–verifier disagreement. Subtracting the group-normalized OPD assessment focuses calibration on their discrepancy rather than directly adding verifier feedback. Relative-advantage-based credit assignment (RACA) distributes this residual across tokens according to their relative OPD advantages while retaining the original dense OPD signal.

Our contributions are as follows:

- We identify a shared pattern across two distributed-evidence long-context tasks: trajectory-level OPD scores become progressively less aligned with verifier rewards over longer inputs, as measured by pairwise disagreement rate and OPD preference gap.
- We propose GC-OPD, which retains the original OPD advantage while forming a signed residual between group-normalized verifier rewards and trajectory-level OPD scores. RACA distributes this residual across tokens using relative OPD advantages.
- Across two model scales and five long-context benchmarks, GC-OPD achieves the highest average score under the shared setup. Ablations show that the signed residual is more effective than either an additional OPD-derived term or direct group-normalized verifier reward addition, while RACA improves over uniform token allocation.

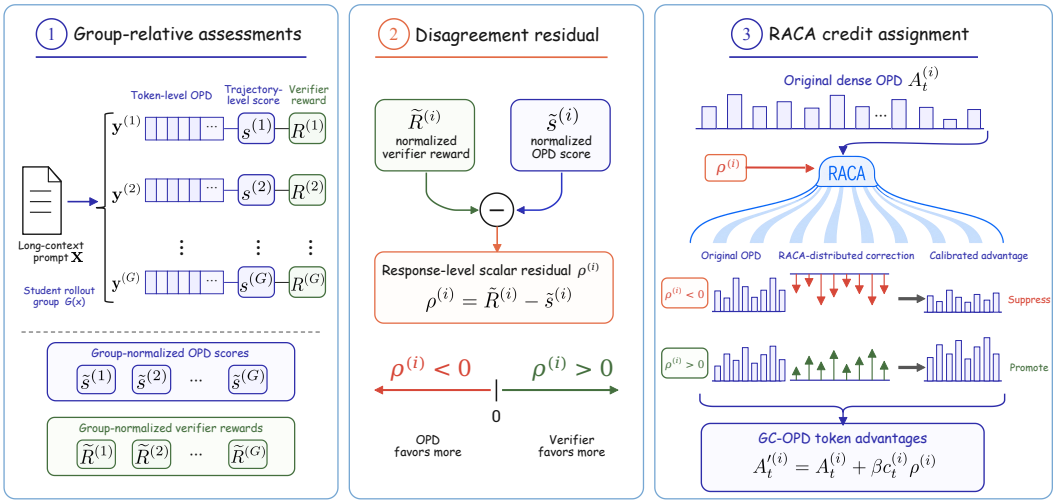

Figure 2: Overview of GC-OPD. GC-OPD computes group-normalized verifier rewards and trajectory-level OPD scores, forms their signed residual, and uses RACA to distribute the residual across response tokens while retaining the original OPD advantage.

2 Related Work

Long-context reasoning and verifiable post-training. Long-context benchmarks show that nominal context capacity does not guarantee reliable retrieval, integration, or reasoning over evidence distributed across long inputs [3, 10, 18, 6, 21]. Corresponding post-training work improves these capabilities through long-instruction data, length-aware optimization, reinforcement-learning curricula, and task-specific feedback [2, 25, 27, 5, 33]. These studies improve the construction of long-context data, curricula, and rewards. Our setting instead begins with an existing task verifier and asks how its response-level assessment should interact with dense token-level OPD guidance.

Teacher-centric on-policy distillation. MiniLLM and GKD distill on student-generated sequences, exposing the teacher to the student's own generation distribution [7, 1]. ExOPD extrapolates along a teacher–reference direction [31], FiRe-OPD filters trajectories and reweights informative tokens [16], and PowerOPD bounds sampled-token rewards for stability [34]. Complementary analyses examine teacher–student compatibility and whether large token-level disagreement is learnable, while token-selective distillation concentrates supervision on selected positions [15, 28, 12]. This line improves how teacher-derived supervision is formed, selected, or optimized. It does not, however, explicitly compare the resulting trajectory-level OPD assessment with a task-specific verifier reward.

Verifier-aware on-policy distillation. Recent methods connect verifier outcomes to teacher supervision through several interfaces. SCOPE routes correct responses to weighted maximum likelihood and incorrect responses to teacher-perplexity-weighted OPD [36]. MOPD conditions teacher supervision on successful and failed peer responses [32]. Uni-OPD calibrates trajectory returns using an outcome-class margin [9]. SG-OPD gates token updates by sign consistency and supplements early training with verifier-endorsed teacher rollouts [29]. Reward-Weighted OPD selects verifier-passable rollouts and weights their dense teacher supervision [37]. These methods introduce verifier feedback through objective routing, teacher conditioning, return calibration, token gating, or trajectory weighting. GC-OPD instead leaves the teacher pass unchanged and forms a response-specific residual from group-relative verifier and OPD assessments. It retains graded within-group differences and the original token-level OPD advantage.

Trajectory-level feedback and token-level credit. A terminal verifier supplies one response-level reward without specifying its tokenwise influence. Process-supervision methods use human or automatically constructed step labels [17, 26], while VinePPO estimates action advantages from auxiliary continuations [11]. These approaches obtain finer-grained evidence through additional labels or sampling. GC-OPD requires neither step labels nor auxiliary continuations. RACA instead allocates the trajectory-level residual with bounded weights derived from relative OPD advantages without inferring token correctness.

3 Background and Motivation

3.1 On-Policy Distillation

Let $\mathbf{x}$ denote a long-context input and π_θ the trainable student policy. At the beginning of each update, $\pi_{\theta_{\text{old}}}$ denotes a frozen copy of π_θ used to generate the current rollouts. For each input, we sample a rollout group of G responses, $\mathcal{G}(\mathbf{x}) = \{\mathbf{y}^{(i)}\}_{i=1}^{G}$, from $\pi_{\theta_{\text{old}}}$, where $\mathbf{y}^{(i)} = (y_1^{(i)}, \ldots, y_{T^{(i)}}^{(i)})$ has length $T^{(i)}$.

Let π_T denote the teacher policy. For each response sampled from $\pi_{\theta_{\text{old}}}$, vanilla OPD defines the token-level advantage of $y_t^{(i)}$ as

$$\begin{aligned} A_t^{(i)} = \; & \log \pi_T(y_t^{(i)} \mid \mathbf{x}, \mathbf{y}_{<t}^{(i)}) \\ & - \log \pi_{\theta_{\text{old}}}(y_t^{(i)} \mid \mathbf{x}, \mathbf{y}_{<t}^{(i)}). \end{aligned} \tag{1}$$

Let $h_t^{(i)} = (\mathbf{x}, \mathbf{y}_{<t}^{(i)})$ denote the input and response prefix at position t. Taking the expectation over a token sampled from the frozen rollout policy gives

$$\mathbb{E}_{y_t^{(i)} \sim \pi_{\theta_{\text{old}}}(\cdot \mid h_t^{(i)})}\left[A_t^{(i)}\right] = -\mathrm{KL}\Big(\pi_{\theta_{\text{old}}}(\cdot \mid h_t^{(i)}) \| \pi_T(\cdot \mid h_t^{(i)})\Big). \tag{2}$$

Thus, $A_t^{(i)}$ is a sampled-token contribution whose expectation equals the negative reverse KL [7, 34]. A positive value indicates greater teacher than student support for the sampled token, whereas a negative value indicates the converse. The signal is dense but reflects teacher preference rather than verified task outcome.

During actor optimization, $A_t^{(i)}$ is treated as fixed and weights the clipped token-level objective for π_θ, whose importance ratio is computed relative to $\pi_{\theta_{\text{old}}}$. Positive advantages increase the probability of sampled tokens, whereas negative advantages decrease it.

3.2 Teacher–Verifier Disagreement in Long-Context OPD

A task-specific verifier assigns each response a reward $R^{(i)}$, which may encode binary correctness or graded partial success. To compare this response-level outcome feedback with token-level OPD guidance, we aggregate the token-level advantages into a trajectory-level OPD score:

$$s^{(i)} = \frac{1}{T^{(i)}} \sum_{t=1}^{T^{(i)}} A_t^{(i)}. \tag{3}$$

Averaging avoids direct scaling with response length and summarizes the teacher's mean support for the response relative to the frozen rollout policy.

The trajectory-level OPD score and verifier reward assess different properties of the same response. The former summarizes teacher support relative to the frozen rollout policy, whereas the latter measures verified task outcome. Their orderings within a rollout group can therefore disagree: a response with a higher OPD score may receive a lower verifier reward than another response. We refer to such discrepancies as *teacher–verifier disagreement*. This disagreement is particularly consequential in long-context OPD, where task success can depend on aggregating evidence distributed across distant positions. In this setting, token-level teacher support may favor a locally plausible response even when it omits evidence required by the verifier. Because the two signals also differ in numerical scale, we characterize their agreement through relative comparisons within each rollout group rather than their raw values.

We examine how teacher–verifier disagreement varies with prompt length using a fixed set of generated responses from Multi-Table Extraction and High-Recall Retrieval in GoLongRL [20]. Both tasks require aggregating evidence distributed across the input and use graded verifier rewards. The analysis covers 751 Multi-Table prompts and 2,908 High-Recall prompts across three prompt-length ranges below 64K tokens. We use Qwen3-8B as the student and Qwen3-30B-A3B-Thinking-2507 as the teacher. We extend the student's context window with YaRN using a scaling factor of 4 [22], while the teacher uses its native context window. For each prompt, the student generates eight responses, the

teacher supplies the token-level log-probabilities used to compute $s^{(i)}$, and the task-specific verifier assigns $R^{(i)}$.

For a prompt $\mathbf{x}$, let $\mathcal{P}(\mathbf{x}) = \{(i, j) : R^{(i)} > R^{(j)}\}$ contain all response pairs with distinct verifier rewards, oriented toward the higher-reward response. We normalize the trajectory-level OPD scores to zero mean and unit variance within the rollout group and denote them by $\tilde{s}^{(i)}$. Let $\mathbb{I}(\cdot)$ denote the indicator function. The pairwise disagreement rate is

$$d_{\text{pair}}(\mathbf{x}) = \frac{1}{|\mathcal{P}(\mathbf{x})|} \sum_{(i,j)\in\mathcal{P}(\mathbf{x})} \mathbb{I}(\tilde{s}^{(i)} < \tilde{s}^{(j)}), \tag{4}$$

and the OPD preference gap is

$$g_{\text{OPD}}(\mathbf{x}) = \frac{1}{|\mathcal{P}(\mathbf{x})|} \sum_{(i,j)\in\mathcal{P}(\mathbf{x})} (\tilde{s}^{(i)} - \tilde{s}^{(j)}). \tag{5}$$

The first metric measures how often the OPD ordering disagrees with the verifier ordering. The second captures the direction and magnitude of OPD preference along the verifier ordering, measured in units of the within-group OPD standard deviation. Positive gaps indicate agreement with the verifier ordering, whereas negative gaps indicate greater OPD support for lower-reward responses. We macro-average both metrics over prompts whose eight responses contain at least two distinct verifier rewards.

Figure 1 shows that the pairwise disagreement rate increases over longer prompt-length ranges in both tasks, while the OPD preference gap declines from positive to negative. For Multi-Table Extraction, the disagreement rate rises from 40.6% below 8K to 64.0% at 32–64K, and the preference gap declines from $+0.35$ to -0.37. For High-Recall Retrieval, the corresponding values change from 35.2% to 60.2% and from $+0.65$ to -0.35. Within these two tasks, the consistent trends indicate that longer inputs can amplify teacher–verifier disagreement, motivating verifier-based calibration of dense OPD guidance.

4 Method

4.1 Overview

Group-Calibrated On-Policy Distillation (GC-OPD) calibrates vanilla OPD with response-level verifier feedback while retaining its dense token-level guidance. As illustrated in Figure 2, GC-OPD first normalizes verifier rewards and trajectory-level OPD scores within each rollout group, then represents their difference as a signed teacher–verifier disagreement residual. Relative-advantage-based credit assignment (RACA) distributes this trajectory-level residual across tokens according to their relative OPD advantages, and the resulting correction is added to the original token-level OPD advantage.

4.2 Group-Relative Assessments

For each response $\mathbf{y}^{(i)} \in \mathcal{G}(\mathbf{x})$, vanilla OPD supplies a token-level OPD advantage $A_t^{(i)}$ for every response token. Equation 3 aggregates these advantages into the trajectory-level OPD score $s^{(i)}$, while the verifier assigns the response a reward $R^{(i)}$. These quantities capture different properties of the response. The trajectory-level OPD score summarizes the teacher's average support relative to the frozen rollout policy, whereas the verifier reward measures verified task outcome.

Since the two signals differ in their raw scales, GC-OPD normalizes them separately within each rollout group. Let $z(\cdot)$ denote within-group z-score normalization, which subtracts the within-group mean and divides by the within-group standard deviation. We define the group-normalized verifier reward and group-normalized trajectory-level OPD score as

$$\begin{aligned} \tilde{R}^{(i)} &= z\left(R^{(i)}\right), \\ \tilde{s}^{(i)} &= z\left(s^{(i)}\right). \end{aligned} \tag{6}$$

Normalizing the two signals separately within each rollout group avoids directly comparing their raw values. For graded verifier rewards, $\tilde{R}^{(i)}$ retains the relative spacing of outcomes within the group. Binary verifier rewards, by contrast, indicate only whether each response succeeds or fails. GC-OPD operates on these group-normalized values rather than their induced rankings alone, allowing the magnitude of graded outcome differences to inform the residual.

4.3 Teacher–Verifier Disagreement Residual

Given the group-normalized quantities in Equation 6, we define the signed residual as

$$\rho^{(i)} = \tilde{R}^{(i)} - \tilde{s}^{(i)}. \tag{7}$$

The sign of $\rho^{(i)}$ indicates the direction of teacher–verifier disagreement. A positive value means that the group-normalized verifier reward for response $\mathbf{y}^{(i)}$ exceeds its group-normalized OPD score, whereas a negative value indicates the opposite. Its magnitude reflects the difference between the two group-normalized assessments.

GC-OPD uses this difference instead of directly adding the group-normalized verifier reward. Direct reward addition can reinforce a relative preference already expressed by OPD. In contrast, the residual vanishes when the two group-normalized assessments coincide, while its magnitude grows with their discrepancy. The residual therefore focuses calibration on differences between the verifier and OPD assessments.

The residual also has the desired ordering under a strict disagreement between the verifier and OPD orderings. For two responses i and j in the same rollout group, let σ_R and σ_s denote the positive within-group standard deviations. If $R^{(i)} > R^{(j)}$ but $s^{(i)} < s^{(j)}$, then

$$\rho^{(i)} - \rho^{(j)} = \frac{R^{(i)} - R^{(j)}}{\sigma_R} - \frac{s^{(i)} - s^{(j)}}{\sigma_s} > 0. \tag{8}$$

Thus, when the verifier prefers response i but OPD prefers response j, the residual assigns a larger value to response i. Conversely, reversing the two preferences reverses the residual ordering. This verifier-consistent ordering directs the subsequent token-level correction toward the response with the higher verifier reward.

If either signal has a near-zero within-group standard deviation, the corresponding normalization is not well-defined. We therefore set $\rho^{(i)} = 0$ for all responses in that group, reducing GC-OPD to vanilla OPD for the group.

4.4 Relative-Advantage-based Credit Assignment

The trajectory-level residual specifies the direction and magnitude of calibration for each response, but not how the correction should be distributed across tokens. RACA uses the token-level OPD advantages within each response as the allocation signal. We first normalize each token's OPD advantage relative to the response mean:

$$u_t^{(i)} = \frac{A_t^{(i)} - s^{(i)}}{\sqrt{\frac{1}{T^{(i)}} \sum_{v=1}^{T^{(i)}} \left(A_v^{(i)} - s^{(i)} \right)^2 + \epsilon}}, \tag{9}$$

where ϵ is a small positive constant for numerical stability. The resulting relative OPD advantage $u_t^{(i)}$ is positive for tokens whose OPD advantages exceed the response mean and negative for those below it. This normalization preserves the within-response ordering of the signed OPD advantages. Here, $u_t^{(i)}$ serves only as an allocation signal and should not be interpreted as token correctness or causal importance.

RACA maps the relative OPD advantage to a positive, bounded token credit:

$$c_t^{(i)} = 1 + \tanh\left(\frac{u_t^{(i)}}{2} \right). \tag{10}$$

This mapping is monotonic and satisfies $c_t^{(i)} \in (0, 2)$. Tokens with larger relative OPD advantages therefore receive larger credits, while $u_t^{(i)} = 0$ yields unit credit $c_t^{(i)} = 1$. Because the credit is always positive, it scales but does not reverse the sign of the trajectory-level residual.

The resulting GC-OPD advantage is

$$A_t'^{(i)} = A_t^{(i)} + \beta c_t^{(i)} \rho^{(i)}, \tag{11}$$

where $\beta \geq 0$ is the residual coefficient that controls the strength of calibration. The original token-level OPD advantage remains the base term, and $\beta c_t^{(i)} \rho^{(i)}$ is the token-level residual correction. GC-OPD substitutes $A_t'^{(i)}$ for $A_t^{(i)}$ in the clipped token-level policy objective described in Section 3.1, changing the advantage construction without modifying the actor objective. Setting $\beta = 0$ recovers vanilla OPD. Given the OPD advantages and verifier rewards already produced by the training pipeline, GC-OPD adds only group-level aggregation, normalization, and elementwise token transformations. It requires no additional teacher or student forward pass.

5 Experiments

We organize our experiments around three questions. First, does GC-OPD improve upon vanilla OPD under the same long-context training and evaluation setup? Second, does residualizing group-normalized verifier feedback improve upon adding that feedback directly? Third, does relative-advantage-based credit assignment (RACA) improve upon uniform residual allocation?

5.1 Experimental Setup

Models. We use Qwen3-4B[3] and Qwen3-8B[4] as student models and Qwen3-30B-A3B-Thinking-2507[5] as the teacher model. Both students operate in no-thinking mode. The teacher provides token-level log probabilities for OPD training.

Training data. We construct the training set from GoLongRL [20] by retaining prompts no longer than 32K tokens. The resulting subset contains 9,527 prompts across nine task families. Table 1 summarizes their distribution and native verifier metrics. Three task families use binary rewards, while the remaining six use graded rewards. The two largest families, precise long-range retrieval and evidence-grounded reasoning, account for 82.9% of the training set.

Table 1: Composition and reward interfaces of the 9,527-prompt GoLongRL subset after the 32K filter. "B" denotes a binary reward in $\{0, 1\}$; "G" denotes a graded score in $[0, 1]$.

Task Family	# Samples	Native Verifier
Precise Long-Range Retrieval	4,693	Exact Match (B)
Evidence-Grounded Reasoning	3,204	Accuracy (B)
Multi-Table Extraction	729	IoU (G)
High-Recall Retrieval	540	Set F1 (G)
Fragment Matching and Induction	148	Subset EM (G)
Sequence Reconstruction	69	Pairwise Acc. (G)
Numerical Reasoning	54	Math Verify (B)
Graded Retrieval and Ranking	45	NDCG (G)
Long-Document Summarization	45	ROUGE-L (G)

Training configuration. All post-training methods use the same 100-step optimization budget. At each step, we sample a batch of 32 prompts and generate eight student responses per prompt. The maximum prompt length is 32,768 tokens, and response generation is capped at 10,240 tokens. Complete optimization and implementation details are provided in the supplementary material.

[3] `https://huggingface.co/Qwen/Qwen3-4B`

[4] `https://huggingface.co/Qwen/Qwen3-8B`

[5] `https://huggingface.co/Qwen/Qwen3-30B-A3B-Thinking-2507`

Evaluation. We evaluate DocMath [35], Frames [14], MRCR [24], CorpusQA [19], and LBv1QA [3], covering numerical reasoning, multi-hop synthesis, multi-round co-reference, corpus aggregation, and long-context question answering, respectively. Following QwenLong-L1 [25], each student receives at most 120,000 input tokens and generates at most 8,192 tokens within a 131,072-token serving context. We extend the context window with YaRN [22] using a scaling factor of 4. All students remain in no-thinking mode and use the same evaluation prompts and decoding configuration. We report the score on each benchmark and the unweighted mean across the five benchmarks.

Baselines and comparison settings. Raw denotes the corresponding officially released Qwen3 checkpoint evaluated before any post-training in our pipeline [30]. We compare GC-OPD with Raw, vanilla OPD [1], ExOPD [31], Uni-OPD [9], FiRe-OPD [16], and PowerOPD [34]. Vanilla OPD serves as the dense-distillation control. ExOPD introduces teacher–reference extrapolation, Uni-OPD performs outcome-conditioned margin calibration, FiRe-OPD filters trajectories and reweights token-level signals, and PowerOPD bounds token-level OPD rewards for stability. Daggered rows in Table 2 implement each method's principal training-signal mechanism under our shared long-context setup. To isolate these mechanisms, the runs use the same student initialization, teacher, training data, rollout configuration, 100-step training budget, and evaluation pipeline as GC-OPD.

Residual-coefficient selection. We select a single residual coefficient for both model scales using a fixed GoLongRL holdout. Before constructing the training subset, we reserve the first 256 examples in the ordered GoLongRL shards. Applying the same 32K-token limit leaves 231 validation examples with no overlap with the 9,527 training examples. Figure 3 reports the reward at the final checkpoint and the mean over the last five validation checkpoints, using one stochastic response per example. Both summaries select $\beta = 0.10$ for Qwen3-4B and Qwen3-8B, so we use this shared value in the main experiments. Because the holdout contains only the High-Recall Retrieval task family, we use it exclusively for coefficient selection rather than downstream evaluation.

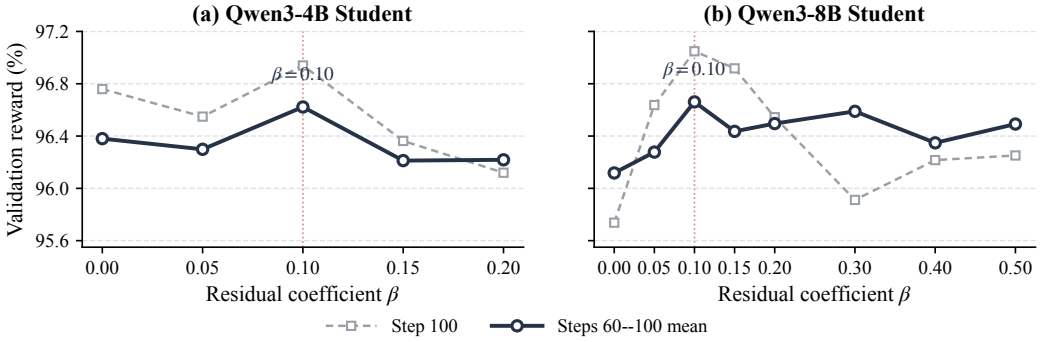

Figure 3: Held-out validation used to select the residual coefficient β. The fixed set contains 231 GoLongRL examples after 32K-token filtering and has no overlap with training. White squares show the step-100 reward. Circles show the mean over steps 60, 70, 80, 90, and 100. Both summaries select $\beta = 0.10$ at both model scales.

5.2 Main Results

GC-OPD improves performance across model scales. As shown in Table 2, GC-OPD achieves the highest five-task average for both students. Relative to vanilla OPD, it raises the average score from 39.31 to 40.47 for Qwen3-4B and from 43.56 to 44.65 for Qwen3-8B. It also achieves the highest average among the evaluated shared-setup implementations at both model scales. Competing methods often lead on individual benchmarks but do not maintain the strongest aggregate performance. GC-OPD therefore provides a more favorable balance across the heterogeneous evaluation suite rather than relying on a single task.

Gains concentrate on structured reasoning and evidence aggregation. Relative to vanilla OPD, GC-OPD improves DocMath, MRCR, and CorpusQA for both students, with the largest gains on CorpusQA. These benchmarks emphasize structured reasoning and evidence aggregation. The concentration of gains is consistent with the intended role of response-level verifier calibration,

Table 2: Qwen3-4B and Qwen3-8B results on five long-context benchmarks (%). "Avg." is the unweighted five-task mean. † denotes an implementation of the method's principal training-signal mechanism under the shared setup.

Model	Method	Benchmark					
		Avg.	**DocMath**	**Frames**	**MRCR**	**CorpusQA**	**LBv1QA**
Qwen3-4B	Raw	29.08	43.63	26.58	22.02	6.99	46.20
	OPD	39.31	49.37	<u>30.95</u>	26.90	32.22	**57.10**
	ExOPD†	38.22	47.75	29.61	25.95	31.91	55.90
	Uni-OPD†	38.53	**51.88**	29.98	27.93	27.36	55.50
	PowerOPD†	38.88	49.63	30.10	**28.36**	<u>32.52</u>	53.80
	FiRe-OPD†	<u>39.50</u>	49.37	**31.31**	<u>28.20</u>	<u>32.52</u>	<u>56.10</u>
	GC-OPD (ours)	**40.47**	<u>50.38</u>	30.34	27.82	**37.99**	55.80
Qwen3-8B	Raw	35.12	45.88	30.95	23.87	22.19	52.70
	OPD	43.56	<u>55.13</u>	**34.59**	30.44	39.82	57.80
	PowerOPD†	41.53	51.38	32.65	<u>31.65</u>	37.39	54.60
	Uni-OPD†	43.41	54.13	**34.59**	25.25	<u>42.86</u>	<u>60.20</u>
	ExOPD†	43.49	53.12	33.50	**32.73**	37.69	**60.40**
	FiRe-OPD†	<u>44.01</u>	54.00	<u>34.34</u>	30.96	41.64	59.10
	GC-OPD (ours)	**44.65**	**55.50**	**34.59**	31.10	**43.77**	58.30

although the table alone does not isolate task-specific mechanisms. Frames and LBv1QA show smaller or model-dependent changes, so the benefit is not uniform across tasks. Improvements over Raw confirm the effectiveness of the complete long-context post-training pipeline, whereas comparisons with vanilla OPD isolate the additional contribution of GC-OPD. The following ablations separately test the signed residual and its token-allocation rule.

5.3 Ablation Studies

We next examine the two components introduced by GC-OPD: the signed residual and RACA token allocation. All ablations use independently trained Qwen3-8B students under the shared setup and 100-step budget. We first vary the signal added to vanilla OPD while holding the coefficient and token-allocation rule fixed. We then examine how the residual should be distributed across tokens. Detailed formulations of all ablation variants are provided in Appendix C.2.

Residualization improves upon OPD-only and direct-reward controls. Table 3 isolates the signal added to vanilla OPD. The three augmented variants share RACA and $\beta = 0.10$, so only the added term changes:

$$A_t'^{(i)} = A_t^{(i)} + \begin{cases} 0, & \text{Vanilla OPD,} \\ \beta c_t^{(i)} A_t^{(i)}, & \text{Additional OPD,} \\ \beta c_t^{(i)} \tilde{R}^{(i)}, & \text{Direct reward,} \\ \beta c_t^{(i)} \rho^{(i)}, & \text{GC-OPD.} \end{cases} \tag{12}$$

Additional OPD changes the average score only from 43.56 to 43.60, indicating that another OPD-derived term does not explain the improvement. Direct reward reaches 44.19, confirming that group-normalized verifier feedback complements dense OPD guidance. Replacing the direct reward with the signed residual further raises the average score to 44.65 under the same coefficient and token-allocation rule. This additional gain supports accounting for the relative preference already expressed by OPD rather than adding verifier feedback alone.

RACA improves residual allocation. Having established the contribution of the residual, we next examine its token allocation. Table 4 uses vanilla OPD as the no-residual anchor. All residual variants use $A_t'^{(i)} = A_t^{(i)} + 0.10 c_t^{(i)} \rho^{(i)}$ and differ only in $c_t^{(i)}$. Uniform sets $c_t^{(i)} = 1$. Absolute OPD assigns credit using

$$c_t^{(i)} = \mathrm{clip}_{[0,5]} \left(\frac{|A_t^{(i)}|}{(T^{(i)})^{-1} \sum_{v=1}^{T^{(i)}} |A_v^{(i)}|} \right), \tag{13}$$

Table 3: Response-level signal ablation on Qwen3-8B. All augmented variants use RACA with coefficient $\beta = 0.10$ and differ only in whether the added signal is $A_t^{(i)}$, $\tilde{R}^{(i)}$, or $\rho^{(i)} = \tilde{R}^{(i)} - \tilde{s}^{(i)}$. "Avg." is the unweighted five-task mean, and "Δ" is computed from the unrounded average relative to vanilla OPD.

Variant	Avg.	DocMath	Frames	MRCR	CorpusQA	LBv1QA	Δ
Vanilla OPD	43.56	55.13	34.59	30.44	39.82	57.80	–
Additional OPD	43.60	54.63	34.59	30.03	39.51	59.20	+0.04
Direct reward	44.19	55.13	34.59	30.40	40.73	**60.10**	+0.63
GC-OPD	**44.65**	**55.50**	34.59	**31.10**	**43.77**	58.30	**+1.10**

whereas RACA uses Equation 10.

Uniform allocation raises the average score from 43.56 to 44.28, showing that the signed residual is useful even without token-dependent credit. RACA further raises it to 44.65, whereas Absolute OPD reaches only 43.93. Because Absolute OPD discards the sign of the OPD advantage, it can assign large credits to strongly negative OPD values. This result supports preserving the signed within-response ordering used by RACA.

Table 4: Token-credit ablation on Qwen3-8B. All residual variants use the same signed residual with $\beta = 0.10$ and differ only in token allocation. "Avg." is the unweighted five-task mean, and "Δ" is computed from the unrounded average relative to vanilla OPD.

Token allocation	Avg.	DocMath	Frames	MRCR	CorpusQA	LBv1QA	Δ
Vanilla OPD	43.56	55.13	34.59	30.44	39.82	57.80	–
Absolute OPD	43.93	54.63	35.56	29.53	41.34	58.60	+0.38
Uniform	44.28	54.12	**37.01**	30.64	40.12	**59.50**	+0.72
RACA	**44.65**	**55.50**	34.59	**31.10**	**43.77**	58.30	**+1.10**

5.4 Disagreement Across Task Families

Beyond downstream performance, we use a fixed response set to test whether teacher–verifier disagreement extends beyond the two tasks in Figure 1. This diagnostic measures disagreement prevalence rather than trained-checkpoint performance.

Teacher–verifier disagreement extends across task families. Figure 4 in Appendix A reports pairwise disagreement and top-1 mismatch for four task families containing at least 500 prompts, together with a prompt-macro aggregate over all nine families. The Multi-Table Extraction and High-Recall Retrieval rows aggregate the same frozen responses used in Figure 1 over prompts shorter than 32K. The additional task rows show that disagreement extends beyond these two tasks, while the variation across rows indicates that its prevalence remains task dependent.

6 Conclusion

We introduced GC-OPD to address teacher–verifier disagreement in long-context OPD. GC-OPD forms a signed residual between group-normalized verifier rewards and trajectory-level OPD scores, then uses RACA to distribute it across tokens. This preserves dense OPD guidance while supporting binary and graded rewards without cross-task calibration.

Diagnostics reveal stronger disagreement over longer inputs in two tasks and task-dependent prevalence across the training mixture. Across five benchmarks, GC-OPD improves vanilla OPD from 39.31 to 40.47 for Qwen3-4B and from 43.56 to 44.65 for Qwen3-8B. Ablations identify residual calibration as the main contributor, with an additional gain from RACA over uniform allocation. Overall, GC-OPD integrates response-level verification with dense token-level guidance.

References

[1] 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 *International Conference on Learning Representations*, volume 2024, pages 21246–21263, 2024.

[2] Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. Longalign: A recipe for long context alignment of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1376–1395, 2024.

[3] Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3119–3137, 2024.

[4] Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, et al. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3639–3664, 2025.

[5] Guanzheng Chen, Michael Qizhe Shieh, and Lidong Bing. LongRLVR: Long-context reinforcement learning requires verifiable context rewards. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=omVhYvyTPJ.

[6] Omer Goldman, Alon Jacovi, Aviv Slobodkin, Aviya Maimon, Ido Dagan, and Reut Tsarfaty. Is it really long context if all you need is retrieval? towards genuinely difficult long context nlp. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16576–16586, 2024.

[7] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pages 32694–32717, 2024.

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[9] Wenjin Hou, Shangpin Peng, Weinong Wang, Zheng Ruan, Yue Zhang, Zhenglin Zhou, Mingqi Gao, Yifei Chen, Kaiqi Wang, Hongming Yang, et al. Uni-opd: Unifying on-policy distillation with a dual-perspective recipe. *arXiv preprint arXiv:2605.03677*, 2026.

[10] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

[11] Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining credit assignment in RL training of LLMs. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 29557–29590. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/kazemnejad25a.html.

[12] Minsang Kim and Seung Baek. Explain in your own words: Improving reasoning via token-selective dual knowledge distillation. In *International Conference on Learning Representations*, volume 2026, pages 103174–103196, 2026.

[13] Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 24872–24895. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/ko24c.html.

[14] Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4745–4759, 2025.

[15] Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026.

[16] Yuying Li, Leqi Zheng, Yongzi Yu, Wenrui Zhou, Xuchang Zhong, Xing Hu, Jing Jin, Hangjie Yuan, and Tao Feng. Filter, then reweight: Rethinking optimization granularity in on-policy distillation. *arXiv preprint arXiv:2606.02684*, 2026.

[17] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601, 2024.

[18] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

[19] Zhiyuan Lu, Chenliang Li, Yingcheng Shi, Weizhou Shen, Ming Yan, and Fei Huang. Corpusqa: A 10 million token benchmark for corpus-level analysis and reasoning. *arXiv preprint arXiv:2601.14952*, 2026.

[20] Minxuan Lv, Tiehua Mei, Tanlong Du, Junmin Chen, Zhenpeng Su, Ziyang Chen, Ziqi Wang, Zhennan Wu, Ruotong Pan, Ruiming Tang, et al. Golongrl: Capability-oriented long context reinforcement learning with multitask alignment. *arXiv preprint arXiv:2605.19577*, 2026.

[21] Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A. Rossi, Seunghyun Yoon, and Hinrich Schuetze. NoLiMa: Long-context evaluation beyond literal matching. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 44554–44570. PMLR, 13–19 Jul 2025. URL `https://proceedings.mlr.press/v267/modarressi25a.html`.

[22] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. In *International Conference on Learning Representations*, volume 2024, pages 31932–31951, 2024.

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

[24] Kiran Vodrahalli, Santiago Ontanon, Nilesh Tripuraneni, Kelvin Xu, Sanil Jain, Rakesh Shivanna, Jeffrey Hui, Nishanth Dikkala, Mehran Kazemi, Bahare Fatemi, et al. Michelangelo: Long context evaluations beyond haystacks via latent structure queries. *arXiv preprint arXiv:2409.12640*, 2024.

[25] Fanqi Wan, Weizhou Shen, Shengyi Liao, Yingcheng Shi, Chenliang Li, Ziyi Yang, Ji Zhang, Fei Huang, Jingren Zhou, and Ming Yan. Qwenlong-l1: Towards long-context large reasoning models with reinforcement learning. *arXiv preprint arXiv:2505.17667*, 2025.

[26] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, 2024.

[27] Siyuan Wang, Gaokai Zhang, Li Lyna Zhang, Ning Shang, Fan Yang, Dongyao Chen, and Mao Yang. Loongrl: Reinforcement learning for advanced reasoning over long contexts. In *International Conference on Learning Representations*, volume 2026, pages 140078–140097, 2026.

[28] Yuanyi Wang, Su Lu, Yanggan Gu, Pengkai Wang, Yifan Yang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, and Hongxia Yang. Not all disagreement is learnable: Token teachability in on-policy distillation. *arXiv preprint arXiv:2605.26844*, 2026.

[29] Haoran Xu, Hongyu Wang, Yifei Gao, Jiaze Li, Xiaofeng Zhang, and Xiaosong Yuan. Sg-opd: Sign-gated on-policy distillation via sign-consistency gating and phased teacher sampling. *arXiv preprint arXiv:2606.09304*, 2026.

[30] 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.

[31] 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.

[32] Weichen Yu, Xiaomin Li, Yizhou Zhao, Xiaoze Liu, Ruowang Zhang, Haixin Wang, Yinyi Luo, Chen Henry Wu, Gaurav Mittal, Matt Fredrikson, et al. Multi-rollout on-policy distillation via peer successes and failures. *arXiv preprint arXiv:2605.12652*, 2026.

[33] Jiajie Zhang, Zhongni Hou, Xin Lv, Shulin Cao, Zhenyu Hou, Yilin Niu, Lei Hou, Yuxiao Dong, Ling Feng, and Juanzi Li. Longreward: Improving long-context large language models with ai feedback. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3718–3739, 2025.

[34] Anhao Zhao, Junlong Tong, Yingqi Fan, Ping Nie, Wenjie Li, and Xiaoyu Shen. Poweropd: Stabilizing on-policy distillation with bounded power transformation. *arXiv preprint arXiv:2606.17199*, 2026.

[35] Yilun Zhao, Yitao Long, Hongjun Liu, Ryo Kamoi, Linyong Nan, Lyuhao Chen, Yixin Liu, Xiangru Tang, Rui Zhang, and Arman Cohan. Docmath-eval: Evaluating math reasoning capabilities of llms in understanding long and specialized documents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16103–16120, 2024.

[36] Binbin Zheng, Xing Ma, Yiheng Liang, Jingqing Ruan, Xiaoliang Fu, Kepeng Lin, Benchang Zhu, Ke Zeng, and Xunliang Cai. Scope: Signal-calibrated on-policy distillation enhancement with dual-path adaptive weighting. *arXiv preprint arXiv:2604.10688*, 2026.

[37] Qingyun Zou, Yingze Li, Tianen Liu, Bingsheng He, and Weng-Fai Wong. Reward-weighted on-policy distillation with an open property-equivalence verifier for nl-to-sva generation. *arXiv preprint arXiv:2605.13501*, 2026.

A Task-Conditioned Disagreement Analysis

The main paper analyzes how teacher–verifier disagreement changes with prompt length in two representative tasks. Here, we complement that analysis with a task-conditioned view of four GoLongRL task families containing at least 500 prompts. For Multi-Table Extraction and High-Recall Retrieval, we aggregate the same frozen responses used in Figure 1 over prompts shorter than 32K, without separating the two length ranges. Each of the other task rows is likewise computed from eight fixed Qwen3-8B responses per prompt. Overall pools all nine task families by giving each informative prompt equal weight.

Within each informative rollout group, the pairwise disagreement rate is the fraction of response pairs with distinct verifier rewards for which the OPD and verifier orderings disagree. We average this rate across prompts. The top-1 mismatch rate is the fraction of these groups in which the response with the highest trajectory-level OPD score does not attain the group's highest verifier reward. Their variation across task families complements the length-conditioned trends in the main paper and motivates comparing the two assessments locally within each rollout group. Because the responses are fixed, the analysis characterizes signal disagreement rather than task-specific training gains or causal effects of task type or context length.

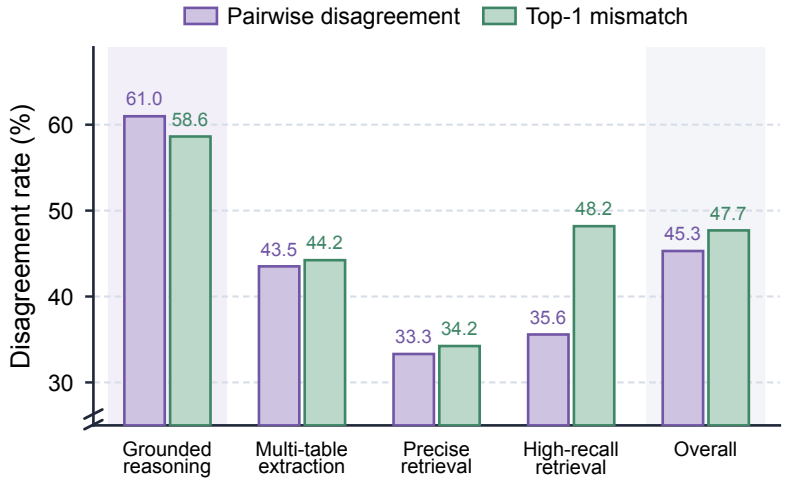

Figure 4: Task-conditioned teacher–verifier disagreement in four GoLongRL task families. Bars report prompt-macro pairwise disagreement rates and top-1 mismatch rates over rollout groups with at least two distinct verifier rewards. Overall pools all nine task families.

B GC-OPD Implementation Details

Using the notation of the main paper, $A_t^{(i)}$ is the detached token-level OPD advantage, $s^{(i)} = (T^{(i)})^{-1} \sum_t A_t^{(i)}$ is its trajectory-level OPD score, and $R^{(i)}$ is the verifier reward. The main paper defines the compact pre-clipping update $A_t'^{(i)}$. Here, $T^{(i)}$ is the number of valid tokens in response i, G is the rollout-group size, and β is the residual coefficient in the main update. This section specifies the numerical guards and update order used in the completed runs. The thresholds τ_G and τ_T guard group-level and token-level standard deviations, respectively, and $a_{\max}$ is the final advantage-clipping bound. For $h \in \{R, s\}$, let μ_h and σ_h denote the population mean and standard deviation over the G responses. The implemented residual is

$$\rho^{(i)} = \begin{cases} \frac{R^{(i)}-\mu_R}{\sigma_R} - \frac{s^{(i)}-\mu_s}{\sigma_s}, & \sigma_R, \sigma_s > \tau_G, \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

If either group-level signal has negligible variation, the residual is zero and the update reduces to vanilla OPD.

The token credit uses an analogous guard. Within response i, let $\sigma_A^{(i)}$ denote the population standard deviation of its valid token advantages. The guarded RACA credit is

$$u_t^{(i)} = \begin{cases} \dfrac{A_t^{(i)} - s^{(i)}}{\sigma_A^{(i)}}, & T^{(i)} \geq 2 \text{ and } \sigma_A^{(i)} > \tau_T, \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

$$c_t^{(i)} = 1 + \tanh\left(\frac{u_t^{(i)}}{2}\right), \tag{16}$$

Here, $u_t^{(i)}$ is the token's OPD advantage relative to the response mean, measured in units of the within-response standard deviation. It describes relative OPD support rather than token correctness. Thus negligible token variation gives unit credit. After forming $A_t'^{(i)}$, we clip it to $[-a_{\max}, a_{\max}]$ before the policy update. Table 6 reports τ_G, τ_T, and $a_{\max}$. These Given the OPD advantages and verifier rewards, these operations require no additional teacher or student forward pass.

Algorithm 1 Group-Calibrated On-Policy Distillation

Require: Prompt batch $\mathcal{B}$; trainable student policy π_θ; teacher π_T; verifier V; group size G; residual coefficient β; thresholds τ_G, τ_T; clipping bound $a_{\max}$

Ensure: Updated student policy π_θ

Set $\theta_{\text{old}} \leftarrow \theta$ and freeze $\pi_{\theta_{\text{old}}}$ for rollout generation
for each prompt $\mathbf{x} \in \mathcal{B}$ **do**
  Sample $\mathcal{G}(\mathbf{x}) = \{\mathbf{y}^{(i)}\}_{i=1}^{G}$ from $\pi_{\theta_{\text{old}}}(\cdot \mid \mathbf{x})$
  for $i = 1, \ldots, G$ **do**
    $R^{(i)} \leftarrow V(\mathbf{x}, \mathbf{y}^{(i)})$
    Compute detached $A_t^{(i)}$ under π_T and $\pi_{\theta_{\text{old}}}$ for $t = 1, \ldots, T^{(i)}$
    $s^{(i)} \leftarrow (T^{(i)})^{-1} \sum_{t=1}^{T^{(i)}} A_t^{(i)}$
  end for
  Compute population statistics (μ_R, σ_R) and (μ_s, σ_s) over the G responses
  if $\sigma_R > \tau_G \wedge \sigma_s > \tau_G$ **then**
    $\rho^{(i)} \leftarrow \dfrac{R^{(i)} - \mu_R}{\sigma_R} - \dfrac{s^{(i)} - \mu_s}{\sigma_s}$ for $i = 1, \ldots, G$
  else
    $\rho^{(i)} \leftarrow 0$ for all i
  end if
  for each response i **do**
    Compute $\sigma_A^{(i)}$ from $\{A_t^{(i)}\}_{t=1}^{T^{(i)}}$
    if $T^{(i)} \geq 2 \wedge \sigma_A^{(i)} > \tau_T$ **then**
      $u_t^{(i)} \leftarrow \dfrac{A_t^{(i)} - s^{(i)}}{\sigma_A^{(i)}}$ for $t = 1, \ldots, T^{(i)}$
      $c_t^{(i)} \leftarrow 1 + \tanh\left(u_t^{(i)}/2\right)$ for $t = 1, \ldots, T^{(i)}$
    else
      $u_t^{(i)} \leftarrow 0$ and $c_t^{(i)} \leftarrow 1$ for $t = 1, \ldots, T^{(i)}$
    end if
    $A_t'^{(i)} \leftarrow A_t^{(i)} + \beta c_t^{(i)} \rho^{(i)}$ for $t = 1, \ldots, T^{(i)}$
    $\widehat{A}_t^{(i)} \leftarrow \text{clip}(A_t'^{(i)}, -a_{\max}, a_{\max})$ for all t
  end for
end for
Update π_θ using the same policy surrogate as vanilla OPD, with $\widehat{A}_t^{(i)}$ as the token-level advantages

C Experimental Details

C.1 Models, Data, and Training Configuration

We initialize Qwen3-4B and Qwen3-8B students from their official checkpoints and use Qwen3-30B-A3B-Thinking-2507 as the teacher. Students generate in no-thinking mode, and the teacher scores the sampled response tokens without generating separate responses.

The 32K-filtered GoLongRL training file contains 9,527 prompts. Table 5 reports its prompt-length distribution, complementing the task-family and verifier statistics in the main paper. The median, 90th percentile, and maximum lengths are 9,923, 26,940, and 32,766 tokens.

Table 6 records the common configuration. Raw rows receive no training; every trained method uses this configuration and is evaluated at its final checkpoint.

Compute resources. Each training run used eight 80-GB NVIDIA H800 or H100 GPUs. Rollout tensor parallelism and actor/reference sequence parallelism were both set to 8.

C.2 Ablation Configurations

The ablations in the main paper use independent Qwen3-8B training runs under the shared configuration in Table 6. All runs use the same student initialization, teacher, filtered training data, optimizer, rollout budget, and 100-step training horizon. They are evaluated at step 100 with the same five-benchmark pipeline as the main comparison.

Added-signal ablation. Vanilla OPD provides the common anchor for this comparison. The three augmented variants use the same RACA credit and a scalar coefficient of 0.10, while changing the signal added to the vanilla OPD advantage. Their updates before final advantage clipping are

$$A_t'^{(i)} = A_t^{(i)} \qquad \text{(vanilla OPD)}, \tag{17}$$

$$A_t'^{(i)} = A_t^{(i)} + \beta c_t^{(i)} A_t^{(i)}, \quad \beta = 0.10 \qquad \text{(Additional OPD)}, \tag{18}$$

$$A_t'^{(i)} = A_t^{(i)} + \beta c_t^{(i)} \tilde{R}^{(i)}, \quad \beta = 0.10 \qquad \text{(Direct reward)}, \tag{19}$$

$$A_t'^{(i)} = A_t^{(i)} + \beta c_t^{(i)} \rho^{(i)}, \quad \beta = 0.10 \qquad \text{(GC-OPD)}. \tag{20}$$

The comparison holds the original OPD signal, RACA allocation, and coefficient fixed. Additional OPD adds another credit-modulated OPD term, Direct reward uses the group-normalized verifier reward $\tilde{R}^{(i)}$, whereas GC-OPD subtracts the group-normalized trajectory OPD score and uses $\rho^{(i)} = \tilde{R}^{(i)} - \tilde{s}^{(i)}$. It therefore separates the effect of adding another OPD-derived term from adding verifier feedback, and tests whether accounting for the relative preference already expressed by OPD improves upon adding verifier feedback directly.

Token-allocation ablation. This comparison fixes the signed residual and the same $\beta = 0.10$ used by the main configuration, so every calibrated variant uses $A_t'^{(i)} = A_t^{(i)} + 0.10 c_t^{(i)} \rho^{(i)}$. Only the token credit $c_t^{(i)}$ changes. Uniform allocation sets $c_t^{(i)} = 1$. Absolute OPD is a token-allocation control, not an alternative trajectory-level signal. It assigns credit in proportion to the magnitude of the tokenwise OPD advantage:

$$c_t^{(i)} = \mathrm{clip}_{[0,5]}\left(\frac{|A_t^{(i)}|}{(T^{(i)})^{-1} \sum_{v=1}^{T^{(i)}} |A_v^{(i)}|} \right), \tag{21}$$

The upper bound of 5 limits extreme token weights, and unit credit is used when the denominator is numerically negligible. RACA uses Equation 16. Thus this comparison holds the trajectory-level residual and its coefficient fixed while isolating how the correction is distributed across response tokens.

C.3 Baselines under the Shared Setup

All trained baselines share the student initialization, teacher, filtered training data, rollout budget, optimizer, training horizon, and clipped surrogate. We hold these factors fixed to isolate each

Table 5: Prompt-length distribution of the exact 32K-filtered training file.

Prompt length	Samples
$\leq$ 2K	2,222
2–8K	1,958
8–16K	2,155
16–24K	1,914
24–32K	1,278
Total	9,527

Table 6: Shared training configuration and GC-OPD settings.

Setting	Value
Training horizon	100 steps
Prompt batch size	32
Responses per prompt	8
Maximum prompt length	32,768
Training response cap	10,240
Learning rate	10^{-6}
Warmup	0
Weight decay	0.01
Policy mini-batch size	4
PPO epochs	1
PPO clip ratio	0.2
Loss aggregation	Token mean
Rollout temperature	1
Rollout top-p	1
Precision	bfloat16
Random seed	42
Residual coefficient	$\beta = 0.10$
Group std. threshold	$\tau_G = 10^{-6}$
Token std. threshold	$\tau_T = 10^{-6}$
Final advantage clip	$[-10, 10]$ ($a_{\max} = 10$)

method's principal training-signal mechanism. The comparisons should therefore be interpreted as controlled mechanism comparisons under a shared setup rather than exact reproductions of the original experimental recipes. Each method generates its own trajectories on-policy. Let $p_{T,t}$, $p_{S,t}$, and $p_{0,t}$ denote the teacher, current-student, and frozen student-base probabilities of the sampled token. The paragraphs below record only method-specific implementation choices. To avoid clutter, the response index i is suppressed in baseline-specific token formulas unless it is needed explicitly.

Raw checkpoint. Raw evaluates the official Qwen3-4B or Qwen3-8B checkpoint without additional training, using the same downstream evaluation pipeline.

OPD. OPD uses $A_t^{\mathrm{OPD}} = \log p_{T,t} - \log p_{S,t}$ as the detached token advantage. The native verifier is executed by the shared pipeline, but its terminal score is not added to this advantage. Equivalently, vanilla OPD is recovered from GC-OPD by setting $\beta = 0$, and we use this convention throughout the supplement.

ExOPD. ExOPD replaces the teacher log-probability target with an extrapolated target anchored at the corresponding frozen student-base checkpoint:

$$A_t^{\mathrm{Ex}} = \lambda \log p_{T,t} + (1 - \lambda) \log p_{0,t} - \log p_{S,t}. \tag{22}$$

Our runs use $\lambda = 1.25$ and the corresponding frozen Qwen3 initialization as p_0, following the default student-base-reference formulation. The reference contributes only to the detached token target and is not used as a KL regularizer. In the ExOPD paper, "reward correction" denotes the optional replacement of p_0 with the teacher's pre-RL base model. This reference-based variant is distinct from the verifier reward used by GC-OPD and is not used here.

Uni-OPD. Uni-OPD combines token-level teacher supervision with response-level outcome separation. Within the eight responses for one prompt, responses with positive verifier reward form the correct set and the others form the incorrect set. Let $\bar{s}^+$ and $\bar{s}^-$ be the mean response-level OPD scores of these two sets. If both sets exist, the required shift is

$$d = \max\{0,\, \delta - (\bar{s}^+ - \bar{s}^-)\}. \tag{23}$$

With bidirectional calibration, $d/2$ is added to every token of a correct response and subtracted from every token of an incorrect response. Groups lacking either set receive no margin shift.

For a controlled comparison, we isolate Uni-OPD's outcome-guided margin calibration and omit its offline and online data-balancing components. This keeps the training data and rollout budget consistent across methods. The implementation uses group scope, mean statistics, $\delta = 0.4$, and bidirectional shifts. Following the released implementation, the resulting group-level shift is added to the original tokenwise OPD advantages. We also retain the released configuration's teacher–student log-probability-gap mask with a threshold of 10 as a training-stability setting.

PowerOPD. PowerOPD changes the geometry of the token signal from a log-probability gap to a bounded probability-power difference:

$$A_t^{\text{Power}} = \text{sg}\left[p_{T,t}^{\alpha} - p_{S,t}^{\alpha}\right], \qquad \alpha = 100, \tag{24}$$

where sg denotes stop-gradient. We use this bounded signal as the method-specific token advantage within the shared clipped surrogate. Both model scales use $\alpha = 100$.

FiRe-OPD. Following the released implementation, FiRe-OPD computes the length-normalized teacher score $q^{(i)} = (T^{(i)})^{-1} \sum_t \log p_{T,t}^{(i)}$ for each trajectory and applies the actor-micro-batch percentile filter. Token weights use teacher confidence and student confusion,

$$c_{T,t} = 1 - \frac{H_{T,t}}{\max H_T}, \qquad c_{S,t} = \frac{H_{S,t}}{\max H_S}, \tag{25}$$

clipped to $[0, 1]$. The OPD advantage is multiplied by $w_t = (1 + \alpha_F c_{T,t})(1 + \beta_F c_{S,t})$ after normalizing w to mean one over valid tokens. Filtered trajectories contribute no policy loss.

Both runs use a 20th-percentile trajectory cutoff and $\alpha_F = \beta_F = 1$. The percentile, entropy maxima, and weight normalization are recomputed within each actor micro-batch supplied to the loss. The actor starts from four-response policy mini-batches and may split them further under dynamic token budgeting.

C.4 Evaluation Implementation

Table 7 records the evaluation units and aggregation rules. The paragraphs below specify the per-example scorers.

Table 7: Evaluation units and aggregation for the five reported long-context benchmark columns.

Benchmark	Evaluation units	Reported aggregation
DocMath	Four splits: 200/100/200/300	Per-item max(rule, judge); four-split aggregate
FRAMES	824 questions	Mean max(CEM, judge) accuracy
MRCR	0–128K $\times$ {2, 4, 8} needles	Mean prefix-gated sequence ratio
CorpusQA	0–128K $\times$ four domains	Overall example accuracy
LBv1QA	Five subsets $\times$ 200	Macro mean of five subset accuracies

DocMath. The evaluator extracts the final answer and applies both the rule-based numerical/equation checker and a semantic-equivalence judge. The per-example score is the maximum of these binary decisions. Scores are then aggregated over the four simple/complex and short/long splits shown in Table 7.

FRAMES. After removing any hidden reasoning segment, the scorer applies cover exact match (CEM). CEM lowercases text, removes punctuation and articles, normalizes comma-separated numbers, and requires the target as a word-bounded span in the answer. A sample is correct if either CEM or the semantic-equivalence judge accepts it.

MRCR. Every gold response begins with a sample-specific random prefix. A prediction without the exact prefix receives zero. Otherwise, the scorer removes the prefix and computes Python character-level `SequenceMatcher` similarity between the prediction and reference. The aggregate includes the 2-, 4-, and 8-needle variants across context lengths.

CorpusQA. We use the 0–128K file covering Chinese finance, English finance, English education, and English real estate. The evaluator extracts text following the prescribed answer marker when present, then obtains a binary semantic-equivalence judgment against the generated reference. The reported column is overall example accuracy across the four domains.

LongBench v1 QA. The evaluator extracts the final occurrence of "Therefore, the answer is" and falls back to the last non-empty line when the marker is absent. It compares the extracted answer with every accepted reference and takes the maximum binary semantic judgment. The reported score is the macro mean over NarrativeQA, Qasper, HotpotQA, 2WikiMultihopQA, and MuSiQue.

Student serving accepts at most 120,000 input tokens and uses a 131,072-token model length with YaRN factor 4 and target length 131,072. The completed evaluation runs cap generation at 8,192 tokens. Inputs that exceed the serving budget are middle truncated so that both the beginning and end are retained. Students remain in no-thinking mode, and no method receives a method-specific evaluation prompt or decoding override. Semantic judging uses Qwen3-30B-A3B-Instruct-2507 with a 32,768-token context and a 2,048-token output cap. The reported aggregate is the naive average of the five benchmark scores.

D Auditable Long-Context Case Study

To make the GC-OPD signal computation concrete, we present one example from GoLongRL. Its Qwen3-8B no-thinking prompt contains 17,265 tokens and asks which matrix receives the largest relative improvement from parallel NUMA optimization: A `rajat31`, B `HV15R`, C `cage15`, or D `ldoor`. The dataset's native label is B. The contextual observations reproduced in Figure 5 support this label, but do not by themselves constitute a direct speedup measurement.

This case is an illustrative audit of the signal computation rather than additional quantitative evidence. We independently reroll the example using the official Qwen3-8B base checkpoint and Qwen3-30B-A3B-Thinking-2507 as the teacher. We sample eight responses with temperature 1, top-$p = 1$, top-$k = -1$, seed 42, and a 10,240-token response cap. Table 8 reports representative token signals recomputed offline with $\beta = 0.10$ and $a_{\max} = 10$. In the table, the response index is suppressed within each response block, and $\Delta A_t = \beta c_t \rho$ denotes the additive residual correction before clipping.

Table 8: Representative token-level RACA recalculations for the correct-B and incorrect-D responses in Figure 5 under the final GC-OPD setting. Values are rounded; ranking checks use unrounded values.

Correct B response, $\rho_{\mathrm{B}} = +3.464102$						**Incorrect D response**, $\rho_{\mathrm{D}} = -1.154701$					
Token	A_t	u_t	c_t	ΔA_t	$\widehat{A}_t$	Token	A_t	u_t	c_t	ΔA_t	$\widehat{A}_t$
`[`	-4.88599	-1.85923	0.26959	0.09339	-4.79260	`[`	-4.88599	-1.98073	0.24248	-0.02800	-4.91399
`Answer`	-0.02188	0.83034	1.39285	0.48250	0.46062	`Answer`	-0.02188	0.68817	1.33112	-0.15370	-0.17558
`]`	-0.64095	0.48803	1.23928	0.42930	-0.21165	`]`	-0.64095	0.34849	1.17250	-0.13539	-0.77634
`B`	-1.91057	-0.21400	0.89341	0.30949	-1.60108	`D`	-0.66057	0.33773	1.16728	-0.13479	-0.79535
`EOS`	-0.15839	0.75486	1.36047	0.47128	0.31290	`EOS`	-0.17101	0.60634	1.29421	-0.14944	-0.32046

Case Study: Auditing Long-Context Teacher–Verifier Disagreement

Gold: B (HV15R); trajectory-level OPD ranks D above B

Selected evidence from the 17,265-token context

Excerpt A: Candidate Matrices in Table 1

Matrix	Dimension	NNZ	Avg values per block (summary)
rajat31	4,690,002	20,316,253	1.4, 1.9, 1.9, 2.1, 2.3, 2.2
HV15R	2,017,169	283,073,458	5.4, 5.7, 10, 9.7, 18, 15
cage15	5,154,859	99,199,551	1.2, 2.0, 2.1, 3.1, 3.6, 3.4
ldoor	952,203	46,522,475	7.0, 6.4, 13, 11, 21, 17

Among the four options, HV15R has by far the largest NNZ, making it the clearest large-matrix case in Table 1.

Excerpt B: Parallel SpMV Performance

By comparing Fig. 4 with Table 1 we can observe that the NUMA optimization gives noticeable speedup for large matrices (shown as the dark part of bars in Fig. 4). Indeed, when a matrix is allocated on a single memory socket, any access by threads on a different socket is very expensive.

Excerpt C: NUMA and Cache Effects

This might not be a severe problem if the matrix (or at least the part used by the threads) fits in the L3 cache. Then, only the first access is costly, especially when the data is read-only. On the other hand, if the matrix does not fit in the L3 cache, multiple expensive memory transfers will take place during the computation without any possibility for the CPU to hide them.

Question

For which matrix does the parallel NUMA optimization provide the most significant relative performance improvement?

A. rajat31
B. HV15R
C. cage15
D. ldoor

Gold: B

Diagnostic Analysis on Fixed Student Rollouts (n = 8, β = 0.1)

#	Full rollout text	Choice	R_i	s_i (vanilla OPD)	$\rho_i = z(R_i) - z(s_i)$	Final mean A_i'
1	[Answer] D <EOS>	D	0	−1.276079	−1.154701	−1.396343
2	[Answer] D <EOS>	D	0	−1.276079	−1.154701	−1.396343
3	[Answer] D <EOS>	D	0	−1.276079	−1.154701	−1.396343
4	[Answer] B <EOS>	B	1	−1.523554	+3.464102	−1.166363
5	[Answer] D <EOS>	D	0	−1.276079	−1.154701	−1.396343
6	[Answer] D <EOS>	D	0	−1.276079	−1.154701	−1.396343
7	[Answer] D <EOS>	D	0	−1.276079	−1.154701	−1.396343
8	[Answer] B <EOS>	B	1	−1.523554	+3.464102	−1.166363

Figure 5: Illustrative fixed-rollout visualization for the 17,265-token case. It reproduces the selected context evidence and all eight independently rerolled student responses; the native label is B.