COMPARE TO CORRECT: PAIRWISE-GUIDED LOGIT ADJUSTMENT FOR LLM TEST-TIME SCALING

Anonymous authors
Paper under double-blind review

ABSTRACT

Sampling-based test-time scaling enhances LLM reasoning by aggregating multiple reasoning paths for the same query. Existing methods use answer frequencies or path confidence to compute candidate scores, which do not explicitly characterize comparisons between competing answers. Inspired by the Condorcet principle in social choice theory, we introduce pairwise confidence comparisons between paths supporting different answers to capture answer-level comparative advantage. We combine this comparative evidence with voting support through a KL-regularized objective. Theoretically, we show that this comparative-advantage is equivalent to the adjustment to the traditional voting logits. In this way, our proposed *Pairwise-Guided Logit Adjustment* (PGLA) framework allows comparative evidence to compensate for weaker voting support and correct voting-based ranking errors. Experiments on 10 challenging benchmarks including mathematical reasoning, knowledge reasoning, and code generation with 7 models ranging from **1.7B** to **32B** parameters speak to the effectiveness of this plug-and-play term. Additional evaluations with external reward scores and online sampling further demonstrate its applicability.

1 INTRODUCTION

Large language models (LLMs) have demonstrated strong reasoning capabilities across mathematical, symbolic, and multi-step decision-making tasks (Wei et al., 2022; Lewkowycz et al., 2022; Yao et al., 2023). Beyond increasing model size, allocating additional computation at inference time can further improve reasoning performance, establishing *test-time scaling* as an important paradigm for LLM reasoning (Snell et al., 2025; Balachandran et al., 2025). Among its variants, *sampling-based* approaches generate multiple reasoning paths for the same query and aggregate evidence from these paths (e.g., answer frequencies and path confidence scores) into a collective decision (Cobbe et al., 2021; Wang et al., 2023). In this direction, the central challenge becomes **how to aggregate path-level evidence into reliable answer-level scores.** Representative approaches include *best-of-N selection* that chooses the answer from the path with the highest confidence score (Cobbe et al., 2021), *majority voting* based on answer frequencies (Wang et al., 2023), and *confidence-based voting* that combines answer frequencies with path confidence scores by weighting or filtering path contributions (Kang et al., 2025; Fu et al., 2026).

Although these methods achieve substantial improvements, we observe that their aggregation rules could overlook some crucial information hidden among sampled paths. To illustrate this, we turn to *social choice theory*, which studies how individual preferences among available options can be combined into a collective decision (Zwicker, 2016). The well-known *Condorcet principle* highlights that the most frequently selected option does not necessarily rank higher than competing options in direct comparisons (Young, 1988; Fishburn, 1977). As shown in Figure 1(a), options A, B, and C receive 3, 2, and 2 first-choice selections, respectively, so majority voting selects A. Under the preference P, however, four of seven rankings place B above A, and five place B above C, so B is selected through pairwise comparisons. Similarly, this principle highlights why prior sampling-based approaches are insufficient: an answer can be assigned the highest score even when its reasoning paths remain at a disadvantage in comparisons with those supporting a competing answer. As shown in Figure 1(b), answer A is selected by majority voting because it appears most frequently, although every path producing answer B has a higher confidence score than those producing A. This

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.

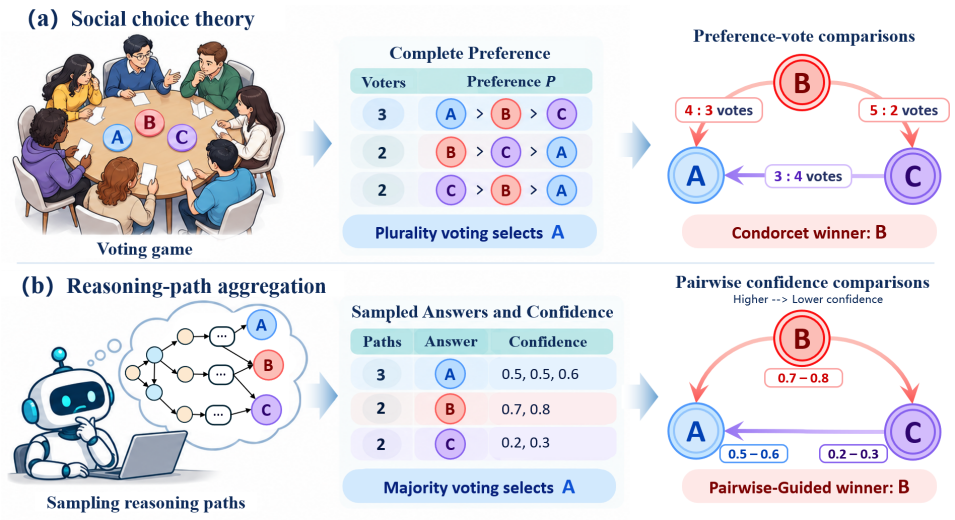


Figure 1: From voting support to pairwise comparative evidence. (a) Plurality voting selects A, whereas pairwise preference comparisons identify B as the Condorcet winner. (b) Sampled reasoning paths provide answers and confidence scores. Pairwise confidence comparisons complement voting support, enabling the proposed aggregation method to select B.

motivates us to **incorporate pairwise information between candidate answers on top of existing voting-based aggregation.**

Ideally, we could apply the Condorcet principle to the reasoning paths if the ranking of candidate answers were available for each path. However, in practice, we only observe each path's first-choice answer and its confidence score, leaving the ordering among the remaining answers unknown. To estimate such missing ordering, we use ***pairwise confidence comparisons*** between paths producing different answers, as illustrated in Figure 1(b). These comparisons yield an answer-level comparative advantage, which we combine with voting evidence through a KL-regularized objective. In this way, the objective can balance this advantage against deviations from the base voting distribution, as an alternative to the naive Condorcet approach. Furthermore, we show that its solution is essentially a *logit adjustment* to the base voting distribution, yielding a simple yet effective *Pairwise-Guided Logit Adjustment* (PGLA) approach. This plug-and-play adjustment allows comparative evidence to compensate for weaker evidence from existing voting-based aggregation.

Finally, we conduct extensive experiments on mathematical reasoning, knowledge reasoning, and code generation using both reasoning and non-reasoning models from the **Qwen**, **DeepSeek**, and **InternLM** families, with model sizes ranging from **1.7B** to **32B** parameters. With internal confidence scores, PGLA achieves an average accuracy gain of approximately 2% over the corresponding voting baselines across 33 model–benchmark settings. Additional experiments show gains with external reward scores. On six mathematical reasoning settings, its online extension also achieves higher mean accuracy with fewer sampled paths than Online SC under the same stopping rule.

2 Problem Formulation and Representative Aggregators

2.1 Problem Formulation

Given an input query, a language model generates K reasoning paths $z_1, \ldots, z_K$. Each path produces an answer $a_i = g(z_i)$ and receives a confidence score p_i from the model or an external verifier. Together, they form the path-level evidence pool $\mathcal{D} = \{(a_i, p_i)\}_{i=1}^{K}$. Let $\mathcal{A}$ be the set of different answers produced by these paths, which may be predefined for multiple-choice questions or dynamically formed from sampled answers in open-ended mathematical reasoning and code generation.

Based on $\mathcal{D}$, we construct a *path-answer matrix* $\mathcal{P}$, with one row per path and one column per candidate answer. Each row records p_i in the column for its output answer a_i and the entries for other

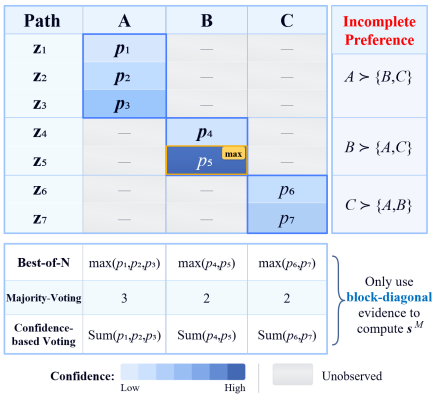


Figure 2: Path-answer evidence and its aggregation. The three methods use each block's maximum confidence, size, or confidence-based contributions.

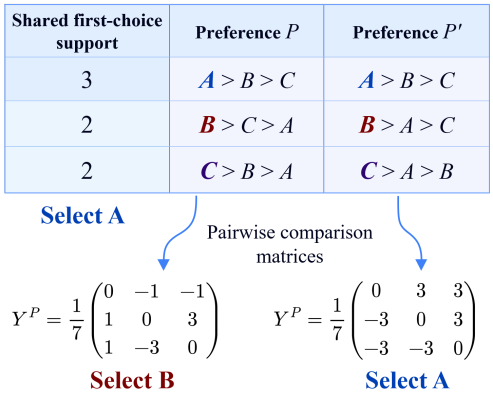


Figure 3: Identical majority-voting scores can correspond to different pairwise comparison matrices and different winners.

answers remain unobserved. Grouping paths by their output answers produces the block-diagonal pattern shown in Fig. 2. Each block contains the confidence scores of paths sharing the same answer.

The aggregation task is to **convert the path-level evidence into an answer-level score for each candidate, followed by selecting the candidate with the highest score**. Formally, an aggregation method $\mathcal{M}$ assigns candidate a the score $s_a^{\mathcal{M}}$ based on $\mathcal{D}$:

$$s_a^{\mathcal{M}} = \mathcal{M}(a; \mathcal{D}), \qquad \hat{a} \in \arg\max_{a \in \mathcal{A}} s_a^{\mathcal{M}}. \tag{1}$$

$\hat{a}$ is the final predicted answer selected from the candidates with the highest score. Representative methods use different information from the path-answer matrix to compute $s_a^{\mathcal{M}}$.

Best-of-N selection. Best-of-N uses the highest confidence among the paths sharing answer a (Cobbe et al., 2021). This corresponds to taking the largest value within the block for a in matrix $\mathcal{P}$:

$$s_a^{\text{BoN}} = \max_{i:a_i=a} p_i. \tag{2}$$

The confidence p_i may be perplexity (Adiwardana et al., 2020), self-certainty (Kang et al., 2025), DeepConf confidence (Fu et al., 2026), or an external verifier's reward (Lightman et al., 2024).

Majority voting. Self-consistency (SC) treats each path as one vote for its output answer and selects the most frequently generated answer (Wang et al., 2023). The answer-level score is therefore its total vote count:

$$s_a^{\text{SC}} = \sum_{i=1}^{K} \mathbf{1}(a_i = a). \tag{3}$$

where $\mathbf{1}(\cdot)$ is the indicator function. In $\mathcal{P}$, this corresponds to the size of the diagonal block for answer a, measured by its number of observed entries. Every supporting path contributes equally, regardless of its confidence.

Confidence-based voting. Confidence-based voting methods adjust path contributions through weighting or filtering (Kang et al., 2025; Fu et al., 2026). Let $h_i \geq 0$ be the weighting or filtering coefficient for path i, with $h_i = 0$ for filtered paths. The resulting answer-level score is

$$s_a^{\text{Conf-Vote}} = \sum_{i=1}^{K} h_i \cdot p_i \, \mathbf{1}(a_i = a). \tag{4}$$

This method aggregates confidence-based contributions within each diagonal block in $\mathcal{P}$.

Despite these differences, **these methods aggregate only the observed evidence in the diagonal blocks of** $\mathcal{P}$. They do not explicitly model the unobserved preferences among alternative answers. In the next section, we use social choice theory to explain why these preferences matter for answer aggregation.

2.2 Motivation: A Social Choice Perspective

Social choice theory studies how individual preferences over available options are aggregated into a collective decision (Zwicker, 2016). To illustrate this, Fig. 3 shows two complete preference profiles, P and P', with the same majority-voting support. To be specific, A, B, and C receive 3, 2, and 2 first-choice selections, respectively. Majority voting therefore selects A in both. However, the remaining options are ranked differently in P and P', giving different pairwise comparison matrices Y. Each entry Y^{P}_{uv} measures the voting advantage of u over v in a direct comparison under P. A positive value means that more rankings place u above v. Then, Y^{P} can be aggregated into *answer-level comparative advantages* $s^{\mathcal{M},\mathrm{P}}_u$ using different rules $\mathcal{M}$. The highest-scoring answer is selected as in Eq. (1). For example, selection under the Condorcet principle (Young, 1988; Fishburn, 1977) can be represented by

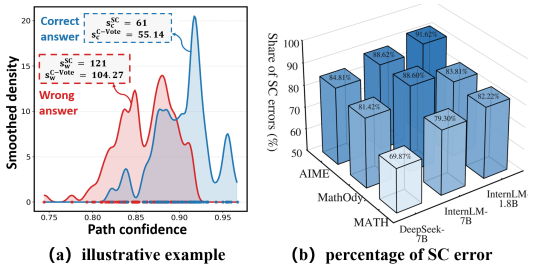
$$s^{\mathrm{Condorcet},\mathrm{P}}_u = \mathbf{1}\left\{Y^{\mathrm{P}}_{uv} > 0 \text{ for every } v \in \mathcal{A} \setminus \{u\}\right\}. \tag{5}$$

This selects the answer that wins every direct comparison, whenever one exists. Otherwise, all scores are zero, and an additional selection rule is needed. Following this rule, we select B under P in Fig. 3 because it defeats A by 4:3 and C by 5:2. Under P', A defeats both B and C by 5:2 and is selected. Thus, we arrive at the following proposition.

Proposition 2.1. *There exist two complete preference profiles P and P' such that:*

(i) Under P, the most frequently selected option is still less preferred than another option in direct pairwise comparisons.
(ii) P has the same majority-voting scores as P', but different pairwise comparison matrices and different winners.

Connection to the Sampling-based LLM reasoning. In sampling-based LLM reasoning, we only observe each path's first-choice answer a_i and its confidence score p_i, leaving the complete preference among the remaining answers unknown. To approximate the missing ordering, we construct a proxy pairwise matrix Y by comparing path confidence scores across answer groups. A similar disagreement highlighted in Prop. 2.1 also occurs in this setting: an answer with the highest aggregation score may still lose to another answer in pairwise comparisons. To examine this phenomenon empirically, we consider SC errors with the following disagreement. An incorrect answer w receives at least as many votes as a correct answer c, i.e., $s^{\mathrm{SC}}_w \geq s^{\mathrm{SC}}_c$, yet paths supporting c tend to have higher confidence scores than those supporting w in pairwise comparisons. The formal definition is provided in App. C. Using PPL confidence, **69.87%–91.62% of recoverable SC errors** (i.e., candidate pools containing at least one correct answer) satisfy this condition across three models and three mathematical benchmarks as shown in Fig. 4(b).



(a) illustrative example (b) percentage of SC error

Figure 4: Disagreement between voting and pairwise confidence comparisons: (a) an illustrative example; (b) percentage of recoverable SC errors.

These observations motivate **enhancing existing voting-based aggregation by incorporating explicit comparisons between answer**. We will discuss our solution in the next section.

3 Pairwise-Guided Path Aggregation

In this section, we propose *Pairwise-Guided Logit Adjustment* (PGLA) for Path Aggregation. This method aims to construct answer-level comparative evidence through pairwise confidence comparisons and combine it with the voting distribution, as an alternative to the Condorcet approach.

3.1 Pairwise Confidence Comparisons

Sampled reasoning paths provide answers and confidence scores rather than complete preference rankings. We therefore use confidence comparisons between paths supporting different answers to

construct a proxy for answer-level preference comparisons. For each answer $u \in \mathcal{A}$, let $C_u = \{i : a_i = u\}$ denote the indices of its supporting paths. For two different answers $u, v \in \mathcal{A}$, we define the corresponding entry of the confidence-based comparison matrix Y as

$$Y_{uv} = \frac{1}{|C_u||C_v|} \sum_{i \in C_u} \sum_{j \in C_v} \operatorname{sgn}(p_i - p_j), \qquad u \neq v, \tag{6}$$

Each comparison contributes $+1$, 0, or -1 when the path supporting u has a higher, equal, or lower confidence score. A positive Y_{uv} means that paths supporting u have higher confidence more often than paths supporting v in these comparisons than the reverse.

We next summarize Y as an answer-level *comparative advantage*. For a voting aggregator $\mathcal{V}$, $s_u^{\mathcal{V}} \geq 0$ is the base voting support for the answer u defined in Eqs. (3) and (4). We give greater weight to comparisons against answers with stronger voting support. Accordingly, the comparative advantage of answer u is

$$s_u^{\text{Pair}} = \sum_{v \neq u} \pi_{v|u} Y_{uv}, \qquad \pi_{v|u} = \frac{s_v^{\mathcal{V}}}{\sum_{k \neq u} s_k^{\mathcal{V}}}, \qquad v \neq u. \tag{7}$$

$s_u^{\text{Pair}} \in [-1, 1]$ is a weighted average of comparison scores. Alternative rules include uniform averaging of pairwise scores; Copeland, which counts pairwise wins minus losses (Zwicker, 2016); and Maximin, which takes the minimum pairwise score (Fischer et al., 2016). We use Eq. (7) by default. Definitions and matched comparisons of the alternatives appear in App. G.11.

Regardless of the aggregation rule, s_u^{Pair} summarizes confidence comparisons rather than the preference votes used by the Condorcet principle in Eq. (5). Since this score does not directly consider answer u's own voting support, we therefore explore to combine this comparative advantage with the voting information provided by any voting-based aggregation strategy $\mathcal{V}$.

3.2 Combining with Voting Information

Our goal is to refine the answer-level scores $s_u^{\mathcal{V}}$ with two complementary objectives: **(G1)** favoring answers with stronger comparative advantages and **(G2)** exploiting the voting support provided by the aggregator $\mathcal{V}$. We formulate this refinement over answer distributions, using the normalized base scores to obtain the reference distribution $\bar{\mathbf{s}}^{\mathcal{V}}$, with $\bar{s}_u^{\mathcal{V}} = s_u^{\mathcal{V}} / \sum_{v \in \mathcal{A}} s_v^{\mathcal{V}}$. We then seek an adjusted distribution through the following regularized objective:

$$\bar{\mathbf{s}}^{\mathsf{PGLA}} \in \arg\max_{\bar{\mathbf{s}}} \left[\underbrace{\sum_{u \in \mathcal{A}} \bar{s}_u s_u^{\text{Pair}}}_{\text{(G1)}} - \underbrace{D_{\text{KL}}\big(\bar{\mathbf{s}} \,\|\, \bar{\mathbf{s}}^{\mathcal{V}}\big)}_{\text{(G2)}} \right]. \tag{8}$$

Here, $\bar{\mathbf{s}}$ is the optimization variable, with $\bar{s}_u \geq 0$ and $\sum_{u \in \mathcal{A}} \bar{s}_u = 1$; $\bar{\mathbf{s}}^{\mathcal{V}}$ and s_u^{Pair} are fixed by the observed candidate pool.

The first term **rewards favorable answer comparisons**, allowing pairwise confidence evidence to guide answer selection **(G1)**. The KL term **penalizes departures from the voting distribution**, using the original voting support to constrain this adjustment **(G2)**. The resulting distribution balances comparative evidence with voting support. Intuitively, reversing a strong voting preference requires a larger difference in comparative scores, whereas pairwise evidence can more readily influence selection when answers have similar voting support. This formulation extends the **pairwise-comparison perspective of the Condorcet principle** to confidence-based path aggregation.

At first glance, Eq. (8) makes the property of the solution unclear. Fortunately, the following proposition can alleviate this issue, with proof in App. E.

Proposition 3.1. *If $\bar{s}_u^{\mathcal{V}} > 0$ for every $u \in \mathcal{A}$, the objective in Eq. (8) is strictly concave and has the unique solution*

$$\bar{s}_u^{\mathsf{PGLA}} = \frac{\bar{s}_u^{\mathcal{V}} \exp(s_u^{\text{Pair}})}{\sum_{v \in \mathcal{A}} \bar{s}_v^{\mathcal{V}} \exp(s_v^{\text{Pair}})}. \tag{9}$$

Since $\bar{s}_u^{\mathcal{V}} \propto s_u^{\mathcal{V}}$, the optimal distribution in Eq. (9) can be obtained by normalizing the score $s_u^{\mathsf{PGLA}} = s_u^{\mathcal{V}} \exp(s_u^{\mathrm{Pair}})$. Taking logarithms gives the equivalent formulation in the logit space:

$$\log\big(s_u^{\mathsf{PGLA}}\big) = \log\big(s_u^{\mathcal{V}}\big) + s_u^{\mathrm{Pair}}. \tag{10}$$

The highest-scoring answer is selected as in Eq. (1).

> **Remark 3.1** (Test-time logit adjustment)**.** Thus, we find that the proposed PGLA can be viewed as a form of ***Pairwise-guided logit adjustment***, operating on answer-logit in the observed sampling pool. This additive form connects PGLA to the classical strategy of Logit Adjustment (LA) (Menon et al., 2021). In long-tailed classification, post-hoc LA uses offsets based on class priors to reduce the preference for frequent classes. Here, **this adjustment allows pairwise confidence comparison evidence to compensate for weaker voting support**. What's more, as confidence scores better distinguish correct from incorrect paths, PGLA can make more reliable pairwise comparisons and further improve aggregation performance. In Sec. 4.4, we test how the reliability of these confidence scores affects PGLA accuracy.

4 Experiments

In this section, we conduct extensive experiments to evaluate the effectiveness of PGLA. More experimental details and results are provided in App. F and App. G.

4.1 Experimental Setup

Competitors. We compare against **11** representative sampling-based aggregation baselines using the same pool of sampled reasoning paths. The max-based baselines follow the Best-of-N paradigm, using path **perplexity confidence (PPL)** (Chen et al., 1998), **DeepConf** (Fu et al., 2026), and **Self-Certainty** (Kang et al., 2025) to score each path. The voting-based baselines include **self-consistency (SC)**, three **confidence-weighted SC** variants based on the above scores, the **Self-Certainty Borda rule** (Kang et al., 2025), **DeepConf's offline top-**90% **filtered weighted voting** (Fu et al., 2026), **adaptive self-consistency** (Aggarwal et al., 2023), and **RPC** (Zhou et al., 2025). Beyond internal confidence scores, we evaluate three baselines based on scores from an external process reward model (PRM): **PRM Best-of-**N, **PRM-Weighted SC**, and **PRM-RPC**. We use Qwen2.5-Math-PRM-7B as the external scorer (Lightman et al., 2024; Zhang et al., 2025). PRM-Weighted SC assigns each path a voting weight equal to the product of its step-level PRM scores and sums these weights within each answer group. **We also apply the same PGLA adjustment to all voting-based aggregators**.

LLM Backbones. To test generality across model families and scales, we run our pipeline with **Qwen3-1.7B**, **Qwen3-4B**, **Qwen3-8B**, and **Qwen3-32B** (Qwen, 2025), all in reasoning mode. We also include three open math-specialized LLM backbones: **DeepSeek-Math-RL-7B** (Shao et al., 2024), **InternLM2-Math-Plus-1.8B**, and **InternLM2-Math-Plus-7B** (Ying et al., 2024).

Datasets. We evaluate on ten challenging datasets, **AIME 2024** (AI-MO, 2024), **AIME 2025** (OpenCompass, 2025), **HMMT 2025** (Balunović et al., 2025), **AIME** (Zamil & Rabby, 2024), **MATH** (Hendrycks et al., 2021), **MathOdyssey** (Fang et al., 2025), and **GPQA-Diamond (GPQA-D)**, together with the code benchmarks **HumanEval** (Chen et al., 2021), **MBPP** (Austin et al., 2021), and **LiveCodeBench V6** (Jain et al., 2025). The first four are high-difficulty mathematical competition problems, while GPQA comprises graduate-level STEM reasoning tasks. Most of benchmarks are widely adopted in recent evaluations of top reasoning LLMs. Additionally, we also report **HumanEval+** and **MBPP+** using the strengthened EvalPlus tests (Liu et al., 2023).

4.2 Overall Performance

Performance using Advanced Models. Tab. 1 compares the overall performance of different aggregation methods on Qwen3 reasoning models at three scales. We observe that our method matches or improves upon the corresponding voting baselines across the evaluated reasoning and code-generation tasks. In particular, averaged over the six corresponding voting baselines on Qwen3-8B, PGLA improves accuracy by 4.61%, 1.78%, and 2.63% on AIME24, AIME25, and HMMT25, respectively. The gains also extend to the larger Qwen3-32B model on mathematical reasoning tasks shown in Tab. 22. These consistent gains demonstrate that PGLA generalizes well across model scales and task domains.

Table 1: Performance of reasoning models at different scales on three mathematical reasoning benchmarks, GPQA, and three open-ended code-generation benchmarks, using $K = 32$ sampled responses per problem. Each voting-based baseline is followed by its PGLA-enhanced variant, denoted by **+Ours**. The enhanced HumanEval+ and MBPP+ tests for the same sampled pools are reported in Tab. 6.

Method	AIME24	AIME25	HMMT25	GPQA-D	HE	MBPP	LCB
Qwen3-1.7B Reasoning mode							
PPL Best-of-N	$36.00_{\pm 4.42}$	$34.00_{\pm 2.91}$	$23.00_{\pm 3.48}$	$39.90_{\pm 2.02}$	$81.10_{\pm 2.06}$	$75.13_{\pm 0.68}$	$30.67_{\pm 1.47}$
DeepConf Best-of-N	$62.00_{\pm 4.52}$	$42.67_{\pm 2.49}$	$28.33_{\pm 3.73}$	$42.93_{\pm 1.52}$	$84.15_{\pm 2.69}$	$76.72_{\pm 1.18}$	$34.29_{\pm 1.62}$
Self-Certainty Best-of-N	$60.67_{\pm 4.16}$	$46.67_{\pm 3.33}$	$30.67_{\pm 4.42}$	$42.17_{\pm 0.25}$	$85.06_{\pm 2.70}$	$81.75_{\pm 1.04}$	$33.33_{\pm 1.64}$
SC	$69.50_{\pm 1.98}$	$47.33_{\pm 3.67}$	$27.00_{\pm 1.00}$	$44.44_{\pm 0.51}$	$83.84_{\pm 1.32}$	$83.07_{\pm 0.89}$	$29.90_{\pm 1.33}$
+ Ours (DeepConf)	$\mathbf{75.00}_{\pm 1.67}$	$\mathbf{51.33}_{\pm 2.67}$	$\mathbf{30.00}_{\pm 2.98}$	$\mathbf{46.72}_{\pm 0.25}$	$\mathbf{88.11}_{\pm 2.24}$	$\mathbf{85.45}_{\pm 0.80}$	$\mathbf{34.29}_{\pm 1.36}$
+ Ours (Self-Certainty)	$74.00_{\pm 1.33}$	$50.67_{\pm 3.27}$	$\mathbf{30.33}_{\pm 2.77}$	$44.95_{\pm 0.51}$	$87.80_{\pm 1.61}$	$\mathbf{85.45}_{\pm 0.80}$	$\mathbf{33.33}_{\pm 1.35}$
DeepConf-Weighted SC	$70.00_{\pm 1.49}$	$48.33_{\pm 4.28}$	$27.00_{\pm 1.00}$	$44.19_{\pm 0.76}$	$85.06_{\pm 1.11}$	$84.66_{\pm 0.67}$	$33.71_{\pm 1.30}$
+ Ours	$\mathbf{75.00}_{\pm 1.67}$	$\mathbf{51.00}_{\pm 3.00}$	$\mathbf{31.33}_{\pm 3.40}$	$\mathbf{46.21}_{\pm 0.76}$	$\mathbf{87.50}_{\pm 2.22}$	$\mathbf{85.45}_{\pm 0.80}$	$\mathbf{34.67}_{\pm 1.63}$
Self-Certainty-Weighted SC	$70.33_{\pm 1.80}$	$48.67_{\pm 4.27}$	$27.00_{\pm 1.00}$	$44.19_{\pm 0.76}$	$83.54_{\pm 1.07}$	$84.66_{\pm 0.57}$	$32.57_{\pm 1.11}$
+ Ours	$\mathbf{74.00}_{\pm 1.33}$	$\mathbf{50.67}_{\pm 3.27}$	$\mathbf{31.00}_{\pm 3.35}$	$\mathbf{44.70}_{\pm 0.76}$	$\mathbf{88.11}_{\pm 1.94}$	$\mathbf{85.71}_{\pm 0.86}$	$\mathbf{33.71}_{\pm 1.31}$
Self-Certainty Borda	$73.33_{\pm 1.49}$	$50.33_{\pm 3.48}$	$29.33_{\pm 2.91}$	$43.69_{\pm 2.27}$	$86.59_{\pm 1.54}$	$85.19_{\pm 0.86}$	$32.95_{\pm 1.28}$
+ Ours	$\mathbf{74.67}_{\pm 1.63}$	$\mathbf{50.67}_{\pm 3.27}$	$\mathbf{34.33}_{\pm 3.96}$	$\mathbf{44.95}_{\pm 2.02}$	$\mathbf{88.11}_{\pm 1.83}$	$\mathbf{85.45}_{\pm 0.77}$	$\mathbf{33.52}_{\pm 1.43}$
DeepConf-Filter (90%)	$70.33_{\pm 1.80}$	$50.67_{\pm 4.42}$	$27.00_{\pm 1.80}$	$44.44_{\pm 1.01}$	$86.28_{\pm 1.39}$	$85.19_{\pm 0.71}$	$33.90_{\pm 1.38}$
+ Ours	$\mathbf{75.00}_{\pm 1.67}$	$\mathbf{52.33}_{\pm 3.67}$	$\mathbf{33.67}_{\pm 2.33}$	$\mathbf{47.22}_{\pm 0.25}$	$\mathbf{87.50}_{\pm 2.32}$	$\mathbf{85.71}_{\pm 0.85}$	$\mathbf{34.48}_{\pm 1.63}$
Qwen3-4B Reasoning mode							
PPL Best-of-N	$57.00_{\pm 2.33}$	$64.33_{\pm 3.96}$	$40.33_{\pm 4.33}$	$54.71_{\pm 1.41}$	$92.38_{\pm 0.77}$	$91.85_{\pm 0.75}$	$44.00_{\pm 1.26}$
DeepConf Best-of-N	$77.00_{\pm 2.33}$	$76.00_{\pm 3.89}$	$50.00_{\pm 4.94}$	$58.42_{\pm 2.00}$	$96.19_{\pm 0.81}$	$93.44_{\pm 0.28}$	$47.43_{\pm 0.85}$
Self-Certainty Best-of-N	$81.67_{\pm 2.24}$	$71.67_{\pm 3.07}$	$51.67_{\pm 4.01}$	$59.93_{\pm 1.12}$	$96.04_{\pm 0.82}$	$93.70_{\pm 0.41}$	$47.81_{\pm 1.16}$
SC	$80.00_{\pm 0.00}$	$\mathbf{76.33}_{\pm 0.67}$	$51.50_{\pm 1.57}$	$58.25_{\pm 0.45}$	$93.60_{\pm 1.00}$	$93.92_{\pm 0.58}$	$41.71_{\pm 1.26}$
+ Ours (DeepConf)	$\mathbf{83.00}_{\pm 1.00}$	$\mathbf{76.33}_{\pm 1.80}$	$54.00_{\pm 2.91}$	$\mathbf{60.27}_{\pm 0.88}$	$\mathbf{95.88}_{\pm 0.95}$	$94.29_{\pm 0.41}$	$\mathbf{47.62}_{\pm 0.85}$
+ Ours (Self-Certainty)	$\mathbf{82.00}_{\pm 1.63}$	$\mathbf{78.67}_{\pm 1.63}$	$\mathbf{55.67}_{\pm 1.53}$	$\mathbf{61.28}_{\pm 0.85}$	$95.73_{\pm 0.55}$	$\mathbf{94.31}_{\pm 0.49}$	$\mathbf{48.00}_{\pm 1.12}$
DeepConf-Weighted SC	$80.00_{\pm 0.00}$	$\mathbf{76.33}_{\pm 1.00}$	$51.33_{\pm 1.63}$	$58.59_{\pm 0.53}$	$94.36_{\pm 1.01}$	$94.21_{\pm 0.48}$	$47.24_{\pm 0.95}$
+ Ours	$\mathbf{83.33}_{\pm 1.49}$	$\mathbf{76.33}_{\pm 2.33}$	$\mathbf{54.00}_{\pm 2.91}$	$\mathbf{60.10}_{\pm 0.79}$	$\mathbf{95.88}_{\pm 0.87}$	$\mathbf{94.26}_{\pm 0.44}$	$\mathbf{47.62}_{\pm 0.88}$
Self-Certainty-Weighted SC	$80.00_{\pm 0.00}$	$76.33_{\pm 1.00}$	$52.67_{\pm 2.00}$	$58.42_{\pm 0.38}$	$94.21_{\pm 0.86}$	$\mathbf{94.31}_{\pm 0.49}$	$47.81_{\pm 1.18}$
+ Ours	$\mathbf{82.00}_{\pm 1.63}$	$\mathbf{79.33}_{\pm 1.33}$	$\mathbf{55.33}_{\pm 1.63}$	$\mathbf{61.45}_{\pm 0.94}$	$\mathbf{95.73}_{\pm 0.58}$	$\mathbf{94.31}_{\pm 0.49}$	$\mathbf{48.19}_{\pm 1.15}$
Self-Certainty Borda	$81.00_{\pm 1.53}$	$\mathbf{79.33}_{\pm 1.33}$	$55.00_{\pm 1.67}$	$60.10_{\pm 0.60}$	$94.97_{\pm 0.90}$	$94.18_{\pm 0.49}$	$47.81_{\pm 1.18}$
+ Ours	$\mathbf{82.33}_{\pm 2.13}$	$\mathbf{79.33}_{\pm 1.33}$	$\mathbf{55.33}_{\pm 1.63}$	$\mathbf{60.94}_{\pm 0.75}$	$\mathbf{95.73}_{\pm 0.58}$	$\mathbf{94.26}_{\pm 0.31}$	$\mathbf{48.00}_{\pm 1.14}$
DeepConf-Filter (90%)	$80.00_{\pm 0.00}$	$78.67_{\pm 2.21}$	$52.67_{\pm 3.27}$	$58.59_{\pm 0.72}$	$94.66_{\pm 0.81}$	$94.23_{\pm 0.39}$	$47.24_{\pm 0.95}$
+ Ours	$\mathbf{83.33}_{\pm 2.11}$	$\mathbf{79.00}_{\pm 1.53}$	$\mathbf{55.67}_{\pm 2.60}$	$\mathbf{61.11}_{\pm 1.10}$	$\mathbf{96.04}_{\pm 0.71}$	$\mathbf{94.29}_{\pm 0.36}$	$\mathbf{47.43}_{\pm 0.81}$
Qwen3-8B Reasoning mode							
PPL Best-of-N	$61.00_{\pm 3.35}$	$59.33_{\pm 2.00}$	$35.33_{\pm 3.06}$	$59.43_{\pm 0.63}$	$93.70_{\pm 1.24}$	$93.47_{\pm 0.40}$	$46.29_{\pm 1.87}$
DeepConf Best-of-N	$79.33_{\pm 4.16}$	$76.00_{\pm 2.91}$	$53.67_{\pm 3.14}$	$63.64_{\pm 0.41}$	$96.14_{\pm 0.51}$	$94.44_{\pm 0.44}$	$49.71_{\pm 0.00}$
Self-Certainty Best-of-N	$83.67_{\pm 1.80}$	$76.33_{\pm 2.77}$	$52.67_{\pm 2.00}$	$63.47_{\pm 0.63}$	$96.54_{\pm 0.66}$	$95.24_{\pm 0.33}$	$52.38_{\pm 1.08}$
SC	$81.50_{\pm 1.57}$	$75.67_{\pm 2.91}$	$55.94_{\pm 1.17}$	$63.22_{\pm 0.95}$	$93.90_{\pm 0.60}$	$94.71_{\pm 0.39}$	$44.57_{\pm 1.68}$
+ Ours (DeepConf)	$\mathbf{86.67}_{\pm 2.11}$	$\mathbf{79.67}_{\pm 1.80}$	$58.67_{\pm 1.63}$	$\mathbf{64.98}_{\pm 0.24}$	$96.14_{\pm 0.86}$	$95.68_{\pm 0.39}$	$49.90_{\pm 0.27}$
+ Ours (Self-Certainty)	$86.00_{\pm 2.00}$	$76.67_{\pm 1.49}$	$\mathbf{60.00}_{\pm 2.11}$	$64.31_{\pm 0.86}$	$\mathbf{96.54}_{\pm 1.14}$	$\mathbf{96.12}_{\pm 0.52}$	$\mathbf{52.38}_{\pm 1.08}$
DeepConf-Weighted SC	$81.67_{\pm 1.67}$	$76.00_{\pm 3.27}$	$56.67_{\pm 1.49}$	$63.30_{\pm 1.19}$	$94.11_{\pm 0.62}$	$94.97_{\pm 0.46}$	$49.14_{\pm 0.47}$
+ Ours	$\mathbf{87.00}_{\pm 2.33}$	$\mathbf{79.67}_{\pm 1.80}$	$\mathbf{59.00}_{\pm 1.53}$	$\mathbf{64.81}_{\pm 0.24}$	$\mathbf{96.14}_{\pm 0.72}$	$\mathbf{95.68}_{\pm 0.37}$	$\mathbf{49.90}_{\pm 0.27}$
Self-Certainty-Weighted SC	$81.67_{\pm 1.67}$	$76.00_{\pm 3.27}$	$56.67_{\pm 1.49}$	$63.30_{\pm 1.19}$	$94.31_{\pm 0.62}$	$95.15_{\pm 0.41}$	$51.81_{\pm 0.71}$
+ Ours	$\mathbf{86.33}_{\pm 1.80}$	$\mathbf{76.67}_{\pm 1.49}$	$\mathbf{60.00}_{\pm 1.49}$	$\mathbf{64.31}_{\pm 0.86}$	$\mathbf{96.54}_{\pm 1.01}$	$\mathbf{96.21}_{\pm 0.54}$	$\mathbf{52.38}_{\pm 1.08}$
Self-Certainty Borda	$83.67_{\pm 1.80}$	$\mathbf{76.67}_{\pm 1.49}$	$59.33_{\pm 2.49}$	$63.97_{\pm 0.95}$	$95.33_{\pm 0.49}$	$95.59_{\pm 0.33}$	$52.00_{\pm 0.47}$
+ Ours	$\mathbf{86.33}_{\pm 1.80}$	$\mathbf{76.67}_{\pm 1.49}$	$\mathbf{60.00}_{\pm 2.11}$	$\mathbf{64.31}_{\pm 0.86}$	$\mathbf{96.34}_{\pm 0.91}$	$\mathbf{96.12}_{\pm 0.52}$	$\mathbf{52.38}_{\pm 1.08}$
DeepConf-Filter (90%)	$81.67_{\pm 1.67}$	$78.00_{\pm 2.67}$	$59.00_{\pm 1.53}$	$63.30_{\pm 0.95}$	$94.31_{\pm 0.49}$	$94.97_{\pm 0.40}$	$49.14_{\pm 0.47}$
+ Ours	$\mathbf{87.00}_{\pm 2.33}$	$\mathbf{79.33}_{\pm 1.33}$	$\mathbf{61.67}_{\pm 2.24}$	$\mathbf{64.31}_{\pm 0.63}$	$\mathbf{96.14}_{\pm 0.51}$	$\mathbf{95.68}_{\pm 0.39}$	$\mathbf{49.90}_{\pm 0.27}$

Performance with the Reliability of Confidence Estimation. Tab. 2 compares the aggregation methods under internal confidence estimates from three mathematics-specialized backbones. We observe that our method consistently improves accuracy while reducing ECE across the four voting baselines, achieving an average accuracy improvement of 1.96% and an average ECE reduction of 1.42%. This demonstrates the robustness of our method to confidence estimates of varying reliability.

Performance with the External Reward Setting. Tab. 3 and Tab. 9 compares the methods using product-aggregated rewards from an external process reward model. We observe that our method improves PRM-Weighted SC and PRM-RPC in most settings. To be specific, at the full budget, PGLA raises their average accuracy from 33.98 to 35.51 and from 34.30 to 35.55, respectively. This further demonstrates that our method can effectively use externally provided reward signals.

4.3 Case Analysis

We examine two cases that highlight different weaknesses of answer aggregation. In Case I, an incorrect answer has higher maximum path confidence, while a correct answer has stronger voting support. In Case II, an incorrect answer receives equal or stronger voting support, while pairwise

Table 2: Performance comparison of different mathematics-specialized models in terms of accuracy and expected calibration error (ECE). Following the RPC setting, perplexity-based confidence (PPL) is used by default. Each voting-based baseline is followed by its PGLA-enhanced variant, denoted by **+Ours**.

Method	MATH		MathOdyssey		AIME		Average	
	Acc. ↑	ECE ↓	Acc. ↑	ECE ↓	Acc. ↑	ECE ↓	Acc. ↑	ECE ↓
InternLM2-Math-Plus-1.8B								
PPL	$33.23_{\pm 0.25}$	$61.58_{\pm 0.25}$	$16.63_{\pm 0.83}$	$77.20_{\pm 0.82}$	$1.66_{\pm 0.15}$	$92.28_{\pm 0.16}$	17.17	77.02
SC	$36.48_{\pm 0.14}$	$6.66_{\pm 0.14}$	$14.57_{\pm 0.45}$	$18.51_{\pm 0.41}$	$2.66_{\pm 0.20}$	$16.84_{\pm 0.16}$	17.90	14.00
+ Ours	$\mathbf{39.77}_{\pm 0.17}$	$\mathbf{5.63}_{\pm 0.13}$	$\mathbf{18.28}_{\pm 0.56}$	$\mathbf{13.97}_{\pm 0.59}$	$\mathbf{3.37}_{\pm 0.17}$	$\mathbf{15.51}_{\pm 0.15}$	**20.47**	**11.70**
Weighted SC	$36.72_{\pm 0.12}$	$6.52_{\pm 0.14}$	$14.86_{\pm 0.47}$	$18.05_{\pm 0.50}$	$2.75_{\pm 0.23}$	$16.65_{\pm 0.19}$	18.11	13.74
+ Ours	$\mathbf{39.91}_{\pm 0.22}$	$\mathbf{5.55}_{\pm 0.17}$	$\mathbf{18.56}_{\pm 0.56}$	$\mathbf{13.47}_{\pm 0.56}$	$\mathbf{3.37}_{\pm 0.17}$	$\mathbf{15.34}_{\pm 0.15}$	**20.61**	**11.45**
Adaptive SC	$36.48_{\pm 0.16}$	$10.61_{\pm 0.23}$	$14.43_{\pm 0.51}$	$22.69_{\pm 0.73}$	$2.65_{\pm 0.20}$	$19.25_{\pm 0.25}$	17.85	17.52
+ Ours	$\mathbf{39.05}_{\pm 0.23}$	$\mathbf{9.28}_{\pm 0.27}$	$\mathbf{17.47}_{\pm 0.63}$	$\mathbf{17.96}_{\pm 0.60}$	$\mathbf{3.23}_{\pm 0.21}$	$\mathbf{17.53}_{\pm 0.41}$	**19.92**	**14.92**
RPC	$37.87_{\pm 0.14}$	$6.45_{\pm 0.15}$	$16.43_{\pm 0.60}$	$16.25_{\pm 0.76}$	$3.28_{\pm 0.23}$	$16.32_{\pm 0.20}$	19.19	13.01
+ Ours	$\mathbf{40.29}_{\pm 0.27}$	$\mathbf{5.50}_{\pm 0.22}$	$\mathbf{20.28}_{\pm 0.48}$	$\mathbf{10.78}_{\pm 0.60}$	$\mathbf{3.57}_{\pm 0.21}$	$\mathbf{14.67}_{\pm 0.23}$	**21.38**	**10.32**
InternLM2-Math-Plus-7B								
PPL	$47.00_{\pm 0.20}$	$48.98_{\pm 0.20}$	$27.51_{\pm 1.18}$	$67.54_{\pm 1.18}$	$5.96_{\pm 0.48}$	$88.98_{\pm 0.49}$	26.82	68.50
SC	$50.57_{\pm 0.18}$	$6.72_{\pm 0.19}$	$28.30_{\pm 0.61}$	$12.18_{\pm 0.57}$	$9.40_{\pm 0.21}$	$14.35_{\pm 0.23}$	29.42	11.08
+ Ours	$\mathbf{53.99}_{\pm 0.18}$	$\mathbf{6.10}_{\pm 0.20}$	$\mathbf{35.30}_{\pm 0.55}$	$\mathbf{9.69}_{\pm 1.02}$	$\mathbf{9.95}_{\pm 0.24}$	$\mathbf{13.90}_{\pm 0.23}$	**33.08**	**9.90**
Weighted SC	$50.81_{\pm 0.20}$	$6.56_{\pm 0.21}$	$28.82_{\pm 0.67}$	$11.57_{\pm 0.68}$	$9.45_{\pm 0.23}$	$14.27_{\pm 0.25}$	29.69	10.80
+ Ours	$\mathbf{53.99}_{\pm 0.13}$	$\mathbf{6.15}_{\pm 0.17}$	$\mathbf{35.78}_{\pm 0.51}$	$\mathbf{9.37}_{\pm 1.16}$	$\mathbf{9.88}_{\pm 0.28}$	$\mathbf{13.86}_{\pm 0.27}$	**33.22**	**9.79**
Adaptive SC	$50.54_{\pm 0.22}$	$11.02_{\pm 0.23}$	$28.20_{\pm 0.68}$	$18.17_{\pm 1.02}$	$9.38_{\pm 0.23}$	$17.39_{\pm 0.41}$	29.37	15.53
+ Ours	$\mathbf{52.66}_{\pm 0.26}$	$\mathbf{10.53}_{\pm 0.35}$	$\mathbf{33.14}_{\pm 0.73}$	$\mathbf{13.22}_{\pm 1.52}$	$\mathbf{9.83}_{\pm 0.21}$	$\mathbf{16.45}_{\pm 0.33}$	**31.87**	**13.40**
RPC	$51.94_{\pm 0.14}$	$\mathbf{6.41}_{\pm 0.16}$	$31.70_{\pm 0.71}$	$9.88_{\pm 0.80}$	$9.74_{\pm 0.23}$	$14.31_{\pm 0.21}$	31.13	10.20
+ Ours	$\mathbf{53.81}_{\pm 0.12}$	$6.72_{\pm 0.15}$	$\mathbf{36.53}_{\pm 1.06}$	$\mathbf{7.90}_{\pm 1.09}$	$\mathbf{10.08}_{\pm 0.24}$	$\mathbf{13.31}_{\pm 0.22}$	**33.47**	**9.31**
Deepseek-Math-RL-7B								
PPL	$42.50_{\pm 0.23}$	$53.84_{\pm 0.23}$	$22.34_{\pm 1.00}$	$73.85_{\pm 0.99}$	$3.37_{\pm 0.46}$	$93.38_{\pm 0.46}$	22.74	73.69
SC	$53.33_{\pm 0.08}$	$\mathbf{6.43}_{\pm 0.11}$	$36.57_{\pm 0.56}$	$9.40_{\pm 1.21}$	$9.42_{\pm 0.23}$	$12.12_{\pm 0.26}$	33.11	9.32
+ Ours	$\mathbf{53.52}_{\pm 0.10}$	$6.86_{\pm 0.09}$	$\mathbf{37.87}_{\pm 0.55}$	$\mathbf{9.31}_{\pm 0.95}$	$\mathbf{10.02}_{\pm 0.23}$	$\mathbf{10.88}_{\pm 0.25}$	**33.80**	**9.02**
Weighted SC	$53.35_{\pm 0.09}$	$\mathbf{6.43}_{\pm 0.10}$	$36.68_{\pm 0.61}$	$9.55_{\pm 1.31}$	$9.43_{\pm 0.31}$	$12.10_{\pm 0.34}$	33.16	9.36
+ Ours	$\mathbf{53.53}_{\pm 0.10}$	$6.84_{\pm 0.09}$	$\mathbf{37.94}_{\pm 0.58}$	$\mathbf{9.53}_{\pm 0.79}$	$\mathbf{10.00}_{\pm 0.24}$	$\mathbf{10.84}_{\pm 0.25}$	**33.82**	**9.07**
Adaptive SC	$53.28_{\pm 0.09}$	$\mathbf{10.01}_{\pm 0.14}$	$\mathbf{36.30}_{\pm 0.58}$	$13.25_{\pm 1.14}$	$9.36_{\pm 0.22}$	$14.88_{\pm 0.41}$	32.98	12.71
+ Ours	$\mathbf{53.44}_{\pm 0.16}$	$10.10_{\pm 0.15}$	$36.27_{\pm 0.80}$	$\mathbf{12.33}_{\pm 0.87}$	$\mathbf{9.92}_{\pm 0.23}$	$\mathbf{13.31}_{\pm 0.49}$	**33.21**	**11.91**
RPC	$53.36_{\pm 0.11}$	$\mathbf{6.47}_{\pm 0.12}$	$37.12_{\pm 0.61}$	$9.54_{\pm 1.54}$	$9.51_{\pm 0.28}$	$11.94_{\pm 0.32}$	33.33	9.32
+ Ours	$\mathbf{53.52}_{\pm 0.11}$	$6.85_{\pm 0.10}$	$\mathbf{38.10}_{\pm 0.61}$	$\mathbf{8.89}_{\pm 0.76}$	$\mathbf{10.01}_{\pm 0.26}$	$\mathbf{10.44}_{\pm 0.29}$	**33.88**	**8.73**

Table 3: Accuracy (%) using Qwen2.5-Math-PRM-7B scores at sampling budgets $K/2$ and K ($K = 64$ for MATH; 128 otherwise). **+ Ours** denotes the PGLA-enhanced voting rule.

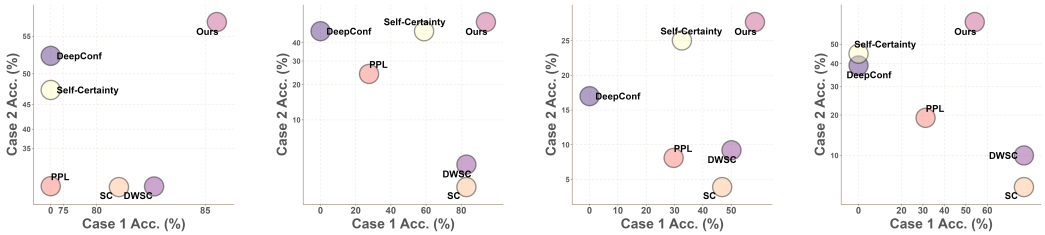
Method	MATH		MathOdyssey		AIME		Average	
	$K/2 \uparrow$	$K \uparrow$	$K/2 \uparrow$	$K \uparrow$	$K/2 \uparrow$	$K \uparrow$	$K/2 \uparrow$	$K \uparrow$
InternLM2-Math-Plus-7B								
PRM-Weighted SC	$57.56_{\pm 0.31}$	$58.67_{\pm 0.10}$	$33.57_{\pm 1.13}$	$34.34_{\pm 0.58}$	$15.49_{\pm 0.36}$	$16.23_{\pm 0.48}$	35.54	36.42
+ Ours	$\mathbf{58.86}_{\pm 0.29}$	$\mathbf{60.40}_{\pm 0.24}$	$\mathbf{37.63}_{\pm 1.18}$	$\mathbf{38.79}_{\pm 0.62}$	$\mathbf{15.55}_{\pm 0.42}$	$\mathbf{16.57}_{\pm 0.35}$	**37.35**	**38.59**
PRM-RPC	$57.74_{\pm 0.28}$	$58.90_{\pm 0.13}$	$34.83_{\pm 1.18}$	$35.48_{\pm 0.40}$	$\mathbf{15.57}_{\pm 0.42}$	$16.31_{\pm 0.44}$	36.05	36.90
+ Ours	$\mathbf{58.88}_{\pm 0.24}$	$\mathbf{60.41}_{\pm 0.25}$	$\mathbf{37.63}_{\pm 1.24}$	$\mathbf{38.87}_{\pm 0.60}$	$15.53_{\pm 0.37}$	$\mathbf{16.50}_{\pm 0.28}$	**37.35**	**38.59**
DeepSeek-Math-RL-7B								
PRM-Weighted SC	$60.76_{\pm 0.25}$	$62.04_{\pm 0.20}$	$39.69_{\pm 0.59}$	$40.28_{\pm 0.53}$	$15.77_{\pm 0.27}$	$16.93_{\pm 0.43}$	38.74	39.75
+ Ours	$\mathbf{61.17}_{\pm 0.24}$	$\mathbf{62.41}_{\pm 0.20}$	$\mathbf{40.18}_{\pm 0.64}$	$\mathbf{40.69}_{\pm 0.53}$	$\mathbf{16.13}_{\pm 0.48}$	$\mathbf{17.49}_{\pm 0.57}$	**39.16**	**40.20**
PRM-RPC	$60.91_{\pm 0.27}$	$62.26_{\pm 0.22}$	$\mathbf{40.05}_{\pm 0.83}$	$40.67_{\pm 0.51}$	$15.85_{\pm 0.32}$	$17.05_{\pm 0.36}$	38.94	39.99
+ Ours	$\mathbf{61.19}_{\pm 0.23}$	$\mathbf{62.55}_{\pm 0.19}$	$39.85_{\pm 0.75}$	$\mathbf{40.69}_{\pm 0.42}$	$\mathbf{16.14}_{\pm 0.51}$	$\mathbf{17.52}_{\pm 0.66}$	**39.06**	**40.26**

confidence comparisons favor a correct answer. Best-of-N often fails in Case I, while voting-based methods typically struggle in Case II. As shown in Fig. 5 and Fig. 10, PGLA better balances the two cases by combining voting support with pairwise comparisons. See App. C for detailed definitions and examples.

4.4 Fine-grained Analysis

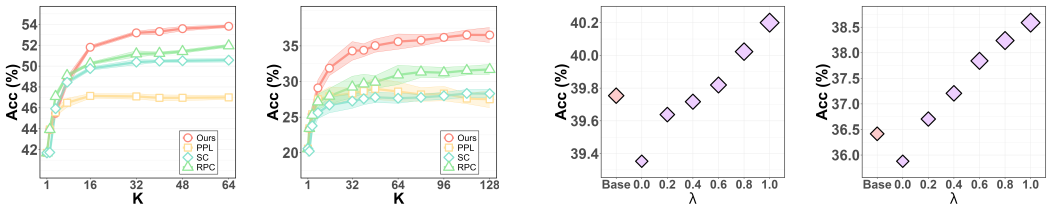
Effect of Confidence Reliability. Larger λ indicates more reliable PRM scores for pairwise comparisons. As shown in Fig. 7, more reliable scores lead to higher PGLA accuracy on both models. Base denotes the PRM-Weighted SC baseline. Experimental details are provided in App. G.5.

Scaling with Varying K Values We examine how the number of sampled reasoning paths affects aggregation performance on MATH and MathOdyssey using **InternLM2-Math-Plus-7B**. Fig. 6



(a) AIME24, 1.7B (b) AIME24, 8B (c) AIME25, 1.7B (d) AIME25, 8B

Figure 5: Tradeoff between Case I and Case II on AIME24 and AIME25 using Qwen3-1.7B and Qwen3-8B. DWSC denotes the DeepConf-Weighted SC method.



(a) MATH (b) MathOdyssey (a) DeepSeek (b) InternLM2

Figure 6: Scaling behavior under varying numbers of sampled reasoning paths K on MATH and MathOdy. using InternLM2-Math-Plus-7B.

Figure 7: Effect of confidence reliability in pairwise comparisons on accuracy, averaged over three math datasets.

varies K over $\{1, 2, 4, 8, 16, 32, 40, 48, 64, 80, 96, 112, 128\}$. Increasing K generally improves performance by providing richer candidate sets, and PGLA remains consistently strong across most of the sampling range.

Performance using High Sampling Temperatures. To further examine robustness under more diverse sampled reasoning paths, we conduct additional experiments on MathOdyssey and AIME using InternLM2-Math-Plus-1.8B, InternLM2-Math-Plus-7B, and DeepSeek-Math-RL-7B. At higher sampling temperatures $T = 1.1$ and $T = 1.3$, PGLA generally outperforms the comparison methods as shown in Tab. 16. This supports its robustness to more diverse reasoning paths.

Effect of Various Comparative Advantage Functions. We compare the default voting-weighted rule with uniform averaging, Copeland, and Maximin in App. G.11. These rules generally improve accuracy over the voting baselines, with the default achieving strong overall performance. This shows that PGLA benefits from pairwise evidence under different comparative advantage rules.

Performance in the Online Setting. As shown in Tab. 4, Online SC+PGLA improves accuracy over full-budget SC by 0.28–3.63% while reducing the sampling cost by 50.8%–73.1% on InternLM2-Math-Plus-1.8B. Full comparisons with Adaptive SC and Online SC, along with results on InternLM2-Math-Plus-7B, are provided in App. G.9.

Table 4: Online setting on InternLM2-1.8B.

Dataset	Method	Acc.	$\overline{K}$	Saving
MATH	SC (Full Budget)	$36.48_{\pm 0.14}$	64.0	–
	Online SC+PGLA	$\mathbf{39.32}_{\pm 0.31}$	**31.5**	**50.8%**
MathOdyssey	SC (Full Budget)	$14.57_{\pm 0.45}$	128.0	–
	Online SC+PGLA	$\mathbf{18.20}_{\pm 0.90}$	**34.5**	**73.1%**
AIME	SC (Full Budget)	$2.66_{\pm 0.20}$	128.0	–
	Online SC+PGLA	$\mathbf{2.94}_{\pm 0.29}$	**38.3**	**70.1%**

5 Conclusion

Inspired by social choice theory and the Condorcet principle, we propose *Pairwise-Guided Logit Adjustment* (PGLA), a plug-and-play framework for sampling-based LLM reasoning. Through pairwise confidence comparisons between sampled reasoning paths, PGLA estimates the implicit preference structure among candidate answers and captures their comparative advantages. A KL-regularized objective combines this comparative evidence with voting support, yielding a simple adjustment to the base voting logits. The framework accommodates different voting methods, confidence sources, and comparative advantage rules. Experiments on mathematical reasoning, knowledge reasoning, and code generation demonstrate overall gains over voting baselines, while the online extension improves sampling efficiency.

REFERENCES

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. Towards a human-like open-domain chatbot. *CoRR*, abs/2001.09977, 2020.

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with LLMs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 12375–12396, 2023.

AI-MO. AIME 2024, 2024. URL https://huggingface.co/datasets/AI-MO/aimo-validation-aime.

Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program synthesis with large language models. *CoRR*, abs/2108.07732, 2021.

Vidhisha Balachandran, Jingya Chen, Lingjiao Chen, Shivam Garg, Neel Joshi, Yash Lara, John Langford, Besmira Nushi, Vibhav Vineet, Yue Wu, and Safoora Yousefi. Inference-time scaling for complex tasks: Where we stand and what lies ahead. *CoRR*, abs/2504.00294, 2025.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin T. Vechev. MathArena: Evaluating LLMs on uncontaminated math competitions. In *Advances in Neural Information Processing Systems*, 2025.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021.

Stanley F. Chen, Douglas Beeferman, and Ronald Rosenfeld. Evaluation metrics for language models. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pp. 275–280, 1998.

Wenbo Chen, Puheng Li, Mengyang Liu, Weijie Su, and Tianpei Xie. MARS: Margin-adversarial risk-controlled stopping for parallel LLM test-time scaling. *CoRR*, abs/2606.12935, 2026.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal self-consistency for large language model generation. *CoRR*, abs/2311.17311, 2023.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021.

Meng Fang, Xiangpeng Wan, Fei Lu, Fei Xing, and Kai Zou. MathOdyssey: Benchmarking mathematical problem-solving skills in large language models using odyssey math data. *Scientific Data*, 12(1):1392, 2025.

Austin Feng, Marius Alonso, Ambroise Odonnat, Vasilii Feofanov, and Ievgen Redko. Optimal self-consistency for efficient reasoning with large language models. In *International Conference on Machine Learning*, 2026.

Felix A. Fischer, Olivier Hudry, and Rolf Niedermeier. Weighted tournament solutions. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (eds.), *Handbook of Computational Social Choice*, pp. 85–102. Cambridge University Press, 2016.

Peter C. Fishburn. Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33(3): 469–489, 1977.

Yichao Fu, Xuewei Wang, Hao Zhang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. In *International Conference on Learning Representations*, pp. 94355–94377, 2026.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, 2025.

Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-N selection for large language models via self-certainty. In *Advances in Neural Information Processing Systems*, pp. 19720–19745, 2025.

Junseok Kim, Nakyeong Yang, Kyungmin Min, and Kyomin Jung. Reliability-aware adaptive self-consistency for efficient sampling in LLM reasoning. In *Findings of the Association for Computational Linguistics*, pp. 21575–21590, 2026.

Joshua Ong Jun Leang, Zheng Zhao, Aryo Pradipta Gema, Sohee Yang, Wai-Chung Kwan, Xuanli He, Wenda Li, Pasquale Minervini, Eleonora Giunchiglia, and Shay B. Cohen. PiCSAR: Probabilistic confidence selection and ranking for reasoning chains. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 31511–31544, 2026.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, pp. 3843–3857, 2022.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. In *International Conference on Learning Representations*, 2024.

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*, 2024.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*, pp. 21558–21572, 2023.

Maria Marina, Daniil Moskovskiy, Sergey Pletenev, Mikhail Salnikov, Alexander Panchenko, and Viktor Moskvoretskii. Boosting self-consistency with ranking. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 1017–1032, 2026.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021.

Jungsuk Oh and Jay-Yoon Lee. Latent self-consistency for reliable majority-set selection in short- and long-answer reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 32591–32599, 2026.

OpenCompass. AIME 2025, 2025. URL https://huggingface.co/datasets/opencompass/AIME2025.

Qwen. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. In *Conference on Language Modeling*, 2024.

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

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. Confidence improves self-consistency in LLMs. In *Findings of the Association for Computational Linguistics*, pp. 20090–20111, 2025.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Soft self-consistency improves language models agents. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 287–301, 2024.

Weiqin Wang, Yile Wang, and Hui Huang. Ranked voting based self-consistency of large language models. In *Findings of the Association for Computational Linguistics*, pp. 14410–14426, 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, pp. 24824–24837, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, pp. 11809–11822, 2023.

Huaiyuan Ying, Shuo Zhang, Linyang Li, Zhejian Zhou, Yunfan Shao, Zhaoye Fei, Yichuan Ma, Jiawei Hong, Kuikun Liu, Ziyi Wang, Yudong Wang, Zijian Wu, Shuaibin Li, Fengzhe Zhou, Hongwei Liu, Songyang Zhang, Wenwei Zhang, Hang Yan, Xipeng Qiu, Jiayu Wang, Kai Chen, and Dahua Lin. Internlm-math: Open math large language models toward verifiable reasoning. *CoRR*, abs/2402.06332, 2024.

H. P. Young. Condorcet's theory of voting. *American Political Science Review*, 82(4):1231–1244, 1988.

Parvez Zamil and Gollam Rabby. AIME problems 1983 to 2024, 2024. URL https://www.kaggle.com/datasets/tourist800/aime-problems-1983-to-2024.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics*, pp. 10495–10516, 2025.

Zhi Zhou, Yuhao Tan, Zenan Li, Yuan Yao, Lan-Zhe Guo, Yu-Feng Li, and Xiaoxing Ma. A theoretical study on bridging internal probability and self-consistency for LLM reasoning. In *Advances in Neural Information Processing Systems*, pp. 87380–87413, 2025.

William S. Zwicker. Introduction to the theory of voting. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (eds.), *Handbook of Computational Social Choice*, pp. 23–56. Cambridge University Press, 2016.

APPENDIX CONTENTS

A Related Work **14**

B Representative Methods **15**

B.1 Path-Confidence Measures and Selection 15

B.2 Answer-Level Voting 16

C Case Analysis **16**

D Online PGLA Extension **17**

E Proof for Proposition 3.1 **18**

F Additional Experimental Setup **19**

F.1 Comparison Methods 19

F.2 LLM Backbones 20

F.3 Datasets 20

F.4 Implementation Details 21

G Additional Experimental Results **22**

G.1 Enhanced Code Tests in Reasoning Mode 22

G.2 Comparison of Non-Thinking and Reasoning Modes 22

G.3 Performance with the External Reward Setting 24

G.4 Tradeoff between Case I and Case II 25

G.5 Effect of Ranking Quality by Dataset 26

G.6 Temperature Sensitivity 26

G.7 Efficiency Comparison 26

G.8 Complete Scaling Results with Varying K 29

G.9 Online Sampling Efficiency 29

G.10 Extension to 32B Models 33

G.11 Effect of Comparative Advantage Rules 34

G.12 Computational Overhead 36

G.13 Connection to Rank-Based Borda Aggregation 37

H Impact Statement **39**

I AI Use Statement **39**

A Related Work

Sampling-based reasoning and confidence aggregation. Sampling-based test-time scaling improves LLM reasoning by generating multiple reasoning paths and selecting an answer from their collective evidence. Self-Consistency (SC) aggregates paths through majority voting over their final answers (Wang et al., 2023), whereas Best-of-N selects a single candidate using a quality score, such as a learned verifier or process reward model (Cobbe et al., 2021; Lightman et al., 2024). For outputs that cannot be grouped by exact answers, Universal Self-Consistency uses an LLM to assess consistency (Chen et al., 2023), while Latent Self-Consistency learns representations for semantic agreement (Oh & Lee, 2026).

Confidence-aware methods further exploit differences in path quality. Soft Self-Consistency scores candidate actions using model likelihoods in interactive tasks (Wang et al., 2024). RPC combines internal probability with answer consistency through Perplexity Consistency and Reasoning Pruning (Zhou et al., 2025). CISC weights votes by model confidence and shows that distinguishing correct from incorrect responses *within the same question* is more relevant to aggregation than cross-question calibration alone (Taubenfeld et al., 2025). Self-Certainty derives confidence from the token distribution (Kang et al., 2025), while DeepConf uses local confidence to filter traces and weight their votes (Fu et al., 2026). More recently, PiCSAR combines reasoning and answer likelihoods for candidate selection (Leang et al., 2026). Self-Certainty's Borda variant ranks sampled paths by confidence and accumulates rank-dependent votes for their answers (Kang et al., 2025). Ranked Voting SC instead elicits ordered candidate answers within each response and aggregates these lists using ranked voting rules (Wang et al., 2025). RISC learns an answer ranker from frequency, semantic, and reasoning-consistency features (Marina et al., 2026).

Our work studies preference reconstruction from ordinary single-answer samples. Each path provides one answer and a confidence score, leaving its ordering over alternatives unobserved. We seek to reconstruct an *answer-level preference structure* through pairwise confidence comparisons across answer groups, and reconcile this proxy for the missing preferences with voting support through KL-regularized aggregation.

Efficient and adaptive reasoning. The benefits of additional inference computation depend on how it is allocated across problems and reasoning paths (Snell et al., 2025). Adaptive-Consistency and Early-Stopping Self-Consistency reduce the number of completed samples by monitoring answer agreement and stopping when sufficient evidence has accumulated (Aggarwal et al., 2023; Li et al., 2024). DeepConf additionally monitors confidence during generation, terminates low-confidence traces, and uses answer consensus to stop further sampling (Fu et al., 2026).

Recent work refines both allocation and stopping. Blend-ASC analyzes SC through mode estimation and voting theory and dynamically distributes a sampling budget across questions (Feng et al., 2026). ReASC incorporates response-level confidence into its evidence accumulation and stopping decisions (Kim et al., 2026). MARS probes intermediate answers during parallel generation and stops when a conservative bound on future vote changes protects the current leader; its theoretical target is agreement with the full-budget vote (Chen et al., 2026). Our primary focus is the preference structure used for answer aggregation. The online extension updates this structure as paths arrive and stops when the resulting decision remains stable, allowing us to examine whether preference-aware aggregation also improves sample efficiency.

B REPRESENTATIVE METHODS

We follow the notation and scoring rules in Sec. 2.1. The K sampled paths z_i produce answers $a_i = g(z_i)$ and confidence scores p_i, forming $\mathcal{D} = \{(a_i, p_i)\}_{i=1}^{K}$. For each candidate answer $a \in \mathcal{A}$, an aggregation method $\mathcal{M}$ computes $s_a^{\mathcal{M}} = \mathcal{M}(a; \mathcal{D})$; the predicted answer $\hat{a}$ is selected by Eq. (1). We first specify possible confidence scores p_i and then apply the same selection and voting formulas.

B.1 PATH-CONFIDENCE MEASURES AND SELECTION

Let T_i be the number of tokens in path z_i, and let $z_{i,t}$ be its t-th token. For a vocabulary token v, $P_{i,t}(v)$ denotes the model's predicted probability of v at position t, conditioned on the input query and preceding tokens of z_i. The vocabulary is denoted by $\mathcal{V}_{\text{tok}}$. The following superscripts distinguish alternative choices for the confidence p_i in Sec. 2.1.

Perplexity confidence uses inverse perplexity so that higher values indicate higher confidence (Chen et al., 1998):

$$p_i^{\text{PPL}} = \exp\left(\frac{1}{T_i}\sum_{t=1}^{T_i}\log P_{i,t}(z_{i,t})\right) = \text{PPL}(z_i)^{-1}. \tag{11}$$

Here, $\text{PPL}(z_i)$ is the perplexity of the sampled path. Its negative log-perplexity is $\log p_i^{\text{PPL}}$; exponentiation preserves its ordering and ties, so either scale gives the same Best-of-N selection and pairwise confidence comparisons. The positive scale above also supplies the original perplexity-derived voting contribution directly as p_i in Eq. (4).

Self-Certainty (Kang et al., 2025) measures deviation from the uniform distribution over the vocabulary:

$$p_i^{\text{SCert}} = -\frac{1}{T_i|\mathcal{V}_{\text{tok}}|}\sum_{t=1}^{T_i}\sum_{v\in\mathcal{V}_{\text{tok}}}\log(|\mathcal{V}_{\text{tok}}|P_{i,t}(v))\,. \tag{12}$$

The notation $|\mathcal{V}_{\text{tok}}|$ denotes the vocabulary size. Unlike perplexity, this score uses the entire predicted token distribution at each position, rather than only the probability of the sampled token.

DeepConf (Fu et al., 2026) computes token confidences and aggregates them over local windows. Let $k \in \{1, \ldots, |\mathcal{V}_{\text{tok}}|\}$ be the number of highest-probability tokens used at each position, and let $\text{Top}_k(P_{i,t})$ be that set of vocabulary tokens. Its bottom-10% variant is

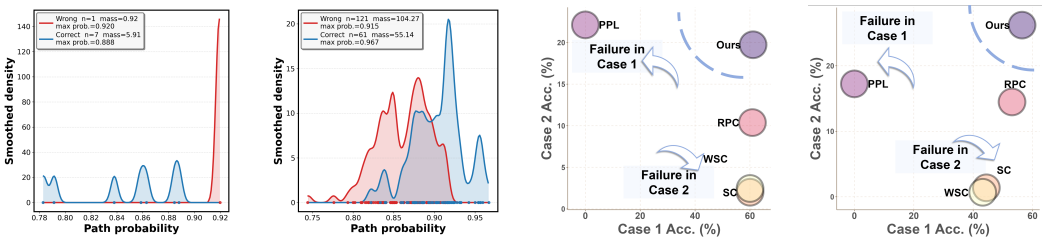
$$\begin{aligned} c_{i,t} &= -\frac{1}{k}\sum_{v\in\text{Top}_k(P_{i,t})}\log P_{i,t}(v), \quad G_{i,\ell} = \frac{1}{|W_{i,\ell}|}\sum_{t\in W_{i,\ell}} c_{i,t}, \\ p_i^{\text{DeepConf}} &= \frac{1}{|\mathcal{L}_i|}\sum_{\ell\in\mathcal{L}_i} G_{i,\ell}. \end{aligned} \tag{13}$$

Here, $c_{i,t}$ is the token confidence; ℓ indexes a local window; $W_{i,\ell} \subseteq \{1, \ldots, T_i\}$ is its nonempty set of token positions; and $G_{i,\ell}$ is the average confidence in that window. The nonempty index set $\mathcal{L}_i$ selects the lowest-10% window confidences, with at least one window retained. Vertical bars denote set size. Tail and minimum-window variants use the same local-confidence principle. We use $k = 20$, following DeepConf. This score averages negative log-probabilities over the top-k candidate tokens and differs from the negative log-likelihood of the sampled token. Larger values are treated as higher confidence. The resulting token scores and their window averages are nonnegative, allowing direct use as voting weights.

For any chosen confidence p_i, Best-of-N uses exactly the answer-level score in Eq. (2):

$$s_a^{\text{BoN}} = \max_{i:a_i=a} p_i, \qquad a \in \mathcal{A}. \tag{14}$$

The index condition $a_i = a$ restricts the maximum to paths producing answer a. Best-of-N is the conventional method name; its sampling budget here is the K paths defined in Sec. 2.1.



(a) Case I. (b) Case II. (c) MATH. (d) MathOdyssey.

Figure 8: Complementary roles of voting support and pairwise confidence comparisons. In Case I, an incorrect answer contains the highest-confidence path despite weaker voting support. In Case II, an incorrect answer has stronger or tied voting support, while pairwise confidence comparisons favor a correct answer. The last two panels report the corresponding behavior on MATH and MathOdyssey.

B.2 Answer-Level Voting

Self-Consistency (SC) (Wang et al., 2023) assigns one vote to each sampled path, as in Eq. (3):

$$s_a^{\mathrm{SC}} = \sum_{i=1}^{K} \mathbf{1}(a_i = a). \tag{15}$$

The indicator $\mathbf{1}(\cdot)$ is one when its condition holds and zero otherwise. Thus s_a^{SC} counts paths supporting answer a without using their confidence values.

Confidence-based voting uses the same coefficient $h_i \geq 0$ as Eq. (4), with $h_i = 0$ for a filtered path:

$$s_a^{\text{Conf-Vote}} = \sum_{i=1}^{K} h_i \cdot p_i \, \mathbf{1}(a_i = a). \tag{16}$$

Unfiltered Weighted SC sets $h_i = 1$ for every path. PPL-Weighted SC uses $p_i = p_i^{\mathrm{PPL}}$ from Eq. (11); DeepConf-Weighted SC and Self-Certainty-Weighted SC use p_i^{DeepConf} and p_i^{SCert}, respectively. Confidence filtering instead sets $h_i = 0$ for discarded paths and $h_i = 1$ for retained paths.

RPC (Zhou et al., 2025) is another instance of this confidence-based voting rule. Perplexity Consistency aggregates probability-derived contributions from distinct path–confidence pairs, while Reasoning Pruning filters low-confidence paths under its original query-specific mixture rule. With $p_i = p_i^{\mathrm{PPL}}$, set $h_i = 1$ for one representative of each retained distinct pair (z_i, p_i), and $h_i = 0$ for pruned paths and repeated copies of the same pair. Its score is therefore

$$s_a^{\mathrm{RPC}} = \sum_{i=1}^{K} h_i \cdot p_i \, \mathbf{1}(a_i = a), \tag{17}$$

where the superscript RPC identifies this particular choice of confidence and filtering coefficients in the Conf-Vote family. Further baseline-specific rules, including rank-weighted voting, are described in App. F.1.

C Case Analysis

We use the Best-of-N, SC, and confidence-based voting scores in Eqs. (2) to (4). Fix a voting method $\mathcal{M}$: its score $s_a^{\mathcal{M}}$ is either s_a^{SC} or an instance of $s_a^{\text{Conf-Vote}}$ with specified confidence scores p_i and coefficients h_i. Thus, Best-of-N uses a confidence maximum, while voting sums the path contributions prescribed in Sec. 2.1.

Let $c, w \in \mathcal{A}$ be two distinct observed answers, with c correct and w incorrect. Each has at least one supporting path, so s_c^{SC} and s_w^{SC} are positive. In **Case I**, voting support favors c, whereas Best-of-N favors w because an incorrect path has the higher confidence maximum:

$$s_c^{\mathcal{M}} > s_w^{\mathcal{M}}, \qquad s_w^{\mathrm{BoN}} > s_c^{\mathrm{BoN}}. \tag{18}$$

To describe **Case II**, define the empirical confidence preference for c over w:

$$\widehat{\rho}_{cw} := \frac{1}{s_c^{\mathrm{SC}} s_w^{\mathrm{SC}}} \sum_{i:a_i=c} \sum_{j:a_j=w} \left[\mathbf{1}(p_i > p_j) + \frac{1}{2}\mathbf{1}(p_i = p_j) \right]. \tag{19}$$

Here, i and j index observed paths supporting c and w, respectively. The denominator counts all such path pairs, so $\widehat{\rho}_{cw}$ is the fraction favoring c, with confidence ties counted as one half. Case II occurs when voting support favors or ties w, while these pairwise comparisons favor c:

$$s_w^{\mathcal{M}} \geq s_c^{\mathcal{M}}, \qquad \widehat{\rho}_{cw} > \frac{1}{2}. \tag{20}$$

The confidence-based comparison entry Y_{cw} in Eq. (6) is the average signed comparison over the same path pairs; hence $Y_{cw} = 2\widehat{\rho}_{cw} - 1$. The SC-error analysis in Fig. 4 takes $\mathcal{M} = \mathrm{SC}$, giving $s_w^{\mathrm{SC}} \geq s_c^{\mathrm{SC}}$ and $Y_{cw} > 0$.

Conditioning on recoverable SC errors. We call an SC error *recoverable* if its candidate pool contains at least one correct answer, and denote the set of such pools by $\mathcal{T}_{\mathrm{rec}}$. For each pool t, let $e_t \in [0, 1]$ be its SC error weight, with fractional credit used for ties. We measure the prevalence of Case II among recoverable SC errors by

$$R_{\mathrm{II}}^{\mathrm{rec}} := \frac{\sum_{t \in \mathcal{T}_{\mathrm{rec}}} e_t \, \mathbf{1}\{\text{Case II}_t\}}{\sum_{t \in \mathcal{T}_{\mathrm{rec}}} e_t}. \tag{21}$$

This conditioning excludes only pools with no correct candidate, for which reranking over the observed answers is impossible.

Tab. 5 reports both the share of NoCorrect pools among all SC errors and the Case-II share after excluding them. We use PPL confidence, with $K = 64$ on MATH and $K = 128$ on MathOdyssey and AIME.

Table 5: Breakdown of SC errors (%). NoCorrect is measured over all SC errors; recoverable Case II is measured after excluding NoCorrect pools. Results use PPL confidence.

Model	MATH	MathOdyssey	AIME
NoCorrect / all SC errors			
InternLM2-Math-Plus-1.8B	50.16	46.94	75.83
InternLM2-Math-Plus-7B	48.58	53.10	63.70
DeepSeek-Math-RL-7B	51.83	59.42	62.18
Case II / recoverable SC errors			
InternLM2-Math-Plus-1.8B	82.22	83.81	91.62
InternLM2-Math-Plus-7B	79.30	88.60	88.62
DeepSeek-Math-RL-7B	69.87	81.42	84.81

These cases reveal complementary weaknesses: Best-of-N is susceptible to Case I because it relies on a single confidence maximum, whereas voting-based methods are susceptible to Case II because they do not explicitly account for answer-level pairwise confidence comparisons. Pairwise-Guided Logit Adjustment (PGLA) combines the two sources of evidence, using voting support to temper isolated confidence outliers and comparative advantage to compensate for weaker voting support. This combination yields a better empirical tradeoff between the two error types, as shown in Figs. 5 and 10.

D Online PGLA Extension

The full-budget formulation above applies PGLA after collecting all sampled paths. To examine whether its comparative evidence can reduce unnecessary inference-time sampling, we extend it to the **online setting**. As new paths become available, PGLA updates its answer prediction and stops sampling once the prediction becomes stable.

Let t_j denote the number of observed paths at the j-th update, and let $\mathcal{D}_j = \{(a_i, p_i)\}_{i=1}^{t_j}$ be this prefix of the evidence pool. Its observed answers form $\mathcal{A}_j$, with supporting-path groups $C_{u,j} =$

$\{i \le t_j : a_i = u\}$ for $u \in \mathcal{A}_j$. From this prefix, we update the voting support $s^{\mathcal{V}}_{u,j}$ and comparative advantage $s^{\mathrm{Pair}}_{u,j}$ using Eqs. (6) and (7), with all answer sums restricted to $\mathcal{A}_j$. We set $s^{\mathrm{Pair}}_{u,j} = 0$ when no other observed answer has positive voting support, including the single-answer case. The adjusted score is

$$s^{\mathsf{PGLA}}_{u,j} = s^{\mathcal{V}}_{u,j} \exp(s^{\mathrm{Pair}}_{u,j}). \tag{22}$$

For $s^{\mathcal{V}}_{u,j} > 0$, this is equivalently $\log s^{\mathsf{PGLA}}_{u,j} = \log s^{\mathcal{V}}_{u,j} + s^{\mathrm{Pair}}_{u,j}$, as in Eq. (10); the multiplicative form also applies to zero voting support without adding a smoothing term.

Let $\hat{a}_j$ denote the unique highest-scoring answer in $\mathcal{A}_j$, with $\hat{a}_j = \varnothing$ under a tie. Given a stability length L, sampling stops at the first $j \ge L$ satisfying

$$\hat{a}_{j-L+1} = \cdots = \hat{a}_j \neq \varnothing. \tag{23}$$

A tie resets the stability count, and sampling continues to $K_{\max}$ if the condition is never met. This label-free rule applies unchanged to a base aggregator and its PGLA-adjusted counterpart, enabling a matched online comparison.

E Proof for Proposition 3.1

Proof. Consider an observed candidate pool with answer set $\mathcal{A}$ and $U = |\mathcal{A}| \ge 1$. The voting distribution $\bar{\mathbf{s}}^{\mathcal{V}}$ and comparative advantages s^{Pair}_u are determined by this pool and are constants with respect to the optimization variable $\bar{\mathbf{s}}$. They may be computed from the same paths; no statistical independence between them is required. Under the proposition's positive-support assumption,

$$\bar{s}^{\mathcal{V}}_u = \frac{s^{\mathcal{V}}_u}{\sum_{v\in\mathcal{A}} s^{\mathcal{V}}_v} > 0, \qquad \sum_{u\in\mathcal{A}} \bar{s}^{\mathcal{V}}_u = 1, \tag{24}$$

and each $s^{\mathrm{Pair}}_u \in [-1, 1]$ is finite. For $U = 1$, we take $s^{\mathrm{Pair}}_u = 0$ because there is no competing answer.

The objective in Eq. (8) is

$$\begin{aligned}\mathcal{J}(\bar{\mathbf{s}}) &:= \sum_{u\in\mathcal{A}} \bar{s}_u s^{\mathrm{Pair}}_u - \sum_{u\in\mathcal{A}} \bar{s}_u \log \frac{\bar{s}_u}{\bar{s}^{\mathcal{V}}_u} \\ &= -\sum_{u\in\mathcal{A}} \bar{s}_u \log \bar{s}_u + \sum_{u\in\mathcal{A}} \bar{s}_u \big(s^{\mathrm{Pair}}_u + \log \bar{s}^{\mathcal{V}}_u\big),\end{aligned} \tag{25}$$

over the probability simplex $\Delta_{\mathcal{A}} = \{\bar{\mathbf{s}} : \bar{s}_u \ge 0, \quad \sum_{u\in\mathcal{A}} \bar{s}_u = 1\}$. Terms with $\bar{s}_u = 0$ are defined by continuity, so $0 \log 0 = 0$ and $0 \log(0/\bar{s}^{\mathcal{V}}_u) = 0$.

Existence and strict concavity. The function $x \mapsto x \log x$, with its continuous extension at zero, is strictly convex on $[0, 1]$. Thus the entropy term $-\sum_{u\in\mathcal{A}} \bar{s}_u \log \bar{s}_u$ is strictly concave on the simplex, while the second term in Eq. (25) is linear in $\bar{\mathbf{s}}$. Consequently, $\mathcal{J}$ is strictly concave for $U \ge 2$. It is also continuous on the compact simplex, so a maximizer exists and is unique. For $U = 1$, the simplex contains only the distribution assigning probability one to its sole answer, and uniqueness is immediate.

Closed-form solution. Define the positive normalizing constant and a candidate distribution by

$$Z := \sum_{v\in\mathcal{A}} \bar{s}^{\mathcal{V}}_v \exp(s^{\mathrm{Pair}}_v), \qquad \bar{s}^{\star}_u := \frac{\bar{s}^{\mathcal{V}}_u \exp(s^{\mathrm{Pair}}_u)}{Z}. \tag{26}$$

Since $\bar{s}^{\mathcal{V}}_u > 0$ and s^{Pair}_u is finite, $0 < Z < \infty$, $\bar{s}^{\star}_u > 0$, and $\sum_{u\in\mathcal{A}} \bar{s}^{\star}_u = 1$. Hence $\bar{\mathbf{s}}^{\star}$ is feasible. Moreover, $\log \bar{s}^{\star}_u = \log \bar{s}^{\mathcal{V}}_u + s^{\mathrm{Pair}}_u - \log Z$. For every $\bar{\mathbf{s}} \in \Delta_{\mathcal{A}}$, including its boundary, substitution gives

$$\begin{aligned}\mathcal{J}(\bar{\mathbf{s}}) &= \sum_{u\in\mathcal{A}} \bar{s}_u \big(s^{\mathrm{Pair}}_u + \log \bar{s}^{\mathcal{V}}_u - \log \bar{s}_u\big) \\ &= \sum_{u\in\mathcal{A}} \bar{s}_u \big(\log Z + \log \bar{s}^{\star}_u - \log \bar{s}_u\big) \\ &= \log Z - D_{\mathrm{KL}}(\bar{\mathbf{s}} \,\|\, \bar{\mathbf{s}}^{\star}).\end{aligned} \tag{27}$$

The KL divergence is nonnegative and vanishes if and only if $\bar{\mathbf{s}} = \bar{\mathbf{s}}^{\star}$. Therefore $\mathcal{J}(\bar{\mathbf{s}}) \le \log Z$, with equality uniquely at $\bar{\mathbf{s}}^{\star}$. This proves that $\bar{\mathbf{s}}^{\mathsf{PGLA}} = \bar{\mathbf{s}}^{\star}$ and yields exactly Eq. (9). □

Connection to the logit adjustment. Substituting Eq. (24) into the optimizer gives

$$\begin{aligned}\bar{s}_u^{\mathsf{PGLA}} &= \frac{s_u^{\mathcal{V}} \exp(s_u^{\mathrm{Pair}})}{\sum_{v\in\mathcal{A}} s_v^{\mathcal{V}} \exp(s_v^{\mathrm{Pair}})} \\ &= \mathrm{softmax}\Big(\big(\log s_v^{\mathcal{V}} + s_v^{\mathrm{Pair}}\big)_{v\in\mathcal{A}}\Big)_u . \end{aligned} \tag{28}$$

Thus the adjusted distribution is obtained by normalizing the answer-level scores $s_u^{\mathsf{PGLA}} = s_u^{\mathcal{V}} \exp(s_u^{\mathrm{Pair}})$. For positive voting support, taking logarithms gives $\log s_u^{\mathsf{PGLA}} = \log s_u^{\mathcal{V}} + s_u^{\mathrm{Pair}}$, exactly as in Eq. (10). If all comparative advantages are equal, their common exponential factor cancels and $\bar{\mathrm{s}}^{\mathsf{PGLA}} = \bar{\mathrm{s}}^{\mathcal{V}}$.

F Additional Experimental Setup

F.1 Comparison Methods

Competitors. We compare 11 internal-confidence and voting baselines on the same sampled paths, together with three external-PRM baselines. We also include Online SC for the sampling-efficiency comparison; **+ Ours** denotes the corresponding voting rule enhanced by PGLA.

PPL Best-of-N. This baseline selects the path with the highest mean token log-probability, equivalently the lowest perplexity (Chen et al., 1998). Its answer-level score is the maximum PPL confidence among paths supporting that answer.

DeepConf Best-of-N. DeepConf summarizes token confidence over local windows to identify low-confidence portions of a path (Fu et al., 2026). We use the average of the bottom-10% window confidences as the path score and select the highest-scoring path.

Self-Certainty Best-of-N. Self-Certainty measures the divergence of the model's next-token distribution from a uniform vocabulary distribution (Kang et al., 2025). This baseline selects the path with the highest average Self-Certainty score.

SC. Self-Consistency gives each sampled path one vote for its final answer (Wang et al., 2023). It selects the answer with the largest voting support, $s_u^{\mathrm{SC}} = |C_u|$.

PPL-Weighted SC. This variant uses the inverse-perplexity confidence p_i^{PPL} in Eq. (11) and sets $h_i = 1$ in Eq. (4). It is denoted by Weighted SC in the PPL-based experiments.

DeepConf-Weighted SC. This variant sums the DeepConf path scores over each answer group (Fu et al., 2026). It uses all eligible paths, with no additional path-level confidence filter.

Self-Certainty-Weighted SC. This variant sums the Self-Certainty path scores over each answer group (Kang et al., 2025). It retains the numerical confidence values as voting weights.

Self-Certainty Borda. This rule ranks paths by decreasing Self-Certainty and assigns rank r the voting weight $(K - r + 1)^\gamma$ (Kang et al., 2025). We use $\gamma = 1.2$ and sum the rank-based weights for each answer.

DeepConf-Filter (90%). This offline rule retains paths at or above the tenth percentile of eligible path-confidence scores and sums their DeepConf weights for each answer (Fu et al., 2026). The top-90% path filter is separate from the bottom-10% window summary used to score each path; threshold ties are retained.

Adaptive SC. Adaptive Self-Consistency stops sampling according to the accumulated evidence for the leading answer (Aggarwal et al., 2023). We retain its original stopping criterion as a dynamic-budget baseline.

RPC. RPC combines Perplexity Consistency, which aggregates probability-derived support from distinct path–confidence pairs, with Reasoning Pruning, which filters low-confidence paths (Zhou et al., 2025). It provides a confidence-aware voting baseline without an external scorer.

PRM Best-of-N. This baseline uses Qwen2.5-Math-PRM-7B to score the reasoning steps of each sampled path (Zhang et al., 2025). It multiplies the step scores into a path score and selects the highest-scoring path.

PRM-Weighted SC. This variant sums the same product of step-level PRM scores over each answer group. It combines answer agreement with externally estimated path quality (Lightman et al., 2024; Zhang et al., 2025).

PRM-RPC. This variant substitutes the product of step-level PRM scores for the internal confidence weights in RPC. It retains RPC's path deduplication and pruning before aggregating the remaining voting support (Zhou et al., 2025; Zhang et al., 2025).

Online SC. This baseline updates the majority-vote prediction after every eight paths and stops after the unique winning answer remains unchanged for three consecutive updates. It uses the same stability rule as Online SC+PGLA, as detailed in App. G.9.

F.2 LLM Backbones

LLM Backbones. We use seven generators spanning the Qwen3, DeepSeekMath, and InternLM-Math families. Qwen3 models use reasoning mode unless a result is explicitly marked as non-thinking; the external PRM serves only as a scorer.

Qwen3-1.7B. This is the smallest Qwen3 generator in our experiments (Qwen, 2025). It provides a small-model setting for evaluating aggregation across mathematical reasoning and code generation.

Qwen3-4B. This Qwen3 model increases the parameter scale to 4B while retaining the family's thinking and non-thinking modes (Qwen, 2025). We include it in the confidence-transfer comparisons and complementary non-thinking evaluations.

Qwen3-8B. This 8B Qwen3 model extends the reasoning-mode comparisons to a larger backbone (Qwen, 2025). Together with the 1.7B and 4B models, it supports comparisons across model scales within the same family.

Qwen3-32B. This is the largest Qwen3 generator evaluated in our study (Qwen, 2025). We report its reasoning-mode results on AIME 2024, AIME 2025, and HMMT 2025 in App. G.10.

DeepSeek-Math-RL-7B. This math-specialized 7B model is the reinforcement-learning variant of DeepSeekMath (Shao et al., 2024). It tests aggregation on a generator trained specifically for mathematical reasoning.

InternLM2-Math-Plus-1.8B. This is the smallest math-specialized generator in our evaluation (Ying et al., 2024). It provides a lower-capacity counterpart for the mathematical-reasoning and online sampling experiments.

InternLM2-Math-Plus-7B. This 7B model belongs to the same math-specialized family as the 1.8B variant (Ying et al., 2024). It is included in the main mathematical-reasoning comparisons and the detailed aggregation analyses.

Qwen2.5-Math-PRM-7B. This process reward model assigns scores to individual reasoning steps (Zhang et al., 2025). We use it as an external scorer for the three PRM baselines and their applicable PGLA variants, separately from the seven generators above.

F.3 Datasets

Datasets. The ten benchmarks in the main experimental setup cover competition mathematics, graduate-level science, and code generation. HumanEval+ and MBPP+ provide additional evaluations with strengthened tests for the corresponding code benchmarks.

AIME 2024. This benchmark contains problems from the 2024 American Invitational Mathematics Examination (AI-MO, 2024). It evaluates short-answer mathematical reasoning and is reported separately from the historical AIME collection.

AIME 2025. This benchmark contains the 2025 AIME problems released by OpenCompass (OpenCompass, 2025). It provides a separate competition-year evaluation under the same short-answer format.

HMMT 2025. We use the February 2025 Harvard–MIT Mathematics Tournament problems provided by MathArena (Balunović et al., 2025). The benchmark covers challenging competition mathematics, including algebra, geometry, combinatorics, and number theory.

AIME. The historical AIME collection contains 993 problems from the American Invitational Mathematics Examination from 1983 to 2024 (Zamil & Rabby, 2024). This larger short-answer benchmark is used in the math-specialized-model experiments.

MATH. MATH contains challenging competition mathematics problems spanning algebra, geometry, number theory, counting, and combinatorics (Hendrycks et al., 2021). We use its 5,000-problem test split.

MathOdyssey. MathOdyssey contains 387 problems covering advanced high-school, university-level, and olympiad-level mathematics (Fang et al., 2025). It extends the evaluation beyond a single competition or curriculum level.

GPQA-Diamond (GPQA-D). GPQA comprises expert-written multiple-choice questions in biology, physics, and chemistry (Rein et al., 2024). We use its 198-question Diamond subset to evaluate graduate-level scientific reasoning.

HumanEval. HumanEval contains 164 Python programming tasks specified by function signatures and natural-language descriptions (Chen et al., 2021). We evaluate the selected programs using the original-test outcomes in our evaluation pipeline.

MBPP. Mostly Basic Programming Problems consists of short Python programming tasks with natural-language descriptions and tests (Austin et al., 2021). Our code-transfer evaluations use the cached 378-problem split.

LiveCodeBench V6. LiveCodeBench collects programming-contest problems from LeetCode, AtCoder, and Codeforces (Jain et al., 2025). We use the V6 code-generation benchmark to evaluate the correctness of selected programs by execution.

HumanEval+ and MBPP+. EvalPlus augments HumanEval and MBPP with additional tests that detect errors missed by the original suites (Liu et al., 2023). We retain the sampled programs and problem sets and report results under the strengthened tests separately from the original-test results.

F.4 Implementation Details

Sampling and consistency. For each query, all methods use matched K-path subsets of the same sampled pool, generated with common decoding parameters. The consistency function treats two paths as consistent in mathematical and knowledge reasoning when their normalized final answers match. For code generation, we follow RPC and define consistency through execution behavior (Zhou et al., 2025). For the math-specialized backbones, we set $K = 64$ on MATH and $K = 128$ on MathOdyssey and AIME. For Qwen3 comparisons in reasoning and non-thinking modes, we set $K = 32$ on AIME 2024, AIME 2025, HMMT 2025, GPQA-D, HumanEval, MBPP, and LiveCodeBench V6. We use a temperature of 1.0 and top-p of 0.95. The maximum generation length is 2,048 tokens for MATH, MathOdyssey, and AIME, and 16,384 tokens for the remaining datasets. Within each experimental setting, all methods are evaluated on the same problem set and sampled-path subsets. The same repetitions are used for each baseline and its PGLA-enhanced counterpart.We report the mean and standard deviation. All experiments are conducted on Linux servers equipped with NVIDIA A100 and RTX 3090 GPUs, except for the CPU-only aggregation microbenchmark in App. G.12.

Aggregation. For a voting rule $\mathcal{V}$, $s_u^{\mathcal{V}}$ denotes its answer-level voting support and s_u^{Pair} denotes the voting-weighted comparative advantage in Eq. (7). The default adjustment is $\log s_u^{\mathsf{PGLA}} = \log s_u^{\mathcal{V}} + s_u^{\mathrm{Pair}}$ from Eq. (10). A illustrative example of the adjustment is given in Fig. 9.

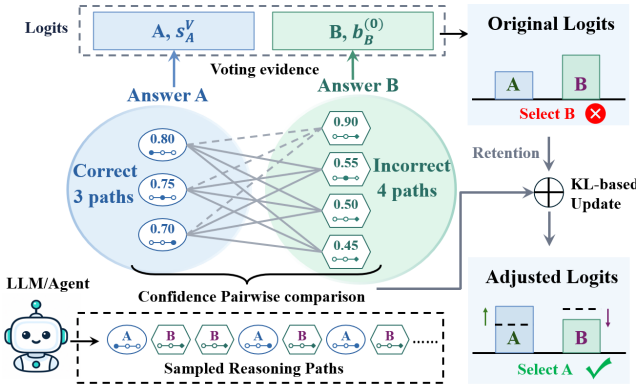


Figure 9: Illustration of PGLA.

G Additional Experimental Results

G.1 Enhanced Code Tests in Reasoning Mode

Tab. 6 reports the strengthened HumanEval+ and MBPP+ tests for the same sampled pools used in Tab. 1. A prediction must pass both the original and enhanced tests. PGLA generally improves accuracy under these strengthened tests, suggesting that its benefits persist under stricter verification.

G.2 Comparison of Non-Thinking and Reasoning Modes

Tabs. 7 and 8 compare non-thinking and reasoning modes of Qwen3-1.7B and Qwen3-4B on AIME 2024, AIME 2025, and HMMT 2025. Reasoning-mode results are reproduced from Tab. 1 for the methods shared by both modes. All columns use $K = 32$. Reasoning mode consistently achieves higher accuracy, and PGLA generally improves the voting baselines in both modes.

Table 6: Enhanced code-test accuracy (%) in reasoning mode ($K = 32$). Here **+ Ours** denotes a PGLA-enhanced variant.

Methods	Qwen3-1.7B Reasoning		Qwen3-4B Reasoning	
	HumanEval+	MBPP+	HumanEval+	MBPP+
PPL Best-of-N	$73.48_{\pm 1.80}$	$64.55_{\pm 0.99}$	$86.99_{\pm 0.55}$	$77.59_{\pm 0.74}$
DeepConf Best-of-N	$77.44_{\pm 2.36}$	$66.40_{\pm 0.76}$	$90.04_{\pm 0.99}$	$78.89_{\pm 0.76}$
Self-Certainty Best-of-N	$78.05_{\pm 2.67}$	$67.46_{\pm 0.97}$	$90.24_{\pm 1.06}$	$78.99_{\pm 0.62}$
SC	$77.29_{\pm 1.14}$	$71.16_{\pm 0.83}$	$88.01_{\pm 1.16}$	$80.58_{\pm 0.62}$
SC + Ours (DeepConf)	$81.10_{\pm 2.46}$	$71.43_{\pm 0.53}$	$90.04_{\pm 1.14}$	$80.79_{\pm 0.46}$
SC + Ours (Self-Certainty)	$\mathbf{81.25}_{\pm 1.91}$	$\mathbf{71.96}_{\pm 0.37}$	$\mathbf{90.65}_{\pm 1.36}$	$\mathbf{80.85}_{\pm 0.41}$
DeepConf-Weighted SC	$78.51_{\pm 1.64}$	$70.37_{\pm 0.85}$	$89.02_{\pm 0.85}$	$\mathbf{80.77}_{\pm 0.46}$
DeepConf-Weighted SC + Ours	$\mathbf{81.10}_{\pm 2.38}$	$\mathbf{71.69}_{\pm 0.46}$	$\mathbf{90.24}_{\pm 0.92}$	$\mathbf{80.77}_{\pm 0.46}$
Self-Certainty-Weighted SC	$77.59_{\pm 1.54}$	$70.90_{\pm 0.82}$	$88.82_{\pm 0.78}$	$80.85_{\pm 0.41}$
Self-Certainty-Weighted SC + Ours	$\mathbf{81.25}_{\pm 1.95}$	$\mathbf{71.96}_{\pm 0.45}$	$\mathbf{90.65}_{\pm 1.22}$	$\mathbf{80.87}_{\pm 0.41}$
Self-Certainty Borda	$80.18_{\pm 1.78}$	$71.43_{\pm 0.66}$	$89.02_{\pm 0.99}$	$\mathbf{80.32}_{\pm 0.50}$
Self-Certainty Borda + Ours	$\mathbf{81.40}_{\pm 1.98}$	$\mathbf{72.22}_{\pm 0.54}$	$\mathbf{90.65}_{\pm 1.09}$	$\mathbf{80.32}_{\pm 0.50}$
DeepConf-Filter (90%)	$80.03_{\pm 1.89}$	$70.63_{\pm 0.73}$	$89.02_{\pm 0.87}$	$80.53_{\pm 0.46}$
DeepConf-Filter (90%) + Ours	$\mathbf{81.10}_{\pm 2.49}$	$\mathbf{71.69}_{\pm 0.44}$	$\mathbf{90.24}_{\pm 0.95}$	$\mathbf{80.58}_{\pm 0.45}$

Table 7: Comparison of non-thinking and reasoning modes for Qwen3-1.7B on three mathematical reasoning benchmarks ($K = 32$).

Methods	Qwen3-1.7B					
	Non-thinking mode			Reasoning mode		
	AIME 2024	AIME 2025	HMMT 2025	AIME 2024	AIME 2025	HMMT 2025
PPL Best-of-N	$12.00_{\pm 2.21}$	$11.33_{\pm 3.40}$	$5.00_{\pm 3.42}$	$36.00_{\pm 4.42}$	$34.00_{\pm 2.91}$	$23.00_{\pm 3.48}$
DeepConf Best-of-N	$12.33_{\pm 2.13}$	$14.33_{\pm 2.60}$	$8.33_{\pm 3.73}$	$62.00_{\pm 4.52}$	$42.67_{\pm 2.49}$	$28.33_{\pm 3.73}$
Self-Certainty Best-of-N	$12.67_{\pm 2.49}$	$15.00_{\pm 3.42}$	$9.67_{\pm 2.33}$	$60.67_{\pm 4.16}$	$46.67_{\pm 3.33}$	$30.67_{\pm 4.42}$
SC	$15.17_{\pm 1.55}$	$14.83_{\pm 1.57}$	$6.94_{\pm 0.57}$	$69.50_{\pm 1.98}$	$47.33_{\pm 3.67}$	$27.00_{\pm 1.00}$
SC + Ours (DeepConf)	$16.00_{\pm 2.49}$	$\mathbf{16.67}_{\pm 0.00}$	$\mathbf{7.67}_{\pm 1.53}$	$\mathbf{75.00}_{\pm 1.67}$	$\mathbf{51.33}_{\pm 2.67}$	$30.00_{\pm 2.98}$
SC + Ours (Self-Certainty)	$\mathbf{16.50}_{\pm 2.17}$	$\mathbf{16.67}_{\pm 0.00}$	$\mathbf{7.67}_{\pm 1.53}$	$74.00_{\pm 1.33}$	$50.67_{\pm 3.27}$	$\mathbf{30.33}_{\pm 2.77}$
DeepConf-Weighted SC	$\mathbf{15.67}_{\pm 2.13}$	$15.67_{\pm 1.53}$	$7.33_{\pm 1.33}$	$70.00_{\pm 1.49}$	$48.33_{\pm 4.28}$	$27.00_{\pm 1.00}$
DeepConf-Weighted SC + Ours	$\mathbf{15.67}_{\pm 2.13}$	$\mathbf{16.67}_{\pm 0.00}$	$\mathbf{7.67}_{\pm 1.53}$	$\mathbf{75.00}_{\pm 1.67}$	$\mathbf{51.00}_{\pm 3.00}$	$\mathbf{31.33}_{\pm 3.40}$
Self-Certainty-Weighted SC	$15.67_{\pm 2.13}$	$15.00_{\pm 1.67}$	$7.33_{\pm 1.33}$	$70.33_{\pm 1.80}$	$48.67_{\pm 4.27}$	$27.00_{\pm 1.00}$
Self-Certainty-Weighted SC + Ours	$\mathbf{16.00}_{\pm 2.49}$	$\mathbf{16.67}_{\pm 0.00}$	$\mathbf{7.67}_{\pm 1.53}$	$\mathbf{74.00}_{\pm 1.33}$	$\mathbf{50.67}_{\pm 3.27}$	$\mathbf{31.00}_{\pm 3.35}$

Table 8: Comparison of non-thinking and reasoning modes for Qwen3-4B on three mathematical reasoning benchmarks ($K = 32$).

Methods	Qwen3-4B					
	Non-thinking mode			Reasoning mode		
	AIME 2024	AIME 2025	HMMT 2025	AIME 2024	AIME 2025	HMMT 2025
PPL Best-of-N	$30.00_{\pm 3.94}$	$24.14_{\pm 2.82}$	$13.00_{\pm 2.77}$	$57.00_{\pm 2.33}$	$64.33_{\pm 3.96}$	$40.33_{\pm 4.33}$
DeepConf Best-of-N	$31.67_{\pm 4.77}$	$25.29_{\pm 1.63}$	$15.33_{\pm 3.06}$	$77.00_{\pm 2.33}$	$76.00_{\pm 3.89}$	$50.00_{\pm 4.94}$
Self-Certainty Best-of-N	$32.33_{\pm 3.96}$	$26.44_{\pm 3.25}$	$14.33_{\pm 2.60}$	$81.67_{\pm 2.24}$	$71.67_{\pm 3.07}$	$51.67_{\pm 4.01}$
SC	$39.56_{\pm 4.24}$	$26.25_{\pm 1.51}$	$15.67_{\pm 1.53}$	$80.00_{\pm 0.00}$	$76.33_{\pm 0.67}$	$51.50_{\pm 1.57}$
SC + Ours (DeepConf)	$\mathbf{41.83}_{\pm 3.69}$	$\mathbf{28.74}_{\pm 1.63}$	$\mathbf{16.67}_{\pm 0.00}$	$\mathbf{83.00}_{\pm 1.00}$	$76.33_{\pm 1.80}$	$54.00_{\pm 2.91}$
SC + Ours (Self-Certainty)	$41.33_{\pm 3.40}$	$\mathbf{28.74}_{\pm 1.63}$	$\mathbf{16.67}_{\pm 0.00}$	$82.00_{\pm 1.63}$	$\mathbf{78.67}_{\pm 1.63}$	$\mathbf{55.67}_{\pm 1.53}$
DeepConf-Weighted SC	$41.00_{\pm 4.23}$	$26.44_{\pm 1.63}$	$15.67_{\pm 1.53}$	$80.00_{\pm 0.00}$	$\mathbf{76.33}_{\pm 1.00}$	$51.33_{\pm 1.63}$
DeepConf-Weighted SC + Ours	$\mathbf{42.33}_{\pm 3.96}$	$\mathbf{27.59}_{\pm 2.82}$	$\mathbf{16.67}_{\pm 0.00}$	$\mathbf{83.33}_{\pm 1.49}$	$\mathbf{76.33}_{\pm 2.33}$	$\mathbf{54.00}_{\pm 2.91}$
Self-Certainty-Weighted SC	$40.33_{\pm 4.07}$	$26.44_{\pm 1.63}$	$15.67_{\pm 1.53}$	$80.00_{\pm 0.00}$	$76.33_{\pm 1.00}$	$52.67_{\pm 2.00}$
Self-Certainty-Weighted SC + Ours	$\mathbf{41.67}_{\pm 3.07}$	$\mathbf{27.59}_{\pm 2.82}$	$\mathbf{16.67}_{\pm 0.00}$	$\mathbf{82.00}_{\pm 1.63}$	$\mathbf{79.33}_{\pm 1.33}$	$\mathbf{55.33}_{\pm 1.63}$

Table 9: Accuracy comparison under the sampling budgets of $\frac{K}{2}$ and K, using product-aggregated rewards from an external process reward model Qwen2.5-Math-PRM-7B (Lightman et al., 2024; Zhang et al., 2025). **+ Ours** denotes the PGLA-enhanced variant of each voting baseline.

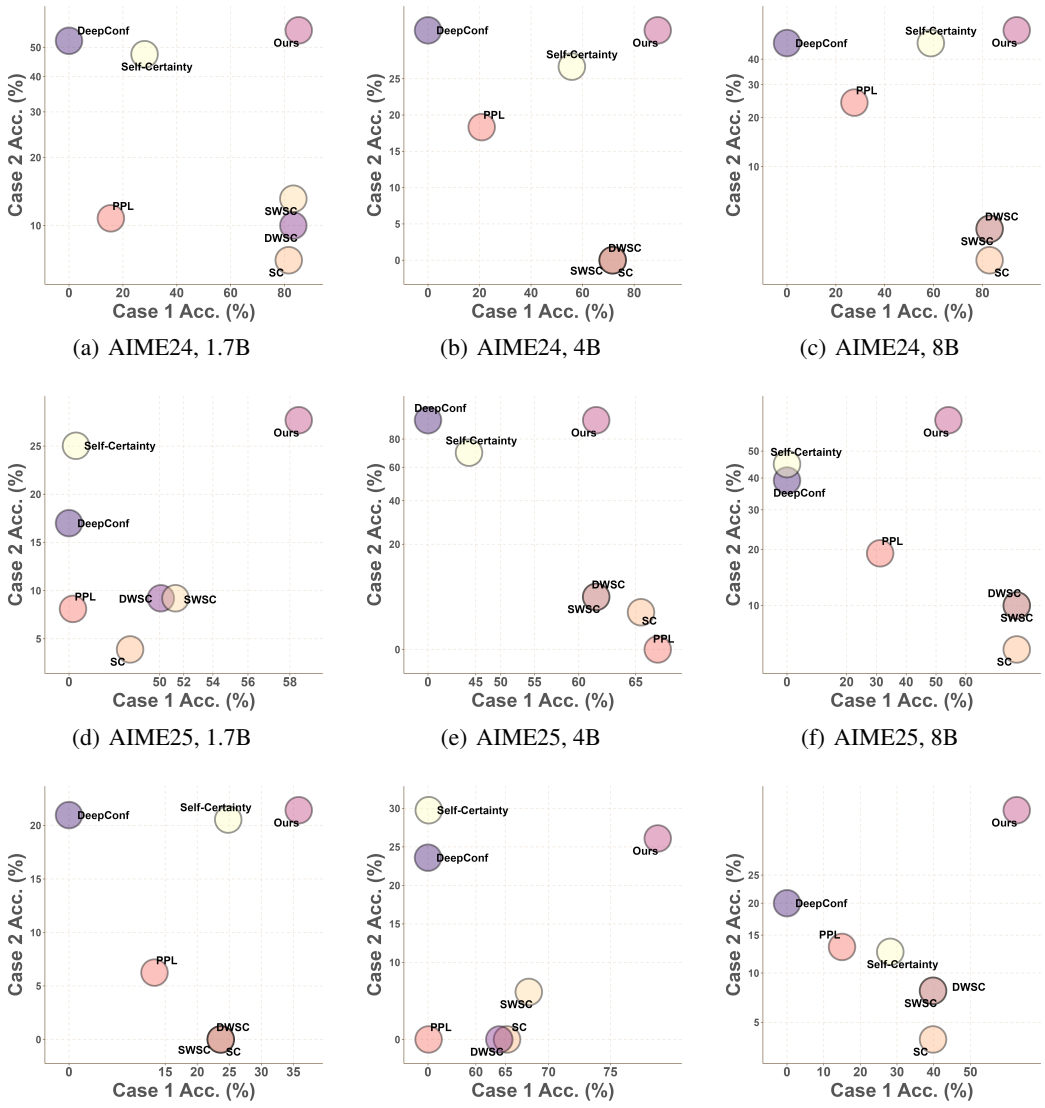
Method	MATH		MathOdyssey		AIME		Average	
	$K/2\uparrow$	$K\uparrow$	$K/2\uparrow$	$K\uparrow$	$K/2\uparrow$	$K\uparrow$	$K/2\uparrow$	$K\uparrow$
InternLM2-Math-Plus-1.8B								
PRM Best-of-N	$46.50_{\pm 0.33}$	$48.48_{\pm 0.21}$	$24.75_{\pm 1.44}$	$25.47_{\pm 1.24}$	$4.43_{\pm 0.24}$	$4.93_{\pm 0.34}$	25.23	26.30
PRM-Weighted SC	$45.64_{\pm 0.26}$	$47.39_{\pm 0.23}$	$23.24_{\pm 1.04}$	$24.91_{\pm 1.07}$	$4.63_{\pm 0.25}$	$5.02_{\pm 0.23}$	24.50	25.77
+ Ours	$\mathbf{46.88}_{\pm 0.29}$	$\mathbf{49.09}_{\pm 0.20}$	$\mathbf{26.99}_{\pm 1.16}$	$\mathbf{28.95}_{\pm 1.26}$	$\mathbf{4.73}_{\pm 0.33}$	$\mathbf{5.17}_{\pm 0.27}$	**26.20**	**27.73**
PRM-RPC	$45.77_{\pm 0.25}$	$47.61_{\pm 0.23}$	$23.60_{\pm 1.19}$	$25.32_{\pm 1.04}$	$4.66_{\pm 0.24}$	$5.07_{\pm 0.21}$	24.68	26.00
+ Ours	$\mathbf{46.88}_{\pm 0.28}$	$\mathbf{49.18}_{\pm 0.17}$	$\mathbf{27.02}_{\pm 1.28}$	$\mathbf{29.05}_{\pm 1.30}$	$\mathbf{4.73}_{\pm 0.32}$	$\mathbf{5.19}_{\pm 0.26}$	**26.21**	**27.80**
InternLM2-Math-Plus-7B								
PRM Best-of-N	$58.22_{\pm 0.34}$	$59.34_{\pm 0.32}$	$32.85_{\pm 1.78}$	$33.07_{\pm 0.99}$	$15.10_{\pm 0.50}$	$15.96_{\pm 0.36}$	35.39	36.12
PRM-Weighted SC	$57.56_{\pm 0.31}$	$58.67_{\pm 0.10}$	$33.57_{\pm 1.13}$	$34.34_{\pm 0.58}$	$15.49_{\pm 0.36}$	$16.23_{\pm 0.48}$	35.54	36.42
+ Ours	$\mathbf{58.86}_{\pm 0.29}$	$\mathbf{60.40}_{\pm 0.24}$	$\mathbf{37.63}_{\pm 1.18}$	$\mathbf{38.79}_{\pm 0.62}$	$\mathbf{15.55}_{\pm 0.42}$	$\mathbf{16.57}_{\pm 0.35}$	**37.35**	**38.59**
PRM-RPC	$57.74_{\pm 0.28}$	$58.90_{\pm 0.13}$	$34.83_{\pm 1.18}$	$35.48_{\pm 0.40}$	$\mathbf{15.57}_{\pm 0.42}$	$16.31_{\pm 0.44}$	36.05	36.90
+ Ours	$\mathbf{58.88}_{\pm 0.24}$	$\mathbf{60.41}_{\pm 0.25}$	$\mathbf{37.63}_{\pm 1.24}$	$\mathbf{38.87}_{\pm 0.60}$	$15.53_{\pm 0.37}$	$\mathbf{16.50}_{\pm 0.28}$	**37.35**	**38.59**
Deepseek-Math-RL-7B								
PRM Best-of-N	$61.27_{\pm 0.24}$	$62.79_{\pm 0.16}$	$34.34_{\pm 1.27}$	$34.31_{\pm 0.66}$	$15.93_{\pm 0.23}$	$17.14_{\pm 0.40}$	37.18	38.08
PRM-Weighted SC	$60.76_{\pm 0.25}$	$62.04_{\pm 0.20}$	$39.69_{\pm 0.59}$	$40.28_{\pm 0.53}$	$15.77_{\pm 0.27}$	$16.93_{\pm 0.43}$	38.74	39.75
+ Ours	$\mathbf{61.17}_{\pm 0.24}$	$\mathbf{62.41}_{\pm 0.20}$	$\mathbf{40.18}_{\pm 0.64}$	$\mathbf{40.69}_{\pm 0.53}$	$\mathbf{16.13}_{\pm 0.48}$	$\mathbf{17.49}_{\pm 0.57}$	**39.16**	**40.20**
PRM-RPC	$60.91_{\pm 0.27}$	$62.26_{\pm 0.22}$	$\mathbf{40.05}_{\pm 0.83}$	$40.67_{\pm 0.51}$	$15.85_{\pm 0.32}$	$17.05_{\pm 0.36}$	38.94	39.99
+ Ours	$\mathbf{61.19}_{\pm 0.23}$	$\mathbf{62.55}_{\pm 0.19}$	$39.85_{\pm 0.75}$	$\mathbf{40.69}_{\pm 0.42}$	$\mathbf{16.14}_{\pm 0.51}$	$\mathbf{17.52}_{\pm 0.66}$	**39.06**	**40.26**

G.3 Performance with the External Reward Setting

Tab. 9 compares the methods using product-aggregated rewards from an external process reward model. We observe that our method improves PRM-Weighted SC and PRM-RPC in most settings. To be specific, at the full budget, PGLA raises their average accuracy from 33.98 to 35.51 and from 34.30 to 35.55, respectively. This further demonstrates that our method can effectively use externally provided reward signals.

G.4 TRADEOFF BETWEEN CASE I AND CASE II

Fig. 10 illustrates complementary failure modes: Best-of-N often fails in Case I, while voting-based methods tend to fail in Case II. By combining voting support with pairwise confidence comparisons, PGLA generally achieves a better balance between the two cases.



(a) AIME24, 1.7B (b) AIME24, 4B (c) AIME24, 8B

(d) AIME25, 1.7B (e) AIME25, 4B (f) AIME25, 8B

(g) HMMT25, 1.7B (h) HMMT25, 4B (i) HMMT25, 8B

Figure 10: Tradeoff between Case I and Case II on AIME24, AIME25, and HMMT25 using Qwen3-1.7B, Qwen3-4B, and Qwen3-8B. Ours denotes PGLA. DeepConf, PPL, and Self-Certainty denote DeepConf Best-of-N, PPL Best-of-N, and Self-Certainty Best-of-N, respectively. SWSC and DWSC denote Self-Certainty-Weighted SC and DeepConf-Weighted SC, respectively.

G.5 Effect of Ranking Quality by Dataset

We study how the ranking quality of path scores affects PGLA. Following Sec. 2.1, let a_i and p_i be the answer and confidence of path i. Here p_i is the product-aggregated reward from Qwen2.5-Math-PRM-7B, floored at 10^{-8}. We keep the sampled paths, answers, and PRM-Weighted SC support $s_a^{\mathcal{V}} = \sum_{i=1}^{K} p_i \mathbf{1}(a_i = a)$ fixed, where $\mathcal{V}$ denotes PRM-Weighted SC. Only the scores used for pairwise comparisons are perturbed; the opponent weights and unit-strength PGLA update remain unchanged.

Controlled perturbation. The retention parameter $\lambda \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ controls how many path-score assignments are preserved. For each query, define

$$m_\lambda = \mathrm{round}((1-\lambda)K), \qquad \tilde{p}_i^{(\lambda)} = p_{\sigma_\lambda(i)}, \tag{29}$$

where round denotes nearest-integer rounding and σ_λ is a random cyclic permutation on m_λ uniformly selected distinct path indices, extended by the identity on the remaining indices. If $m_\lambda < 2$, σ_λ is the identity everywhere. Thus, $\lambda = 1$ preserves the original ranking, while $\lambda = 0$ permutes all score assignments within the query. This procedure preserves the score multiset and uses no correctness labels. Because of rounding and score ties, λ is a nominal retention fraction, rather than the exact fraction of unchanged score values.

Ranking quality. We measure ranking quality by the empirical area under the ROC curve (AUC) for distinguishing correct from incorrect paths. Let q index evaluated queries, and let $\tilde{p}_{q,i}^{(\lambda)}$ be the perturbed confidence of path i for query q. Define $\mathcal{I}_q^+$ and $\mathcal{I}_q^-$ as the index sets of paths whose answers are correct and incorrect, respectively. For a query containing both kinds of paths, its AUC is

$$\widehat{\rho}_q(\lambda) = \frac{1}{|\mathcal{I}_q^+|\,|\mathcal{I}_q^-|} \sum_{i \in \mathcal{I}_q^+} \sum_{j \in \mathcal{I}_q^-} \left[\mathbf{1}\left(\tilde{p}_{q,i}^{(\lambda)} > \tilde{p}_{q,j}^{(\lambda)}\right) + \frac{1}{2}\mathbf{1}\left(\tilde{p}_{q,i}^{(\lambda)} = \tilde{p}_{q,j}^{(\lambda)}\right) \right]. \tag{30}$$

This uses the same tie convention as Eq. (6): a correct path outranking an incorrect path contributes one, and a tie contributes one half. Let $\mathcal{Q}_{\mathrm{mix}}$ contain the queries for which both $\mathcal{I}_q^+$ and $\mathcal{I}_q^-$ are nonempty. The dataset-level AUC is their equally weighted average,

$$\widehat{\rho}(\lambda) = \frac{1}{|\mathcal{Q}_{\mathrm{mix}}|} \sum_{q \in \mathcal{Q}_{\mathrm{mix}}} \widehat{\rho}_q(\lambda). \tag{31}$$

Queries containing only correct or only incorrect paths are excluded from this AUC but remain in the accuracy evaluation. The tables abbreviate $\widehat{\rho}(\lambda)$ as $\widehat{\rho}$; the model-level curves in Fig. 7 further average the dataset metrics equally.

The gains from PGLA generally grow as ranking quality improves across datasets. In Fig. 7, increasing $\widehat{\rho}$ from 0.496 to 0.729 on InternLM2-Math-Plus-7B raises the absolute accuracy gain from -0.54% to $+2.17\%$. A similar trend appears on DeepSeek-Math-RL-7B in Tabs. 11, 13 and 15, supporting the benefit of more reliable pairwise confidence comparisons.

G.6 Temperature Sensitivity

To further examine robustness under more diverse sampled reasoning paths, we conduct additional experiments on MathOdyssey and AIME using **InternLM2-Math-Plus-1.8B**, **InternLM2-Math-Plus-7B**, and **DeepSeek-Math-RL-7B**. Tab. 16 compares the methods under higher decoding temperatures with $T = 1.1$ and $T = 1.3$. Our method remains competitive and generally outperforms the comparison methods, indicating that the proposed aggregation mechanism is still effective when the sampled proposal pool becomes more diverse.

G.7 Efficiency Comparison

Tabs. 17 and 18 compare the sampling budgets of SC, SC + Ours, RPC, and RPC + Ours on MATH, MathOdyssey, and AIME, capped at 64 paths for MATH and 128 otherwise. For each model–dataset setting, the target is the best mean accuracy attained by SC on the evaluated grid. We then find the smallest grid budget at which each other method matches or exceeds that target. The reported savings describe empirical accuracy–budget tradeoffs.

Table 10: Controlled ranking on MATH with InternLM2-Math-Plus-7B ($K = 64$).

Base Acc.: $58.67_{\pm 0.10}$

λ	$\widehat{\rho}$	PGLA Acc.	ΔAcc
0.0	$0.496_{\pm 0.003}$	$58.23_{\pm 0.09}$	$-0.45_{\pm 0.12}$
0.2	$0.557_{\pm 0.002}$	$58.69_{\pm 0.16}$	$+0.01_{\pm 0.16}$
0.4	$0.617_{\pm 0.002}$	$59.23_{\pm 0.20}$	$+0.56_{\pm 0.19}$
0.6	$0.671_{\pm 0.002}$	$59.61_{\pm 0.07}$	$+0.94_{\pm 0.15}$
0.8	$0.732_{\pm 0.002}$	$60.05_{\pm 0.20}$	$+1.37_{\pm 0.17}$
1.0	$0.791_{\pm 0.002}$	$\mathbf{60.40}_{\pm 0.24}$	$+\mathbf{1.73}_{\pm 0.18}$

Table 11: Controlled ranking on MATH with DeepSeek-Math-RL-7B ($K = 64$).

Base Acc.: $62.04_{\pm 0.20}$

λ	$\widehat{\rho}$	PGLA Acc.	ΔAcc
0.0	$0.495_{\pm 0.003}$	$61.53_{\pm 0.18}$	$-0.51_{\pm 0.23}$
0.2	$0.561_{\pm 0.002}$	$61.83_{\pm 0.24}$	$-0.22_{\pm 0.13}$
0.4	$0.627_{\pm 0.003}$	$62.03_{\pm 0.23}$	$-0.01_{\pm 0.07}$
0.6	$0.688_{\pm 0.003}$	$62.15_{\pm 0.16}$	$+0.11_{\pm 0.20}$
0.8	$0.754_{\pm 0.001}$	$62.32_{\pm 0.20}$	$+0.28_{\pm 0.23}$
1.0	$0.821_{\pm 0.001}$	$\mathbf{62.41}_{\pm 0.20}$	$+\mathbf{0.37}_{\pm 0.19}$

Table 12: Controlled ranking on MathOdyssey with InternLM2-Math-Plus-7B ($K = 128$).

Base Acc.: $34.34_{\pm 0.58}$

λ	$\widehat{\rho}$	PGLA Acc.	ΔAcc
0.0	$0.499_{\pm 0.008}$	$33.60_{\pm 0.85}$	$-0.75_{\pm 0.89}$
0.2	$0.538_{\pm 0.008}$	$35.35_{\pm 0.55}$	$+1.00_{\pm 0.72}$
0.4	$0.577_{\pm 0.010}$	$36.25_{\pm 0.46}$	$+1.90_{\pm 0.81}$
0.6	$0.615_{\pm 0.006}$	$37.56_{\pm 0.60}$	$+3.21_{\pm 0.45}$
0.8	$0.652_{\pm 0.009}$	$38.17_{\pm 0.49}$	$+3.83_{\pm 0.82}$
1.0	$0.692_{\pm 0.008}$	$\mathbf{38.79}_{\pm 0.62}$	$+\mathbf{4.45}_{\pm 0.78}$

Table 13: Controlled ranking on MathOdyssey with DeepSeek-Math-RL-7B ($K = 128$).

Base Acc.: $40.28_{\pm 0.53}$

λ	$\widehat{\rho}$	PGLA Acc.	ΔAcc
0.0	$0.497_{\pm 0.009}$	$40.00_{\pm 0.53}$	$-0.28_{\pm 0.52}$
0.2	$0.539_{\pm 0.010}$	$40.41_{\pm 0.47}$	$+0.13_{\pm 0.35}$
0.4	$0.564_{\pm 0.008}$	$40.33_{\pm 0.33}$	$+0.05_{\pm 0.54}$
0.6	$0.603_{\pm 0.007}$	$40.44_{\pm 0.75}$	$+0.15_{\pm 0.88}$
0.8	$0.634_{\pm 0.006}$	$40.57_{\pm 0.69}$	$+0.28_{\pm 0.62}$
1.0	$0.670_{\pm 0.005}$	$\mathbf{40.69}_{\pm 0.53}$	$+\mathbf{0.41}_{\pm 0.52}$

Table 14: Controlled ranking on AIME with InternLM2-Math-Plus-7B ($K = 128$).

Base Acc.: $16.23_{\pm 0.48}$

λ	$\widehat{\rho}$	PGLA Acc.	ΔAcc
0.0	$0.493_{\pm 0.011}$	$15.81_{\pm 0.32}$	$-0.42_{\pm 0.32}$
0.2	$0.542_{\pm 0.007}$	$16.08_{\pm 0.42}$	$-0.15_{\pm 0.38}$
0.4	$0.579_{\pm 0.011}$	$16.14_{\pm 0.53}$	$-0.09_{\pm 0.44}$
0.6	$0.623_{\pm 0.010}$	$16.36_{\pm 0.55}$	$+0.13_{\pm 0.51}$
0.8	$0.659_{\pm 0.010}$	$16.50_{\pm 0.59}$	$+0.27_{\pm 0.38}$
1.0	$0.703_{\pm 0.008}$	$\mathbf{16.57}_{\pm 0.35}$	$+\mathbf{0.34}_{\pm 0.35}$

Table 15: Controlled ranking on AIME with DeepSeek-Math-RL-7B ($K = 128$).

Base Acc.: $16.93_{\pm 0.43}$

λ	$\widehat{\rho}$	PGLA Acc.	ΔAcc
0.0	$0.497_{\pm 0.008}$	$16.53_{\pm 0.42}$	$-0.41_{\pm 0.37}$
0.2	$0.547_{\pm 0.013}$	$16.68_{\pm 0.34}$	$-0.26_{\pm 0.41}$
0.4	$0.590_{\pm 0.008}$	$16.78_{\pm 0.57}$	$-0.15_{\pm 0.38}$
0.6	$0.631_{\pm 0.007}$	$16.87_{\pm 0.60}$	$-0.06_{\pm 0.52}$
0.8	$0.675_{\pm 0.010}$	$17.18_{\pm 0.61}$	$+0.25_{\pm 0.39}$
1.0	$0.720_{\pm 0.007}$	$\mathbf{17.49}_{\pm 0.57}$	$+\mathbf{0.56}_{\pm 0.42}$

Table 16: Temperature sensitivity under the same internal-confidence protocol as Tab. 2. We vary only the decoding temperature; **+ Ours** denotes the PGLA-enhanced variant of each voting baseline.

Method	MathOdyssey				AIME			
	$T = 1.1$		$T = 1.3$		$T = 1.1$		$T = 1.3$	
	Acc. ↑	ECE ↓	Acc. ↑	ECE ↓	Acc. ↑	ECE ↓	Acc. ↑	ECE ↓
InternLM2-Math-Plus-1.8B								
PPL	$16.43_{\pm 0.91}$	$75.99_{\pm 0.89}$	$16.38_{\pm 1.42}$	$71.72_{\pm 1.43}$	$1.97_{\pm 0.29}$	$90.29_{\pm 0.28}$	$2.08_{\pm 0.27}$	$84.85_{\pm 0.27}$
SC	$13.75_{\pm 0.51}$	$18.02_{\pm 0.57}$	$14.95_{\pm 0.87}$	$5.15_{\pm 1.12}$	$2.12_{\pm 0.10}$	$16.52_{\pm 0.13}$	$1.26_{\pm 0.08}$	$8.80_{\pm 0.12}$
SC + Ours	$\mathbf{18.59_{\pm 0.78}}$	$\mathbf{12.75_{\pm 0.75}}$	$\mathbf{19.23_{\pm 0.76}}$	$\mathbf{6.86_{\pm 1.04}}$	$\mathbf{3.11_{\pm 0.17}}$	$\mathbf{14.78_{\pm 0.19}}$	$\mathbf{1.82_{\pm 0.17}}$	$\mathbf{10.64_{\pm 0.16}}$
Weighted SC	$13.91_{\pm 0.54}$	$17.76_{\pm 0.62}$	$15.68_{\pm 0.79}$	$9.41_{\pm 0.86}$	$2.14_{\pm 0.11}$	$16.28_{\pm 0.16}$	$1.58_{\pm 0.07}$	$12.65_{\pm 0.12}$
Weighted SC + Ours	$\mathbf{18.84_{\pm 0.76}}$	$\mathbf{12.25_{\pm 0.79}}$	$\mathbf{21.31_{\pm 1.15}}$	$\mathbf{6.95_{\pm 1.10}}$	$\mathbf{3.29_{\pm 0.24}}$	$\mathbf{14.40_{\pm 0.29}}$	$\mathbf{2.22_{\pm 0.17}}$	$\mathbf{13.80_{\pm 0.26}}$
Adaptive SC	$13.81_{\pm 0.63}$	$22.30_{\pm 0.96}$	$15.05_{\pm 0.77}$	$6.56_{\pm 0.85}$	$2.12_{\pm 0.10}$	$18.93_{\pm 0.27}$	$1.26_{\pm 0.08}$	$9.65_{\pm 0.12}$
Adaptive SC + Ours	$\mathbf{17.60_{\pm 1.01}}$	$\mathbf{17.05_{\pm 1.31}}$	$\mathbf{18.46_{\pm 0.90}}$	$\mathbf{8.02_{\pm 1.09}}$	$\mathbf{3.10_{\pm 0.19}}$	$\mathbf{16.67_{\pm 0.43}}$	$\mathbf{1.81_{\pm 0.18}}$	$\mathbf{11.62_{\pm 0.26}}$
RPC	$15.53_{\pm 0.54}$	$16.10_{\pm 0.82}$	$16.07_{\pm 0.94}$	$11.41_{\pm 0.95}$	$2.80_{\pm 0.18}$	$15.64_{\pm 0.15}$	$1.64_{\pm 0.14}$	$14.42_{\pm 0.21}$
RPC + Ours	$\mathbf{20.41_{\pm 0.69}}$	$\mathbf{10.45_{\pm 0.71}}$	$\mathbf{21.83_{\pm 0.96}}$	$\mathbf{7.69_{\pm 1.24}}$	$\mathbf{3.63_{\pm 0.25}}$	$\mathbf{13.78_{\pm 0.26}}$	$\mathbf{2.33_{\pm 0.17}}$	$\mathbf{14.83_{\pm 0.28}}$
InternLM2-Math-Plus-7B								
PPL	$28.61_{\pm 1.33}$	$65.31_{\pm 1.33}$	$26.38_{\pm 1.41}$	$65.61_{\pm 1.41}$	$6.02_{\pm 0.34}$	$87.67_{\pm 0.35}$	$6.50_{\pm 0.41}$	$84.66_{\pm 0.40}$
SC	$27.89_{\pm 0.43}$	$11.31_{\pm 0.52}$	$27.51_{\pm 0.69}$	$9.38_{\pm 1.14}$	$8.73_{\pm 0.24}$	$13.71_{\pm 0.25}$	$8.02_{\pm 0.20}$	$12.02_{\pm 0.24}$
SC + Ours	$\mathbf{35.84_{\pm 0.84}}$	$\mathbf{9.07_{\pm 0.90}}$	$\mathbf{36.35_{\pm 0.50}}$	$\mathbf{8.82_{\pm 0.75}}$	$\mathbf{9.61_{\pm 0.15}}$	$\mathbf{13.08_{\pm 0.15}}$	$\mathbf{9.09_{\pm 0.26}}$	$\mathbf{11.23_{\pm 0.31}}$
Weighted SC	$28.38_{\pm 0.33}$	$10.77_{\pm 0.45}$	$28.71_{\pm 0.60}$	$8.11_{\pm 0.85}$	$8.89_{\pm 0.22}$	$13.53_{\pm 0.23}$	$8.40_{\pm 0.25}$	$11.52_{\pm 0.28}$
Weighted SC + Ours	$\mathbf{36.07_{\pm 0.86}}$	$\mathbf{9.20_{\pm 0.84}}$	$\mathbf{36.79_{\pm 0.53}}$	$\mathbf{8.85_{\pm 0.94}}$	$\mathbf{9.75_{\pm 0.23}}$	$\mathbf{12.83_{\pm 0.24}}$	$\mathbf{9.32_{\pm 0.28}}$	$\mathbf{10.90_{\pm 0.29}}$
Adaptive SC	$27.92_{\pm 0.61}$	$16.96_{\pm 0.77}$	$27.75_{\pm 0.65}$	$13.27_{\pm 0.80}$	$8.71_{\pm 0.25}$	$16.61_{\pm 0.46}$	$8.00_{\pm 0.16}$	$14.51_{\pm 0.26}$
Adaptive SC + Ours	$\mathbf{34.00_{\pm 0.99}}$	$\mathbf{12.35_{\pm 0.82}}$	$\mathbf{34.24_{\pm 0.83}}$	$\mathbf{10.02_{\pm 0.98}}$	$\mathbf{9.43_{\pm 0.32}}$	$\mathbf{15.62_{\pm 0.48}}$	$\mathbf{9.11_{\pm 0.21}}$	$\mathbf{13.11_{\pm 0.41}}$
RPC	$32.93_{\pm 0.64}$	$8.05_{\pm 0.92}$	$33.01_{\pm 0.50}$	$5.97_{\pm 0.96}$	$9.31_{\pm 0.28}$	$13.52_{\pm 0.25}$	$8.83_{\pm 0.23}$	$11.40_{\pm 0.28}$
RPC + Ours	$\mathbf{36.99_{\pm 1.07}}$	$\mathbf{8.38_{\pm 1.26}}$	$\mathbf{38.02_{\pm 0.68}}$	$\mathbf{7.97_{\pm 0.91}}$	$\mathbf{10.13_{\pm 0.38}}$	$\mathbf{12.17_{\pm 0.40}}$	$\mathbf{9.30_{\pm 0.22}}$	$\mathbf{10.88_{\pm 0.26}}$
Deepseek-Math-RL-7B								
PPL	$24.78_{\pm 1.15}$	$70.85_{\pm 1.18}$	$25.76_{\pm 1.26}$	$68.39_{\pm 1.29}$	$3.94_{\pm 0.37}$	$92.17_{\pm 0.37}$	$4.39_{\pm 0.33}$	$90.33_{\pm 0.34}$
SC	$36.76_{\pm 0.39}$	$9.46_{\pm 0.81}$	$37.75_{\pm 0.25}$	$8.78_{\pm 1.16}$	$8.95_{\pm 0.26}$	$11.95_{\pm 0.34}$	$9.39_{\pm 0.24}$	$10.23_{\pm 0.26}$
SC + Ours	$\mathbf{37.22_{\pm 0.55}}$	$\mathbf{11.03_{\pm 1.11}}$	$\mathbf{38.55_{\pm 0.61}}$	$\mathbf{9.01_{\pm 0.70}}$	$\mathbf{9.39_{\pm 0.30}}$	$\mathbf{11.01_{\pm 0.34}}$	$\mathbf{10.15_{\pm 0.33}}$	$\mathbf{9.57_{\pm 0.36}}$
Weighted SC	$36.84_{\pm 0.45}$	$9.40_{\pm 0.81}$	$37.76_{\pm 0.39}$	$8.91_{\pm 0.86}$	$9.04_{\pm 0.29}$	$11.86_{\pm 0.38}$	$9.42_{\pm 0.24}$	$10.24_{\pm 0.24}$
Weighted SC + Ours	$\mathbf{37.17_{\pm 0.60}}$	$\mathbf{11.22_{\pm 1.23}}$	$\mathbf{38.56_{\pm 0.63}}$	$\mathbf{9.14_{\pm 0.88}}$	$\mathbf{9.41_{\pm 0.33}}$	$\mathbf{10.93_{\pm 0.37}}$	$\mathbf{10.19_{\pm 0.31}}$	$\mathbf{9.46_{\pm 0.34}}$
Adaptive SC	$36.62_{\pm 0.35}$	$12.71_{\pm 1.30}$	$37.63_{\pm 0.42}$	$11.23_{\pm 1.22}$	$8.93_{\pm 0.29}$	$14.23_{\pm 0.42}$	$9.40_{\pm 0.24}$	$12.72_{\pm 0.36}$
Adaptive SC + Ours	$\mathbf{35.94_{\pm 0.38}}$	$\mathbf{12.33_{\pm 0.84}}$	$\mathbf{37.03_{\pm 0.96}}$	$\mathbf{11.30_{\pm 1.89}}$	$\mathbf{9.26_{\pm 0.36}}$	$\mathbf{13.09_{\pm 0.55}}$	$\mathbf{9.89_{\pm 0.34}}$	$\mathbf{11.95_{\pm 0.33}}$
RPC	$36.92_{\pm 0.40}$	$10.19_{\pm 0.60}$	$38.25_{\pm 0.47}$	$8.96_{\pm 0.81}$	$9.09_{\pm 0.34}$	$11.81_{\pm 0.40}$	$10.04_{\pm 0.39}$	$10.35_{\pm 0.44}$
RPC + Ours	$\mathbf{37.22_{\pm 0.43}}$	$\mathbf{11.63_{\pm 1.45}}$	$\mathbf{38.95_{\pm 0.82}}$	$\mathbf{8.82_{\pm 1.06}}$	$\mathbf{9.35_{\pm 0.29}}$	$\mathbf{10.47_{\pm 0.31}}$	$\mathbf{10.32_{\pm 0.26}}$	$\mathbf{8.82_{\pm 0.30}}$

Table 17: Post-hoc sampling-budget comparison for **InternLM2-Math-Plus-1.8B**. Best of SC gives its highest grid mean and smallest attaining K. Alternatives use the minimum K matching or exceeding this unrounded target. Δ gives absolute accuracy change (%) and sample-count change (%) relative to SC.

Method	**MATH**		**MathOdyssey**		**AIME**	
	Acc.	K	Acc.	K	Acc.	K
Best of SC	36.48 ± 0.14	64	14.91 ± 0.95	48	2.67 ± 0.21	112
SC + Ours	37.76 ± 0.34	16	16.53 ± 0.92	8	2.76 ± 0.34	32
Δ vs. SC	+1.27	**-75.0%**	+1.62	**-83.3%**	+0.09	**-71.4%**
RPC	37.09 ± 0.20	32	15.37 ± 1.45	8	2.71 ± 0.39	32
Δ vs. SC	+0.61	**-50.0%**	+0.46	**-83.3%**	+0.04	**-71.4%**
RPC + Ours	37.67 ± 0.36	16	16.20 ± 0.82	8	2.86 ± 0.36	40
Δ vs. SC	+1.18	**-75.0%**	+1.28	**-83.3%**	+0.19	**-64.3%**

Table 18: Post-hoc sampling-budget comparison for **DeepSeek-Math-RL-7B**. Best of SC gives its highest grid mean and smallest attaining K. Alternatives use the minimum K matching or exceeding this unrounded target. Δ gives absolute accuracy change (%) and sample-count change (%) relative to SC.

Method	**MATH**		**MathOdyssey**		**AIME**	
	Acc.	K	Acc.	K	Acc.	K
Best of SC	53.33 ± 0.08	64	36.57 ± 0.56	128	9.56 ± 0.24	96
SC + Ours	53.52 ± 0.10	64	36.76 ± 1.00	40	9.80 ± 0.37	64
Δ vs. SC	+0.19	0.0%	+0.19	**-68.8%**	+0.23	**-33.3%**
RPC	53.36 ± 0.11	64	36.86 ± 0.57	112	-	-
Δ vs. SC	+0.03	0.0%	+0.29	**-12.5%**	-	-
RPC + Ours	53.52 ± 0.11	64	36.58 ± 0.95	40	9.70 ± 0.30	64
Δ vs. SC	+0.19	0.0%	+0.01	**-68.8%**	+0.14	**-33.3%**

Tabs. 17 and 18 show that PGLA-enhanced SC and RPC generally reach SC's best observed accuracy with fewer sampled paths. The savings are larger on InternLM2-Math-Plus-1.8B, while DeepSeek-Math-RL-7B also benefits on MathOdyssey and AIME. These results suggest that pairwise evidence improves the tradeoff between accuracy and sampling budget.

The computational cost of the additional aggregation step is analyzed separately in App. G.12.

G.8 Complete Scaling Results with Varying K

Tabs. 19 and 20 report the complete current-method curves for InternLM2-Math-Plus-1.8B and InternLM2-Math-Plus-7B. We use $K \in \{1, 2, 4, 8, 16, 32, 40, 48, 64, 80, 96, 112, 128\}$, capped at 64 for MATH and 128 for MathOdyssey and AIME by the evaluation protocol. All methods use the same candidate subsets at each budgets. Here **+ Ours** denotes PGLA with the voting-weighted comparative advantage s_u^{Pair} in Eq. (7).

Tabs. 19 and 20 show that PGLA generally improves SC and RPC across sampling budgets. The gains become clearer at moderate and large K, especially on MATH and MathOdyssey, indicating that pairwise evidence helps make better use of additional reasoning paths.

G.9 Online Sampling Efficiency

The full-budget results establish the aggregation benefit of PGLA. We next examine whether this benefit remains when the sampling budget is determined independently for each query. Following App. D, we apply the same stopping rule and hyperparameters to SC and SC+PGLA on identical nested path prefixes, with each rule tracking its own unique winner. Both methods update their

Table 19: Accuracy across sampling budgets for **InternLM2-Math-Plus-1.8B** on MATH, MathOdyssey, and AIME.

			MATH		
K	**PPL**	**SC**	**SC + Ours**	**RPC**	**RPC + Ours**
1	**26.94** ± 0.49	**26.94** ± 0.49	**26.94** ± 0.49	**26.94** ± 0.49	**26.94** ± 0.49
2	**29.62** ± 0.47	27.13 ± 0.44	**29.62** ± 0.47	**29.62** ± 0.47	**29.62** ± 0.47
4	31.32 ± 0.44	31.01 ± 0.36	31.32 ± 0.42	**32.36** ± 0.41	31.30 ± 0.42
8	32.53 ± 0.42	33.93 ± 0.26	**34.81** ± 0.46	34.78 ± 0.32	34.34 ± 0.47
16	33.15 ± 0.42	35.49 ± 0.28	**37.76** ± 0.34	36.08 ± 0.30	37.67 ± 0.36
32	33.58 ± 0.22	36.27 ± 0.22	39.04 ± 0.28	37.09 ± 0.20	**39.13** ± 0.38
40	33.43 ± 0.22	36.34 ± 0.12	39.35 ± 0.25	37.40 ± 0.20	**39.50** ± 0.33
48	33.39 ± 0.28	36.37 ± 0.19	39.53 ± 0.24	37.54 ± 0.21	**39.99** ± 0.28
64	33.23 ± 0.25	36.48 ± 0.14	39.77 ± 0.17	37.87 ± 0.14	**40.29** ± 0.27
			MathOdyssey		
K	**PPL**	**SC**	**SC + Ours**	**RPC**	**RPC + Ours**
1	**11.41** ± 1.23	**11.41** ± 1.23	**11.41** ± 1.23	**11.41** ± 1.23	**11.41** ± 1.23
2	**13.26** ± 1.87	11.47 ± 1.04	**13.26** ± 1.87	**13.26** ± 1.87	**13.26** ± 1.87
4	13.91 ± 1.52	13.21 ± 1.54	13.98 ± 1.53	13.91 ± 1.57	**14.04** ± 1.53
8	14.76 ± 1.26	14.43 ± 0.93	**16.53** ± 0.92	15.37 ± 1.45	16.20 ± 0.82
16	15.35 ± 1.29	14.84 ± 0.65	**17.71** ± 1.08	15.30 ± 0.89	17.51 ± 1.11
32	15.94 ± 1.69	14.52 ± 1.05	18.25 ± 0.93	15.45 ± 1.01	**18.64** ± 1.05
40	16.38 ± 0.76	14.70 ± 0.92	18.11 ± 0.56	15.68 ± 1.07	**18.59** ± 1.44
48	16.09 ± 0.64	14.91 ± 0.95	18.33 ± 0.92	15.99 ± 1.09	**19.41** ± 1.28
64	16.71 ± 1.12	14.76 ± 0.90	18.41 ± 0.69	16.22 ± 0.82	**19.64** ± 0.91
80	16.35 ± 1.35	14.38 ± 0.84	18.48 ± 0.81	15.89 ± 0.52	**19.92** ± 0.87
96	16.35 ± 1.08	14.40 ± 0.76	18.23 ± 0.60	16.02 ± 0.61	**19.61** ± 0.74
112	16.58 ± 1.10	14.69 ± 0.85	18.41 ± 0.67	16.32 ± 0.40	**20.36** ± 0.61
128	16.63 ± 0.83	14.57 ± 0.45	18.28 ± 0.56	16.43 ± 0.60	**20.28** ± 0.48
			AIME		
K	**PPL**	**SC**	**SC + Ours**	**RPC**	**RPC + Ours**
1	**1.50** ± 0.37	**1.50** ± 0.37	**1.50** ± 0.37	**1.50** ± 0.37	**1.50** ± 0.37
2	**1.56** ± 0.28	1.41 ± 0.20	**1.56** ± 0.28	**1.56** ± 0.28	**1.56** ± 0.28
4	1.70 ± 0.35	1.68 ± 0.24	1.70 ± 0.36	**1.77** ± 0.39	1.69 ± 0.36
8	1.78 ± 0.27	2.01 ± 0.35	1.99 ± 0.39	**2.14** ± 0.51	1.92 ± 0.32
16	1.88 ± 0.30	2.33 ± 0.26	**2.45** ± 0.34	2.37 ± 0.36	2.37 ± 0.35
32	1.64 ± 0.24	2.40 ± 0.29	**2.76** ± 0.34	2.71 ± 0.39	2.64 ± 0.31
40	1.75 ± 0.28	2.46 ± 0.29	**2.98** ± 0.38	2.77 ± 0.28	2.86 ± 0.36
48	1.80 ± 0.22	2.50 ± 0.25	**3.02** ± 0.23	2.93 ± 0.28	3.00 ± 0.36
64	1.60 ± 0.26	2.56 ± 0.17	3.10 ± 0.20	2.94 ± 0.25	**3.18** ± 0.29
80	1.61 ± 0.26	2.58 ± 0.16	3.30 ± 0.30	3.19 ± 0.19	**3.38** ± 0.28
96	1.67 ± 0.27	2.63 ± 0.23	3.37 ± 0.15	3.20 ± 0.22	**3.45** ± 0.17
112	1.61 ± 0.27	2.67 ± 0.21	3.37 ± 0.13	3.27 ± 0.18	**3.54** ± 0.24
128	1.66 ± 0.15	2.66 ± 0.20	3.37 ± 0.17	3.28 ± 0.23	**3.57** ± 0.21

Table 20: Accuracy across sampling budgets for **InternLM2-Math-Plus-7B** on MATH, MathOdyssey, and AIME.

MATH					
K	**PPL**	**SC**	**SC + Ours**	**RPC**	**RPC + Ours**
1	**41.69** ± 0.42	**41.69** ± 0.42	**41.69** ± 0.42	**41.69** ± 0.42	**41.69** ± 0.42
2	**43.93** ± 0.39	41.73 ± 0.22	**43.93** ± 0.39	**43.93** ± 0.39	**43.93** ± 0.39
4	45.57 ± 0.38	45.92 ± 0.26	45.51 ± 0.37	**47.09** ± 0.33	45.48 ± 0.38
8	46.49 ± 0.42	48.45 ± 0.21	49.06 ± 0.51	**49.10** ± 0.18	48.54 ± 0.46
16	47.14 ± 0.22	49.76 ± 0.16	**52.07** ± 0.19	50.24 ± 0.13	51.80 ± 0.17
32	47.09 ± 0.26	50.38 ± 0.26	**53.40** ± 0.28	51.17 ± 0.29	53.18 ± 0.18
40	46.96 ± 0.29	50.48 ± 0.10	**53.62** ± 0.22	51.22 ± 0.14	53.31 ± 0.27
48	46.96 ± 0.26	50.51 ± 0.17	**53.70** ± 0.20	51.40 ± 0.16	53.58 ± 0.21
64	47.00 ± 0.20	50.57 ± 0.18	**53.99** ± 0.18	51.94 ± 0.14	53.81 ± 0.12
MathOdyssey					
K	**PPL**	**SC**	**SC + Ours**	**RPC**	**RPC + Ours**
1	**20.57** ± 1.13	**20.57** ± 1.13	**20.57** ± 1.13	**20.57** ± 1.13	**20.57** ± 1.13
2	**23.42** ± 1.17	20.22 ± 0.65	**23.42** ± 1.17	**23.42** ± 1.17	**23.42** ± 1.17
4	25.40 ± 0.90	23.77 ± 0.96	**25.42** ± 0.97	25.27 ± 1.43	25.37 ± 1.16
8	27.51 ± 0.84	25.60 ± 0.80	**29.36** ± 1.19	27.15 ± 1.08	29.10 ± 1.12
16	27.92 ± 1.33	26.69 ± 1.09	**32.12** ± 1.01	27.94 ± 1.08	31.88 ± 1.04
32	28.33 ± 0.73	27.25 ± 1.01	34.14 ± 1.30	29.23 ± 1.51	**34.29** ± 1.31
40	28.66 ± 1.21	27.58 ± 0.83	**34.73** ± 1.01	29.67 ± 1.36	34.40 ± 0.90
48	28.97 ± 1.49	27.79 ± 0.74	34.81 ± 1.00	29.92 ± 0.89	**35.04** ± 0.92
64	28.59 ± 1.35	27.65 ± 0.70	34.96 ± 0.69	30.93 ± 1.36	**35.60** ± 1.07
80	28.17 ± 0.96	27.76 ± 0.44	35.01 ± 0.72	31.34 ± 0.77	**35.81** ± 0.81
96	28.30 ± 1.01	27.99 ± 0.51	35.14 ± 0.85	31.23 ± 0.76	**36.20** ± 0.52
112	27.61 ± 0.73	28.32 ± 0.55	35.17 ± 0.57	31.54 ± 0.61	**36.56** ± 0.68
128	27.51 ± 1.18	28.30 ± 0.61	35.30 ± 0.55	31.70 ± 0.71	**36.53** ± 1.06
AIME					
K	**PPL**	**SC**	**SC + Ours**	**RPC**	**RPC + Ours**
1	**5.06** ± 0.30	**5.06** ± 0.30	**5.06** ± 0.30	**5.06** ± 0.30	**5.06** ± 0.30
2	**5.66** ± 0.54	5.19 ± 0.26	**5.66** ± 0.54	**5.66** ± 0.54	**5.66** ± 0.54
4	5.93 ± 0.35	6.23 ± 0.31	5.92 ± 0.34	**6.57** ± 0.35	5.89 ± 0.37
8	6.14 ± 0.48	7.49 ± 0.67	7.15 ± 0.61	**7.86** ± 0.68	6.86 ± 0.69
16	6.21 ± 0.72	8.51 ± 0.35	8.64 ± 0.35	**8.71** ± 0.43	8.48 ± 0.41
32	6.31 ± 0.60	8.89 ± 0.34	**9.44** ± 0.39	9.10 ± 0.33	9.12 ± 0.33
40	6.21 ± 0.72	9.04 ± 0.26	**9.56** ± 0.41	9.37 ± 0.44	9.39 ± 0.28
48	6.37 ± 0.69	9.05 ± 0.30	**9.63** ± 0.34	9.51 ± 0.44	9.60 ± 0.42
64	6.48 ± 0.43	9.24 ± 0.21	9.68 ± 0.44	9.54 ± 0.33	**9.97** ± 0.35
80	6.41 ± 0.49	9.33 ± 0.22	9.66 ± 0.29	9.69 ± 0.38	**9.92** ± 0.45
96	6.28 ± 0.43	9.31 ± 0.12	9.87 ± 0.32	9.86 ± 0.21	**10.05** ± 0.34
112	6.11 ± 0.44	9.39 ± 0.21	9.87 ± 0.35	9.69 ± 0.15	**10.09** ± 0.38
128	5.96 ± 0.48	9.40 ± 0.21	9.95 ± 0.24	9.74 ± 0.23	**10.08** ± 0.24

Table 21: Online aggregation across the two InternLM2-Math-Plus backbones. $\overline{K}$ is the average number of sampled paths. Gain is the absolute accuracy change (%) against Online SC under the same stopping rule, and saving is relative to $K_{\max}$, which is 64 for MATH and 128 otherwise.

Dataset	Method	Accuracy		Sampling Cost	
		Acc.	Gain over Online SC	$\overline{K}$	Saving
InternLM2-Math-Plus-1.8B					
MATH	SC (Full Budget)	$36.48_{\pm 0.14}$	–	64.0	–
	Adaptive SC	$36.49_{\pm 0.16}$	–	36.3	43.3%
	Online SC	$36.38_{\pm 0.19}$	–	33.4	47.7%
	Online SC + PGLA	$\mathbf{39.32}_{\pm 0.31}$	$+\mathbf{2.95}$	31.5	50.8%
MathOdyssey	SC (Full Budget)	$14.57_{\pm 0.45}$	–	128.0	–
	Adaptive SC	$14.48_{\pm 0.46}$	–	70.0	45.3%
	Online SC	$14.47_{\pm 0.93}$	–	37.4	70.8%
	Online SC + PGLA	$\mathbf{18.20}_{\pm 0.90}$	$+\mathbf{3.73}$	34.5	73.1%
AIME	SC (Full Budget)	$2.66_{\pm 0.20}$	–	128.0	–
	Adaptive SC	$2.66_{\pm 0.19}$	–	92.6	27.7%
	Online SC	$2.55_{\pm 0.24}$	–	45.4	64.5%
	Online SC + PGLA	$\mathbf{2.94}_{\pm 0.29}$	$+\mathbf{0.39}$	38.3	70.1%
InternLM2-Math-Plus-7B					
MATH	SC (Full Budget)	$50.57_{\pm 0.18}$	–	64.0	–
	Adaptive SC	$50.55_{\pm 0.22}$	–	27.2	57.5%
	Online SC	$50.51_{\pm 0.20}$	–	30.2	52.8%
	Online SC + PGLA	$\mathbf{53.49}_{\pm 0.22}$	$+\mathbf{2.98}$	29.0	54.7%
MathOdyssey	SC (Full Budget)	$28.30_{\pm 0.61}$	–	128.0	–
	Adaptive SC	$28.22_{\pm 0.67}$	–	58.8	54.1%
	Online SC	$27.79_{\pm 0.86}$	–	35.3	72.4%
	Online SC + PGLA	$\mathbf{34.19}_{\pm 0.94}$	$+\mathbf{6.40}$	32.0	75.0%
AIME	SC (Full Budget)	$9.40_{\pm 0.21}$	–	128.0	–
	Adaptive SC	$9.38_{\pm 0.23}$	–	81.6	36.3%
	Online SC	$9.04_{\pm 0.41}$	–	42.0	67.2%
	Online SC + PGLA	$\mathbf{9.50}_{\pm 0.42}$	$+\mathbf{0.46}$	36.3	71.7%

predictions every eight paths and stop after the winner remains unchanged for three consecutive updates. We impose no additional warm-up threshold, making $K = 24$ the earliest possible stopping point. To isolate the effect of PGLA, its online variant uses count-based SC support, i.e., $s^{\mathrm{SC}}_{u,j} = |C_{u,j}|$. We fix this protocol across the evaluated datasets and include Adaptive SC with its original stopping criterion as a representative online baseline.

Performance in the Online Setting. As shown in Tab. 21, Online SC+PGLA consistently outperforms Online SC across the evaluated datasets, yielding gains of 0.39–6.40% under the same stopping rule. Meanwhile, it reduces the sampling cost by 50.8%–75.0% and uses fewer paths than Online SC in every setting. Despite this reduction, Online SC+PGLA remains 0.10–5.89% above full-budget SC. These results show that the comparative evidence introduced by PGLA improves both the accuracy and the sampling efficiency of online aggregation.

G.10 Extension to 32B Models

We further evaluate the aggregation methods in Tab. 1 using Qwen3-32B in reasoning mode on AIME24, AIME25, and HMMT25 (February). Each benchmark contains 30 problems with 64 sampled responses per problem. All methods use the same $K = 32$ subsets.

Tab. 22 shows that PGLA matches or improves the corresponding voting baselines on Qwen3-32B, with the most gains on HMMT25. These results suggest that the benefits of pairwise confidence comparisons extend to larger reasoning models.

Table 22: Accuracy (%) of Qwen3-32B in reasoning mode at $K = 32$.

Method	AIME24	AIME25	HMMT25
Qwen3-32B Reasoning mode			
PPL Best-of-N	$75.67_{\pm 3.96}$	$67.33_{\pm 6.29}$	$41.00_{\pm 3.96}$
DeepConf Best-of-N	$84.33_{\pm 3.96}$	$79.33_{\pm 2.00}$	$55.33_{\pm 2.67}$
Self-Certainty Best-of-N	$88.33_{\pm 2.69}$	$77.00_{\pm 2.33}$	$62.33_{\pm 3.00}$
SC	$90.17_{\pm 1.57}$	$81.33_{\pm 1.63}$	$59.50_{\pm 2.67}$
+ Ours (DeepConf)	$\mathbf{90.67}_{\pm 2.00}$	$\mathbf{81.67}_{\pm 1.67}$	$63.67_{\pm 2.77}$
+ Ours (Self-Certainty)	$90.33_{\pm 1.80}$	$81.33_{\pm 1.63}$	$\mathbf{65.00}_{\pm 2.24}$
DeepConf-Weighted SC	$90.33_{\pm 1.80}$	$81.33_{\pm 1.63}$	$60.67_{\pm 2.91}$
+ Ours	$\mathbf{90.67}_{\pm 2.00}$	$\mathbf{81.67}_{\pm 1.67}$	$\mathbf{63.67}_{\pm 1.80}$
Self-Certainty-Weighted SC	$\mathbf{90.33}_{\pm 1.80}$	$\mathbf{81.33}_{\pm 1.63}$	$60.67_{\pm 3.27}$
+ Ours	$\mathbf{90.33}_{\pm 1.80}$	$\mathbf{81.33}_{\pm 1.63}$	$\mathbf{65.00}_{\pm 1.67}$
Self-Certainty Borda	$89.00_{\pm 2.13}$	$80.33_{\pm 1.00}$	$64.33_{\pm 3.00}$
+ Ours	$\mathbf{89.33}_{\pm 2.00}$	$\mathbf{81.00}_{\pm 1.53}$	$\mathbf{65.00}_{\pm 2.24}$
DeepConf-Filter (90%)	$90.33_{\pm 1.80}$	$81.33_{\pm 1.63}$	$64.00_{\pm 3.27}$
+ Ours	$\mathbf{90.67}_{\pm 2.00}$	$\mathbf{82.00}_{\pm 1.63}$	$\mathbf{65.00}_{\pm 2.69}$

G.11 Effect of Comparative Advantage Rules

The comparative term in PGLA is modular: changing its aggregation rule leaves the voting distribution and KL-regularized update unchanged. For $U = |\mathcal{A}| > 1$, we compare the default voting-weighted comparative advantage in Eq. (7) with three alternatives:

$$s_u^{\text{Pair}} = \frac{\sum_{v \neq u} s_v^{\mathcal{V}} Y_{uv}}{\sum_{v \neq u} s_v^{\mathcal{V}}}, \qquad s_u^{\text{Pair,Unif}} = \frac{1}{U-1} \sum_{v \neq u} Y_{uv},$$
$$s_u^{\text{Pair,Copeland}} = \frac{1}{U-1} \sum_{v \neq u} \operatorname{sgn}(Y_{uv}), \quad s_u^{\text{Pair,Maximin}} = \min_{v \neq u} Y_{uv}. \tag{32}$$

Uniform averaging takes the mean of signed pairwise scores; Copeland retains only their signs, with ties contributing zero (Zwicker, 2016); Maximin retains the smallest signed margin (Fischer et al., 2016). All four scores lie in $[-1, 1]$ and are set to zero for $U = 1$. The weighted score is also zero when its denominator vanishes. These rules aggregate confidence-based comparisons, and the default weights apply to opposing answers. Each alternative can replace s_u^{Pair} in Eq. (9).

Matched protocol. We extend Tab. 2 using InternLM2-Math-Plus-1.8B, InternLM2-Math-Plus-7B, and DeepSeek-Math-RL-7B on MATH ($K = 64$), MathOdyssey ($K = 128$), and AIME ($K = 128$), using internal mean log-probability scores. We evaluate SC, Weighted SC, Adaptive SC, and RPC. Only the comparative rule changes.

Results. Tabs. 23 to 25 report the results by model. Voting-weighted comparisons attain higher mean accuracy than uniform averaging in most settings, with macro-average accuracy of 29.06% versus 28.70%, but uniform averaging yields lower macro-average ECE. Copeland can outperform voting-weighted comparisons and has slightly higher model-average accuracy on InternLM2-1.8B, but lower averages on the other models. Maximin is consistently less accurate. Overall, all four rules improve accuracy over their corresponding baselines in most evaluated settings, supporting the effectiveness of incorporating comparative advantage and demonstrating that our general framework accommodates diverse comparative advantage rules.

Table 23: Comparative-advantage functions on **InternLM2-Math-Plus-1.8B**. We expand Tab. 2 using the same coefficients (App. G.11). Voting denotes the default voting-weighted comparative advantage and **+ Ours** denotes the scaled PGLA variants described in App. G.11.

Method	MATH		MathOdyssey		AIME		Average	
	Acc. ↑	ECE ↓	Acc. ↑	ECE ↓	Acc. ↑	ECE ↓	Acc. ↑	ECE ↓
InternLM2-Math-Plus-1.8B								
PPL	$33.23_{\pm 0.25}$	$61.58_{\pm 0.25}$	$16.63_{\pm 0.83}$	$77.20_{\pm 0.82}$	$1.66_{\pm 0.15}$	$92.28_{\pm 0.16}$	17.17	77.02
SC	$36.48_{\pm 0.14}$	$6.66_{\pm 0.14}$	$14.57_{\pm 0.45}$	$18.51_{\pm 0.41}$	$2.66_{\pm 0.20}$	$16.84_{\pm 0.16}$	17.90	14.00
+ Ours (voting)	$\mathbf{39.77}_{\pm 0.17}$	$5.63_{\pm 0.13}$	$18.28_{\pm 0.56}$	$13.97_{\pm 0.59}$	$3.37_{\pm 0.17}$	$15.51_{\pm 0.15}$	20.47	11.70
+ Ours (Uniform)	$39.24_{\pm 0.14}$	$\mathbf{4.94}_{\pm 0.13}$	$17.46_{\pm 0.61}$	$14.39_{\pm 0.69}$	$3.25_{\pm 0.14}$	$\mathbf{15.34}_{\pm 0.11}$	19.98	11.55
+ Ours (Copeland)	$39.65_{\pm 0.17}$	$5.71_{\pm 0.16}$	$\mathbf{18.87}_{\pm 0.62}$	$\mathbf{13.37}_{\pm 0.58}$	$\mathbf{3.53}_{\pm 0.12}$	$15.56_{\pm 0.17}$	**20.68**	**11.55**
+ Ours (Maximin)	$38.17_{\pm 0.19}$	$6.19_{\pm 0.26}$	$16.13_{\pm 0.33}$	$16.35_{\pm 0.37}$	$2.78_{\pm 0.25}$	$16.56_{\pm 0.25}$	19.03	13.04
Weighted SC	$36.72_{\pm 0.12}$	$6.52_{\pm 0.14}$	$14.86_{\pm 0.47}$	$18.05_{\pm 0.50}$	$2.75_{\pm 0.23}$	$16.65_{\pm 0.19}$	18.11	13.74
+ Ours (voting)	$\mathbf{39.91}_{\pm 0.22}$	$5.55_{\pm 0.17}$	$18.56_{\pm 0.56}$	$13.47_{\pm 0.56}$	$3.37_{\pm 0.17}$	$15.34_{\pm 0.15}$	20.61	11.45
+ Ours (Uniform)	$39.40_{\pm 0.17}$	$\mathbf{4.83}_{\pm 0.19}$	$17.53_{\pm 0.73}$	$14.16_{\pm 0.79}$	$3.32_{\pm 0.19}$	$\mathbf{15.11}_{\pm 0.15}$	20.09	11.37
+ Ours (Copeland)	$39.75_{\pm 0.19}$	$5.69_{\pm 0.17}$	$\mathbf{19.13}_{\pm 0.48}$	$\mathbf{13.00}_{\pm 0.42}$	$\mathbf{3.59}_{\pm 0.13}$	$15.40_{\pm 0.19}$	**20.82**	**11.36**
+ Ours (Maximin)	$38.28_{\pm 0.15}$	$6.21_{\pm 0.24}$	$16.38_{\pm 0.42}$	$15.95_{\pm 0.39}$	$2.83_{\pm 0.20}$	$16.39_{\pm 0.22}$	19.16	12.85
Adaptive SC	$36.48_{\pm 0.16}$	$10.61_{\pm 0.23}$	$14.43_{\pm 0.51}$	$22.69_{\pm 0.73}$	$2.65_{\pm 0.20}$	$19.25_{\pm 0.25}$	17.85	17.52
+ Ours (voting)	$\mathbf{39.05}_{\pm 0.23}$	$9.28_{\pm 0.27}$	$17.47_{\pm 0.63}$	$17.96_{\pm 0.60}$	$3.23_{\pm 0.21}$	$17.53_{\pm 0.41}$	**19.92**	14.92
+ Ours (Uniform)	$38.74_{\pm 0.20}$	$\mathbf{8.66}_{\pm 0.24}$	$16.79_{\pm 0.63}$	$18.22_{\pm 0.84}$	$3.14_{\pm 0.14}$	$\mathbf{17.31}_{\pm 0.31}$	19.56	**14.73**
+ Ours (Copeland)	$38.21_{\pm 0.18}$	$10.26_{\pm 0.18}$	$\mathbf{17.76}_{\pm 0.80}$	$\mathbf{17.66}_{\pm 0.90}$	$\mathbf{3.40}_{\pm 0.19}$	$17.59_{\pm 0.47}$	19.79	15.17
+ Ours (Maximin)	$36.95_{\pm 0.28}$	$10.77_{\pm 0.31}$	$16.18_{\pm 0.88}$	$19.32_{\pm 1.28}$	$2.75_{\pm 0.28}$	$18.46_{\pm 0.44}$	18.63	16.18
RPC	$37.87_{\pm 0.14}$	$6.45_{\pm 0.15}$	$16.43_{\pm 0.60}$	$16.25_{\pm 0.76}$	$3.28_{\pm 0.23}$	$16.32_{\pm 0.20}$	19.19	13.01
+ Ours (voting)	$\mathbf{40.29}_{\pm 0.27}$	$5.50_{\pm 0.22}$	$20.28_{\pm 0.48}$	$\mathbf{10.78}_{\pm 0.60}$	$3.57_{\pm 0.21}$	$14.67_{\pm 0.23}$	**21.38**	**10.32**
+ Ours (Uniform)	$39.90_{\pm 0.25}$	$\mathbf{4.83}_{\pm 0.29}$	$19.00_{\pm 0.48}$	$11.89_{\pm 0.55}$	$3.52_{\pm 0.12}$	$\mathbf{14.63}_{\pm 0.12}$	20.80	10.45
+ Ours (Copeland)	$40.13_{\pm 0.23}$	$5.87_{\pm 0.20}$	$\mathbf{20.33}_{\pm 0.62}$	$11.31_{\pm 0.86}$	$\mathbf{3.63}_{\pm 0.09}$	$15.23_{\pm 0.15}$	21.37	10.80
+ Ours (Maximin)	$38.81_{\pm 0.22}$	$6.82_{\pm 0.27}$	$17.92_{\pm 0.61}$	$14.01_{\pm 0.55}$	$3.09_{\pm 0.22}$	$16.37_{\pm 0.23}$	19.94	12.40

Table 24: Comparative-advantage functions on **InternLM2-Math-Plus-7B**. We expand Tab. 2 using the same coefficients (App. G.11). Voting denotes the default voting-weighted comparative advantage and **+ Ours** denotes the scaled PGLA variants described in App. G.11.

Method	MATH		MathOdyssey		AIME		Average	
	Acc. ↑	ECE ↓	Acc. ↑	ECE ↓	Acc. ↑	ECE ↓	Acc. ↑	ECE ↓
InternLM2-Math-Plus-7B								
PPL	$47.00_{\pm 0.20}$	$48.98_{\pm 0.20}$	$27.51_{\pm 1.18}$	$67.54_{\pm 1.18}$	$5.96_{\pm 0.48}$	$88.98_{\pm 0.49}$	26.82	68.50
SC	$50.57_{\pm 0.18}$	$6.72_{\pm 0.19}$	$28.30_{\pm 0.61}$	$12.18_{\pm 0.57}$	$9.40_{\pm 0.21}$	$14.35_{\pm 0.23}$	29.42	11.08
+ Ours (voting)	$\mathbf{53.99}_{\pm 0.18}$	$6.10_{\pm 0.20}$	$\mathbf{35.30}_{\pm 0.55}$	$9.69_{\pm 1.02}$	$9.95_{\pm 0.24}$	$13.90_{\pm 0.23}$	**33.08**	9.90
+ Ours (Uniform)	$53.72_{\pm 0.13}$	$\mathbf{5.13}_{\pm 0.23}$	$33.62_{\pm 0.72}$	$8.68_{\pm 0.75}$	$10.04_{\pm 0.23}$	$\mathbf{13.26}_{\pm 0.25}$	32.46	**9.02**
+ Ours (Copeland)	$52.98_{\pm 0.23}$	$6.54_{\pm 0.25}$	$34.37_{\pm 0.80}$	$\mathbf{8.52}_{\pm 1.18}$	$\mathbf{10.21}_{\pm 0.28}$	$13.67_{\pm 0.30}$	32.52	9.58
+ Ours (Maximin)	$52.56_{\pm 0.17}$	$6.25_{\pm 0.20}$	$31.79_{\pm 0.92}$	$10.16_{\pm 1.01}$	$9.17_{\pm 0.19}$	$14.75_{\pm 0.20}$	31.17	10.38
Weighted SC	$50.81_{\pm 0.20}$	$6.56_{\pm 0.21}$	$28.82_{\pm 0.67}$	$11.57_{\pm 0.68}$	$9.45_{\pm 0.23}$	$14.27_{\pm 0.25}$	29.69	10.80
+ Ours (voting)	$\mathbf{53.99}_{\pm 0.13}$	$6.15_{\pm 0.17}$	$\mathbf{35.78}_{\pm 0.51}$	$9.37_{\pm 1.16}$	$9.88_{\pm 0.28}$	$13.86_{\pm 0.27}$	**33.22**	9.79
+ Ours (Uniform)	$53.82_{\pm 0.15}$	$\mathbf{5.08}_{\pm 0.29}$	$34.06_{\pm 0.81}$	$\mathbf{8.46}_{\pm 0.99}$	$10.01_{\pm 0.24}$	$\mathbf{13.21}_{\pm 0.24}$	32.63	**8.92**
+ Ours (Copeland)	$53.02_{\pm 0.21}$	$6.57_{\pm 0.23}$	$34.55_{\pm 0.75}$	$8.50_{\pm 1.01}$	$\mathbf{10.29}_{\pm 0.36}$	$13.55_{\pm 0.40}$	32.62	9.54
+ Ours (Maximin)	$52.67_{\pm 0.17}$	$6.20_{\pm 0.19}$	$32.11_{\pm 0.72}$	$10.01_{\pm 0.98}$	$9.22_{\pm 0.18}$	$14.64_{\pm 0.20}$	31.33	10.28
Adaptive SC	$50.54_{\pm 0.22}$	$11.02_{\pm 0.23}$	$28.20_{\pm 0.68}$	$18.17_{\pm 1.02}$	$9.38_{\pm 0.23}$	$17.39_{\pm 0.41}$	29.37	15.53
+ Ours (voting)	$\mathbf{52.66}_{\pm 0.26}$	$10.53_{\pm 0.35}$	$\mathbf{33.14}_{\pm 0.73}$	$\mathbf{13.22}_{\pm 1.52}$	$9.83_{\pm 0.21}$	$16.45_{\pm 0.33}$	**31.87**	13.40
+ Ours (Uniform)	$52.62_{\pm 0.28}$	$\mathbf{9.60}_{\pm 0.39}$	$31.93_{\pm 0.46}$	$13.31_{\pm 0.98}$	$\mathbf{9.87}_{\pm 0.19}$	$\mathbf{15.90}_{\pm 0.34}$	31.47	**12.94**
+ Ours (Copeland)	$50.73_{\pm 0.26}$	$12.43_{\pm 0.31}$	$31.75_{\pm 1.00}$	$14.22_{\pm 1.21}$	$9.73_{\pm 0.28}$	$16.58_{\pm 0.35}$	30.74	14.41
+ Ours (Maximin)	$50.66_{\pm 0.35}$	$11.71_{\pm 0.36}$	$30.73_{\pm 0.64}$	$14.99_{\pm 1.19}$	$8.57_{\pm 0.27}$	$17.83_{\pm 0.22}$	29.99	14.85
RPC	$51.94_{\pm 0.14}$	$6.41_{\pm 0.16}$	$31.70_{\pm 0.71}$	$9.88_{\pm 0.80}$	$9.74_{\pm 0.23}$	$14.31_{\pm 0.21}$	31.13	10.20
+ Ours (voting)	$\mathbf{53.81}_{\pm 0.12}$	$6.72_{\pm 0.15}$	$\mathbf{36.53}_{\pm 1.06}$	$7.90_{\pm 1.09}$	$10.08_{\pm 0.24}$	$13.31_{\pm 0.22}$	**33.47**	9.31
+ Ours (Uniform)	$53.80_{\pm 0.10}$	$\mathbf{5.74}_{\pm 0.26}$	$35.19_{\pm 0.59}$	$8.22_{\pm 0.93}$	$10.14_{\pm 0.32}$	$\mathbf{12.98}_{\pm 0.33}$	33.04	**8.98**
+ Ours (Copeland)	$52.83_{\pm 0.19}$	$7.36_{\pm 0.21}$	$35.12_{\pm 1.04}$	$\mathbf{7.87}_{\pm 1.59}$	$\mathbf{10.20}_{\pm 0.32}$	$13.70_{\pm 0.38}$	32.72	9.64
+ Ours (Maximin)	$52.79_{\pm 0.22}$	$6.97_{\pm 0.23}$	$33.73_{\pm 0.79}$	$9.72_{\pm 1.34}$	$9.68_{\pm 0.21}$	$14.45_{\pm 0.23}$	32.07	10.38

Table 25: Comparative-advantage functions on **DeepSeek-Math-RL-7B**. We expand Tab. 2 using the same coefficients (App. G.11). Voting denotes the default voting-weighted comparative advantage and **+ Ours** denotes the scaled PGLA variants described in App. G.11.

Method	MATH		MathOdyssey		AIME		Average	
	Acc. ↑	ECE ↓	Acc. ↑	ECE ↓	Acc. ↑	ECE ↓	Acc. ↑	ECE ↓
DeepSeek-Math-RL-7B								
PPL	$42.50_{\pm 0.23}$	$53.84_{\pm 0.23}$	$22.34_{\pm 1.00}$	$73.85_{\pm 0.99}$	$3.37_{\pm 0.46}$	$93.38_{\pm 0.46}$	22.74	73.69
SC	$53.33_{\pm 0.08}$	$\mathbf{6.43}_{\pm 0.11}$	$36.57_{\pm 0.56}$	$9.40_{\pm 1.21}$	$9.42_{\pm 0.23}$	$12.12_{\pm 0.26}$	33.11	9.32
+ Ours (voting)	$\mathbf{53.52}_{\pm 0.10}$	$6.86_{\pm 0.09}$	$37.87_{\pm 0.55}$	$9.31_{\pm 0.95}$	$\mathbf{10.02}_{\pm 0.23}$	$10.88_{\pm 0.25}$	**33.80**	9.02
+ Ours (Uniform)	$53.49_{\pm 0.11}$	$6.57_{\pm 0.09}$	$37.51_{\pm 0.64}$	$9.24_{\pm 0.76}$	$9.99_{\pm 0.22}$	$\mathbf{10.78}_{\pm 0.20}$	33.66	8.87
+ Ours (Copeland)	$53.30_{\pm 0.16}$	$7.17_{\pm 0.14}$	$\mathbf{38.12}_{\pm 0.55}$	$\mathbf{8.12}_{\pm 0.87}$	$9.90_{\pm 0.25}$	$11.20_{\pm 0.23}$	33.78	**8.83**
+ Ours (Maximin)	$53.38_{\pm 0.15}$	$6.56_{\pm 0.15}$	$37.10_{\pm 0.48}$	$9.29_{\pm 1.27}$	$9.26_{\pm 0.25}$	$11.99_{\pm 0.27}$	33.25	9.28
Weighted SC	$53.35_{\pm 0.09}$	$\mathbf{6.43}_{\pm 0.10}$	$36.68_{\pm 0.61}$	$9.55_{\pm 1.31}$	$9.43_{\pm 0.31}$	$12.10_{\pm 0.34}$	33.16	9.36
+ Ours (voting)	$\mathbf{53.53}_{\pm 0.10}$	$6.84_{\pm 0.09}$	$37.94_{\pm 0.58}$	$9.53_{\pm 0.79}$	$\mathbf{10.00}_{\pm 0.24}$	$10.84_{\pm 0.25}$	**33.83**	9.07
+ Ours (Uniform)	$53.48_{\pm 0.10}$	$6.59_{\pm 0.07}$	$37.56_{\pm 0.69}$	$9.05_{\pm 0.83}$	$9.98_{\pm 0.21}$	$\mathbf{10.75}_{\pm 0.17}$	33.67	**8.80**
+ Ours (Copeland)	$53.30_{\pm 0.18}$	$7.17_{\pm 0.17}$	$\mathbf{38.20}_{\pm 0.60}$	$\mathbf{8.14}_{\pm 0.92}$	$9.92_{\pm 0.30}$	$11.15_{\pm 0.28}$	33.81	8.82
+ Ours (Maximin)	$53.40_{\pm 0.17}$	$6.54_{\pm 0.18}$	$37.12_{\pm 0.53}$	$9.32_{\pm 1.19}$	$9.31_{\pm 0.30}$	$11.92_{\pm 0.33}$	33.28	9.26
Adaptive SC	$53.28_{\pm 0.09}$	$10.01_{\pm 0.14}$	$\mathbf{36.30}_{\pm 0.58}$	$13.25_{\pm 1.14}$	$9.36_{\pm 0.22}$	$14.88_{\pm 0.41}$	32.98	12.71
+ Ours (voting)	$\mathbf{53.44}_{\pm 0.16}$	$10.10_{\pm 0.15}$	$36.27_{\pm 0.80}$	$\mathbf{12.33}_{\pm 0.87}$	$\mathbf{9.92}_{\pm 0.23}$	$13.31_{\pm 0.49}$	**33.21**	11.91
+ Ours (Uniform)	$53.41_{\pm 0.15}$	$9.88_{\pm 0.15}$	$36.22_{\pm 0.87}$	$12.47_{\pm 1.62}$	$9.79_{\pm 0.22}$	$\mathbf{13.20}_{\pm 0.51}$	33.14	**11.85**
+ Ours (Copeland)	$53.09_{\pm 0.17}$	$10.45_{\pm 0.25}$	$33.57_{\pm 1.11}$	$14.89_{\pm 1.22}$	$9.54_{\pm 0.25}$	$13.93_{\pm 0.44}$	32.07	13.09
+ Ours (Maximin)	$53.29_{\pm 0.15}$	$\mathbf{9.85}_{\pm 0.23}$	$34.58_{\pm 0.86}$	$13.50_{\pm 0.95}$	$8.43_{\pm 0.52}$	$14.93_{\pm 0.66}$	32.10	12.76
RPC	$53.36_{\pm 0.11}$	$\mathbf{6.47}_{\pm 0.12}$	$37.12_{\pm 0.61}$	$9.54_{\pm 1.54}$	$9.51_{\pm 0.28}$	$11.94_{\pm 0.32}$	33.33	9.32
+ Ours (voting)	$\mathbf{53.52}_{\pm 0.11}$	$6.85_{\pm 0.10}$	$38.10_{\pm 0.61}$	$8.89_{\pm 0.76}$	$10.01_{\pm 0.26}$	$10.44_{\pm 0.29}$	**33.88**	8.73
+ Ours (Uniform)	$53.48_{\pm 0.10}$	$6.60_{\pm 0.09}$	$\mathbf{38.12}_{\pm 0.65}$	$8.17_{\pm 0.93}$	$\mathbf{10.02}_{\pm 0.27}$	$\mathbf{10.36}_{\pm 0.34}$	33.87	**8.38**
+ Ours (Copeland)	$53.31_{\pm 0.18}$	$7.16_{\pm 0.17}$	$38.07_{\pm 0.68}$	$\mathbf{7.62}_{\pm 0.81}$	$9.99_{\pm 0.36}$	$10.90_{\pm 0.34}$	33.79	8.56
+ Ours (Maximin)	$53.40_{\pm 0.18}$	$6.57_{\pm 0.18}$	$37.48_{\pm 0.73}$	$9.19_{\pm 1.17}$	$9.34_{\pm 0.28}$	$11.85_{\pm 0.33}$	33.41	9.20

G.12 Computational Overhead

Complexity. Let K be the number of sampled paths, $U = |\mathcal{A}|$ the number of candidate answers, and $n_u = |C_u|$. Given answer groups and confidence scores, accumulating voting support requires $O(K+U)$ time. Directly evaluating the pairwise correction in Eqs. (6) and (7) takes $O(K^2)$ time. The number of unordered cross-answer path pairs is $\sum_{u<v} n_u n_v = (K^2 - \sum_u n_u^2)/2 \le K(K-1)/2$. At our experimental budgets $K = 32, 64, 128$, this is at most 496, 2,016, and 8,128 scalar comparisons, respectively. The final score adjustment in Eq. (10) is computed in closed form in $O(U)$ time, without iterative optimization.

Impact on inference time. Pairwise comparisons reuse the scalar confidence scores of the sampled paths; they require neither additional reasoning paths nor per-pair LLM or verifier calls. Even direct evaluation has a reported median aggregation latency below 0.109 ms per query on the real candidate pools in Tab. 26. For context, the $K = 32$ Qwen3-4B/AIME 2024 pools contain approximately 4.77×10^5 generated tokens per pool on average. Relative to this GPU-based autoregressive generation workload, sub-millisecond CPU post-processing is expected to have a negligible effect on total inference time. This assessment concerns aggregation given available scores; separate confidence-model inference is not included, and we do not infer a measured end-to-end latency percentage from these timings.

Exact acceleration. To further reduce aggregation latency as the candidate pool grows, we implement an exact sorting-and-prefix-sum evaluation of the default voting-weighted correction. Define

$$w_i = \frac{s^{\mathcal{V}}_{a_i}}{n_{a_i}}, \qquad d_i = \sum_{j:p_j<p_i} w_j - \sum_{j:p_j>p_i} w_j, \qquad s^{\mathrm{Pair}}_u = \frac{\sum_{i\in C_u} d_i}{n_u \sum_{v\neq u} s^{\mathcal{V}}_v}. \tag{33}$$

Within-answer signed comparisons cancel because all paths in the same group have the same comparison weight w_i, yielding the exact identity above. We sort confidence scores once and collect equal scores into blocks. Prefix and suffix sums of block weights give the total weights below and above each block, hence every d_i; ties contribute zero. A final group-wise reduction computes the correction and adjusted scores. A zero denominator gives zero correction, as in App. D. This implementation takes $O(K \log K + U)$ time and $O(K + U)$ working memory, avoiding a dense pairwise matrix while preserving the original aggregation rule.

Measured implementation cost. We compare two different implementations of PGLA with identical confidence-weighted voting support on the same device. Direct evaluation constructs a dense path-comparison array; the accelerated implementation uses Eq. (33). We use 300 cached Qwen3-4B/AIME 2024 pools with DeepConf ($K = 32$) and 1,000 InternLM2-Math-Plus-1.8B pools each from MATH ($K = 64$) and MathOdyssey ($K = 128$), with inverse-perplexity weights. At $K = 128$, sorting reduces the reported median aggregation time from 108.3 to 60.3 μs, a 1.80× speedup over direct evaluation. For $K = 32$ and 64, the vectorized direct implementation remains faster because of its lower fixed cost. Thus sorting improves scalability and the measured latency at the largest experimental budget, while direct evaluation remains efficient for smaller pools.

Table 26: CPU aggregation latency of two exact PGLA implementations (μs per query), measured on real cached candidate pools. Both use identical inputs and confidence-weighted voting support.

Implementation	$K = 32$ AIME 2024	$K = 64$ MATH	$K = 128$ MathOdyssey
Direct pairwise evaluation	29.7	48.9	108.3
Exact sorting + prefix sums	53.4	57.4	60.3

G.13 Connection to Rank-Based Borda Aggregation

Under SC, PGLA admits an exact rank representation. Let $n_u = |C_u|$, let $R_i \in [1, K]$ be the ascending confidence rank of path i (averaged for ties), and define $\bar{R}_u = n_u^{-1} \sum_{i \in C_u} R_i$. Rank-based Borda assigns answer u the score $T_u^{\text{Borda}} = \sum_{i \in C_u} R_i = n_u \bar{R}_u$.

Proposition G.1 (Rank representation under SC)**.** *For the SC base $s_u^{\mathcal{V}} = n_u$ and at least two observed answer groups, the correction in Eq. (7) satisfies*

$$s_u^{\text{Pair}} = \frac{2\bar{R}_u - K - 1}{K - n_u}. \tag{34}$$

*Consequently, SC-*PGLA *selects the answer maximizing* $\log n_u + (2\bar{R}_u - K - 1)/(K - n_u)$.

Proof. Under SC, the correction averages all signed cross-group comparisons. Summing the rank identity $\sum_j \text{sgn}(p_i - p_j) = 2R_i - K - 1$ over $i \in C_u$, canceling within-group terms by antisymmetry, and dividing by $n_u(K - n_u)$ gives Eq. (34). The selection rule follows from Eq. (10). □

Proposition G.1 makes the connection to rank-based voting precise. Under SC, both Borda and SC-PGLA use the same path-confidence ordering, but they aggregate it differently. With $\gamma = 1$, Borda sums ranks as $T_u^{(1)} = n_u \bar{R}_u$, whereas SC-PGLA selects by $\log n_u + (2\bar{R}_u - K - 1)/(K - n_u)$. The latter treats the ordering as a bounded cross-answer comparative correction while retaining voting support as a separate term. In particular, an answer u with $n_u < n_v$ can overtake v only if $s_u^{\text{Pair}} - s_v^{\text{Pair}} > \log(n_v/n_u)$. Thus, reversing a larger vote margin requires stronger comparative evidence. The rank identity in Eq. (34). is specific to SC; Eq. (7) also combines pairwise comparisons with confidence-weighted and externally scored voting support. The KL solution provides the mechanism for this combination, while the aggregation rule is determined by how the comparative evidence is constructed and weighted.

To test whether PGLA can also enhance Borda itself, we use $s_u^{\mathcal{V}} = T_u^{\text{Borda}}$ in Eqs. (7) and (10), including the opponent weights. Tabs. 27 and 28 compare SC and Borda with their PGLA-adjusted counterparts on the same sampled subsets.

Borda improves over SC on average, showing that confidence ordering can help correct voting-based selection errors. More importantly, applying PGLA directly to Borda raises macro-average accuracy from 59.17% to 61.21% on Qwen's AIME24 and HMMT25 comparisons, and from 25.89% to 27.76% across the six InternLM settings. These paired comparisons support **explicit answer-level preference modeling** as a way to further refine rank-based voting on the evaluated settings.

Table 27: Qwen3 reasoning-mode accuracy on AIME24 and HMMT25 (%; $K = 32$).

Method	Qwen3-1.7B		Qwen3-4B		Avg. results
	AIME24	HMMT25	AIME24	HMMT25	
DeepConf					
SC	$69.50_{\pm 1.98}$	$27.00_{\pm 1.00}$	$80.00_{\pm 0.00}$	$51.50_{\pm 1.57}$	57.00
SC-PGLA	$\mathbf{75.00}_{\pm 1.67}$	$30.00_{\pm 2.98}$	$83.00_{\pm 1.00}$	$\mathbf{54.00}_{\pm 2.91}$	60.50
Borda	$72.67_{\pm 2.49}$	$29.00_{\pm 3.35}$	$82.00_{\pm 1.63}$	$52.33_{\pm 2.60}$	59.00
Borda+PGLA	$74.00_{\pm 2.49}$	$\mathbf{33.17}_{\pm 3.83}$	$\mathbf{83.67}_{\pm 2.33}$	$\mathbf{54.00}_{\pm 2.91}$	**61.21**
Self-Certainty					
SC	$69.50_{\pm 1.98}$	$27.00_{\pm 1.00}$	$80.00_{\pm 0.00}$	$51.50_{\pm 1.57}$	57.00
SC-PGLA	$\mathbf{74.00}_{\pm 1.33}$	$30.33_{\pm 2.77}$	$82.00_{\pm 1.63}$	$\mathbf{55.67}_{\pm 1.53}$	60.50
Borda	$72.67_{\pm 2.00}$	$29.00_{\pm 2.60}$	$80.67_{\pm 1.33}$	$55.00_{\pm 1.67}$	59.33
Borda+PGLA	$\mathbf{74.00}_{\pm 2.00}$	$\mathbf{33.50}_{\pm 4.25}$	$\mathbf{82.33}_{\pm 2.13}$	$55.00_{\pm 1.67}$	**61.21**

Table 28: InternLM2-Math-Plus accuracy (%; PPL ordering), with $K = 64$ for MATH and $K = 128$ otherwise.

Method	InternLM2-Math-Plus-1.8B			InternLM2-Math-Plus-7B			Avg. results
	MATH	MathOdyssey	AIME	MATH	MathOdyssey	AIME	
SC	$36.48_{\pm 0.14}$	$14.57_{\pm 0.45}$	$2.66_{\pm 0.20}$	$50.57_{\pm 0.18}$	$28.30_{\pm 0.61}$	$9.40_{\pm 0.21}$	23.66
SC-PGLA	$39.77_{\pm 0.17}$	$18.28_{\pm 0.56}$	$3.37_{\pm 0.17}$	$\mathbf{53.99}_{\pm 0.18}$	$35.30_{\pm 0.55}$	$9.95_{\pm 0.24}$	26.77
Borda	$38.83_{\pm 0.08}$	$17.12_{\pm 0.62}$	$3.23_{\pm 0.18}$	$53.07_{\pm 0.16}$	$33.14_{\pm 0.59}$	$9.97_{\pm 0.12}$	25.89
Borda+PGLA	$\mathbf{40.58}_{\pm 0.21}$	$\mathbf{21.13}_{\pm 0.57}$	$\mathbf{3.56}_{\pm 0.20}$	$53.60_{\pm 0.16}$	$\mathbf{37.48}_{\pm 0.58}$	$\mathbf{10.20}_{\pm 0.34}$	**27.76**

H IMPACT STATEMENT

PGLA aims to make LLM reasoning more accurate and efficient by improving how sampled responses are aggregated. The observed gains in mathematical reasoning, knowledge reasoning, and code generation may benefit applications in problem solving and software development, while the online extension can reduce the number of sampled responses needed. These benefits do not eliminate errors or biases in the underlying models or confidence signals. In particular, agreement among sampled responses is not a guarantee of correctness, and stronger capabilities in reasoning and code generation may also facilitate misuse. Deployment should therefore include evaluation in the intended setting, appropriate safeguards, and human oversight for consequential decisions.

I AI USE STATEMENT

In this work, we used LLMs to generate the sampled responses evaluated in our test-time scaling experiments, and process reward models to score reasoning steps in the external-reward setting, as described in Sec. 4.1. Generative AI tools assisted with implementation and debugging, checks of mathematical notation and derivations, interpretation and presentation of experimental results, and manuscript drafting, language editing, and LaTeX formatting. AI assistance was also used to create parts of the illustrations, including the voting-game illustration in Fig. 1. The authors reviewed and revised AI-assisted manuscript content and take responsibility for the final text, claims, experimental results, code, and figures.