RECURSIVE SELF-VERIFICATION: PROGRESSIVE BINARY VERIFICATION FOR LANGUAGE MODEL SELF-EVOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-evolving language models learn from their own generations by using the model to identify responses that yield useful training data. However, directly assigning fine-grained, multi-level scores to responses is often unreliable, particularly for smaller models that struggle with subtle quality distinctions, resulting in unreliable preference pairs. To address this limitation, we investigate whether fine-grained self-verification can be achieved by decomposing complex evaluations into a sequence of simpler binary decisions. Specifically, we introduce Recursive Self-Verification (RSV), a concise yet effective framework that evaluates responses through sequential binary verification steps ordered from strict to relaxed quality criteria. At each step, RSV uses normalized token probabilities for Yes and No to recursively allocate probability mass across quality levels, deriving the final evaluation score as the expected value of this distribution. These refined scores are then used to rank responses and construct high-quality preference pairs for model optimization. Experiments on the Knights-and-Knaves, TruthQuest, and MATH-Hard benchmarks demonstrate that RSV-guided preference learning consistently outperforms baseline methods in reasoning tasks across various model scales. Notably, on K&K with Qwen3-1.7B, the best RSV setting achieves an average accuracy of 63.1%, outperforming the strongest non-RSV baseline by 8.3 percentage points. These gains also extend to GRPO, confirming RSV's effectiveness under on-policy optimization for language model self-evolution.

1 INTRODUCTION

Large language models (LLMs) are increasingly used to generate supervision for their own improvement (Zelikman et al., 2022; Huang et al., 2023; Wang et al., 2023). In the generate–verify–learn pipeline, a model produces candidate responses, evaluates their quality, and learns from the resulting supervision, reducing reliance on expanding human-annotated data (Zelikman et al., 2022; Yuan et al., 2024; Singh et al., 2024). Preference optimization methods such as Direct Preference Optimization (DPO) (Rafailov et al., 2023) further simplify this process by learning directly from chosen–rejected pairs without a separate reward model. Building on this paradigm, self-rewarding methods (Yuan et al., 2024; Chen et al., 2025a; Wang et al., 2025) use the language model as both generator and evaluator, providing a scalable path to improvement with little external annotation.

Despite its promise, the generate–verify–learn paradigm depends on reliable verification. Prior work improves different stages of this pipeline: self-refinement enriches generated supervision (Madaan et al., 2023); response-level evaluators use scalar scoring, rubric-based judging, pairwise preferences, or generative verification (Zheng et al., 2023; Kim et al., 2024a; Ye et al., 2025c; Zhang et al., 2025a); and methods such as CREAM mitigate noisy self-generated preferences during learning (Wang et al., 2025). Yet many response-level evaluators still assess fine-grained quality through a single global judgment over the complete response. This burdens the model with distinguishing subtle quality differences in one step, an ability that does not necessarily follow from strong generation performance (Stechly et al., 2025). We therefore ask: can self-verification be made easier and more reliable by decomposing a global quality judgment into simpler binary verification decisions?

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.

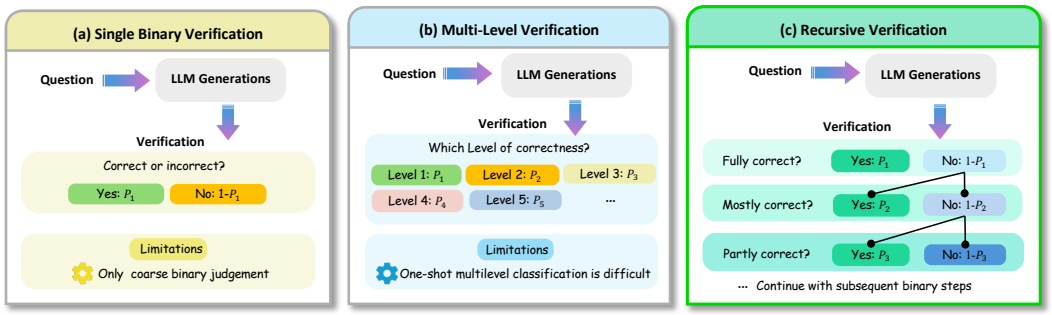

Figure 1: Comparison of self-verification paradigms. (a) Single Binary Verification predicts response correctness, with P_1 denoting the *Yes* probability. (b) Multi-Level Verification directly predicts a quality level, where P_k denotes the probability assigned to level k. (c) Recursive Verification evaluates the response through progressively relaxed binary criteria, where P_k denotes the *Yes* probability at step k and $1 - P_k$ the corresponding *No* probability.

As illustrated in Figure 1, we contrast single binary verification, direct multi-level verification, and our recursive verification paradigm. Our work starts from a simple observation: the difficulty of self-verification depends not only on what a model is asked to evaluate, but also on how the evaluation problem is formulated. Asking a model to directly assign one of several quality levels, or to produce a finely calibrated scalar score, requires it to simultaneously distinguish multiple quality boundaries and calibrate their relative differences in a single prediction. This can be unnecessarily demanding, particularly for smaller models whose generative capabilities may already exceed the reliability of their fine-grained self-assessments. In contrast, deciding whether a response satisfies one explicitly defined criterion provides a relatively simpler decision interface. This motivates us to reformulate fine-grained self-verification as a sequence of binary decisions over progressively relaxed quality criteria. Instead of predicting an absolute multi-class label, the verifier separately evaluates each response against a sequence of progressively relaxed binary quality criteria.

Based on this principle, we introduce Recursive Self-Verification (RSV), a general framework for self-improving language models through hierarchical binary verification. Given a model, we generate multiple candidate responses for each prompt and evaluate each candidate through a configurable sequence of Yes/No verification steps with progressively relaxed quality criteria. At each step, RSV obtains the normalized probabilities of the two verbalizers as step-wise verification confidences and recursively allocates probability mass across ordered quality levels: the strictest step contributes to the highest level, while each relaxed step allocates from the mass remaining after stricter levels. Each quality level is associated with a predefined quality value, and the expected value of the resulting quality distribution is used as the final verification score. These scores identify the preferred and rejected responses and construct preference pairs for DPO training. Since candidate quality is constructed directly from the model's binary verification likelihoods, RSV requires no gold answers, externally supplied correctness labels, or an external reward model. Thus, RSV applies broadly whenever response quality can be verified through ordered binary criteria.

Our main contributions are summarized as follows:

- We introduce a general decomposition principle for fine-grained self-verification, reformulating a difficult multi-way quality judgment as multiple ordered binary verification problems with progressively relaxed quality criteria.
- We develop a probabilistic composition that recursively allocates probability mass from binary verification confidences to form an ordered quality distribution, whose expectation yields a confidence-aware preference score without oracle supervision.
- We integrate RSV with preference optimization and show consistent gains on K&K, TruthQuest, and MATH-Hard across two model families and diverse reasoning settings, with benefits extending from DPO to on-policy GRPO.

2 Related Work

2.1 Preference Optimization

Preference optimization provides an effective way to convert model-generated feedback into learning signals. Direct Preference Optimization (DPO) (Rafailov et al., 2023) optimizes a policy from preferred and rejected response pairs without a separate reward model. This makes DPO particularly suitable for self-improving settings, where preference pairs can derive from the model's generations and evaluations. Accordingly, DPO has been widely adopted in self-rewarding and self-evolving language models (Yuan et al., 2024; Wang et al., 2025; Chen et al., 2025a).

Group Relative Policy Optimization (GRPO) (Shao et al., 2024) provides an alternative on-policy optimization paradigm. Instead of learning from a fixed set of preference pairs, GRPO samples groups of responses from the current policy and optimizes them according to relative reward signals within each group, making it well suited for reasoning-oriented reinforcement learning.

Our work treats the optimization algorithm as complementary to the proposed verification mechanism. RSV focuses on improving how response quality is estimated from self-generated candidates rather than introducing a new optimization objective. We use RSV with standard DPO in our main experiments and further integrate it with on-policy GRPO, demonstrating that recursive self-verification can support different optimization paradigms.

2.2 Self-Evolving Language Models

Recent work has explored enabling language models to improve from their own generations with reduced dependence on external supervision. Early approaches such as STaR (Zelikman et al., 2022) bootstrap reasoning from self-generated rationales. Subsequent methods explore different forms of model-driven improvement, including self-play and self-training (Chen et al., 2024; Singh et al., 2024), self-generated preference supervision (Yuan et al., 2024; Pang et al., 2024; Prasad et al., 2025), bootstrapping from implicit preference rewards (Chen et al., 2025a), and meta-level self-evaluation for improving judgment capabilities (Wu et al., 2025). In particular, Self-Rewarding Language Models (Yuan et al., 2024) use the model itself to generate responses and provide LLM-as-a-Judge rewards for iterative preference optimization. CREAM (Wang et al., 2025) further shows that inaccurate or overconfident self-generated preference labels can degrade the reliability of preference data, motivating explicit mechanisms for improving self-generated supervision.

A closely related line of work studies model-based evaluation and self-verification (Weng et al., 2023; Miao et al., 2024). LLM-as-a-Judge methods evaluate responses through scalar scores, pairwise comparisons, rankings, or rubric-based judgments (Zheng et al., 2023; Kim et al., 2024a;b; Zhu et al., 2025). Other approaches improve verification through process-level supervision and evaluation (Lightman et al., 2024; Zhang et al., 2025b), generative verifiers (Zhang et al., 2025a), contrastive judgment learning (Ye et al., 2025c), or self-feedback and iterative refinement (Madaan et al., 2023). Recent studies further show that even strong LLMs can exhibit substantial self-verification limitations on reasoning and planning tasks (Stechly et al., 2025; Hong et al., 2024).

Many existing approaches address these limitations by improving the verifier through additional supervision (Ye et al., 2025b), specialized training (Hosseini et al., 2024; Yang et al., 2024b; Yu et al., 2025), or stronger evaluation (Li et al., 2024). RSV instead focuses on the *structure of the verification problem itself*. Rather than requiring a single global fine-grained judgment, it decomposes response assessment into a configurable sequence of simpler binary verification decisions and recursively composes their probabilities into a fine-grained quality signal for subsequent self-evolution.

3 METHODOLOGY

In this section, we introduce Recursive Self-Verification (RSV), an approach that improves the verification stage of self-evolving language models by replacing a single fine-grained judgment with a sequence of simpler binary decisions. Figure 2 illustrates the overall generate–verify–learn pipeline for self-evolving language models, which consists of three stages: response generation, recursive self-verification, and preference learning. Specifically, the model first generates multiple candidate responses from each input question. Then, RSV evaluates each response with K-step binary de-

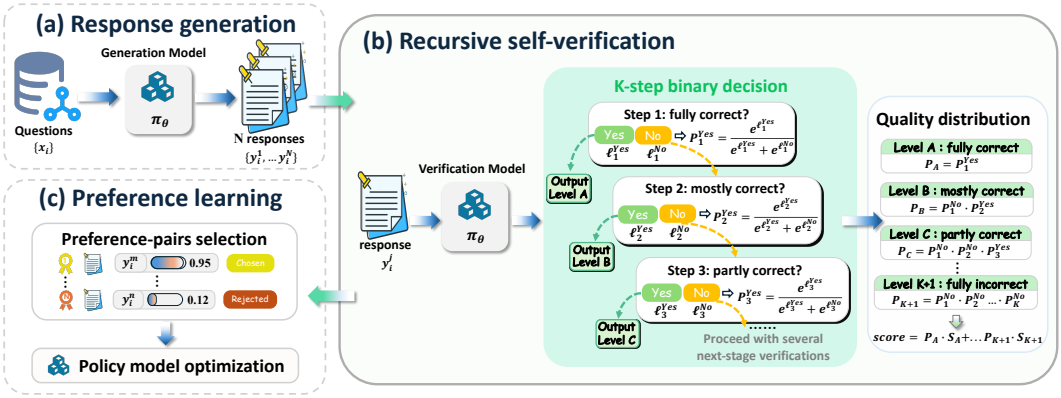

Figure 2: Overview of the RSV-based self-evolution pipeline. (a) Given an input x_i, the policy model π_θ generates N candidate responses $\mathcal{Y}_i = \{y_i^1, \ldots, y_i^N\}$. (b) Each candidate is evaluated through K ordered binary verification steps, whose *Yes* probabilities P_k^{Yes} are recursively composed into an ordered quality distribution. (c) The resulting quality scores rank candidates and select preferred–rejected pairs for subsequent policy optimization.

cisions, which follow a hierarchy of quality criteria ranging from strict to progressively relaxed. After these decisions, RSV converts the decision distribution probabilities into a scalar quality score by recursively combining the probabilities into a distribution over quality levels. These scores are used to construct preference pairs for standard preference optimization. By improving how self-generated responses are evaluated, RSV provides more useful preference supervision for the overall self-evolution pipeline.

3.1 Response Generation

Figure 2(a) illustrates that the current model samples multiple candidate responses for each input. Let π_θ denote the language model used for both response generation and self-verification. Given an input question x_i, we sample N candidate responses using temperature-based sampling with an elevated temperature to promote response diversity, which can be formulated as

$$y_i^n \sim \pi_\theta(\cdot \mid x_i), \qquad n = 1, \ldots, N. \tag{1}$$

As such, the resulted response set $\mathcal{Y}_i = \{y_i^1, \ldots, y_i^N\}$ is prepared for the following quality evaluation and preference-pairs selection in the next two stages.

3.2 Recursive Self-Verification

Figure 2(b) illustrates how each candidate response is evaluated through a sequence of progressively relaxed binary verification criteria. RSV evaluates each candidate using K ordered verification criteria $\mathcal{C}_1, \ldots, \mathcal{C}_K$, ranging from the strictest to increasingly permissive notions of response quality. The criteria are nested such that satisfying a stricter criterion implies satisfying all more permissive ones. RSV uses these ordered criteria to define $K+1$ ordered quality levels $L_1, \ldots, L_{K+1}$ for the subsequent probability-mass allocation, where L_1 denotes the highest quality and L_{K+1} the lowest.

For each criterion $\mathcal{C}_k$, we construct a verification prompt $\phi_k(x, y)$ containing the input, the complete candidate response, and the corresponding criterion. Instead of predicting a fine-grained quality label directly, the verifier answers a binary question using the verbalizers `Yes` and `No`. Let ℓ_k^{Yes} and ℓ_k^{No} denote the log-probabilities of the two verbalizers conditioned on the verification prompt $\phi_k(x, y)$. We normalize only over these two verbalizers:

$$P_k^{\text{Yes}} = \frac{e^{\ell_k^{\text{Yes}}}}{e^{\ell_k^{\text{Yes}}} + e^{\ell_k^{\text{No}}}}, \qquad P_k^{\text{No}} = 1 - P_k^{\text{Yes}}. \tag{2}$$

Importantly, each criterion is evaluated separately on the same candidate response. The hard prediction of one step is never passed to another step. Instead, the recursive structure is introduced when step-wise verification confidences allocate probability mass across the ordered quality levels.

At each step, the normalized Yes confidence determines the fraction of the remaining probability mass allocated to the current quality level, while the complementary mass is retained as residual mass for subsequent lower quality levels. The resulting induced quality distribution is

$$P_{L_k} = \begin{cases} P_1^{\mathrm{Yes}}, & k = 1, \\ \left(\prod_{j=1}^{k-1} P_j^{\mathrm{No}}\right) P_k^{\mathrm{Yes}}, & 2 \le k \le K, \\ \prod_{j=1}^{K} P_j^{\mathrm{No}}, & k = K+1. \end{cases} \tag{3}$$

Thus, each response is associated with $\mathbf{P}(x, y) = [P_{L_1}, \ldots, P_{L_{K+1}}]$, where $P_{L_k} \ge 0$ and $\sum_k P_{L_k} = 1$.

This formulation preserves uncertainty throughout the verification process rather than committing to an early hard decision. Moreover, the number of verification steps is flexible: changing K adjusts the granularity of the induced response-quality hierarchy without changing the underlying formulation.

After obtaining the recursive quality distribution for each candidate response, we associate the ordered quality levels with scalar values $S_{L_1} > S_{L_2} > \cdots > S_{L_{K+1}}$ and define the RSV score as the expected quality under the induced distribution:

$$S_{\mathrm{RSV}}(x, y) = \sum_{k=1}^{K+1} S_{L_k} P_{L_k}. \tag{4}$$

Unlike assigning each response to its most probable quality level, this expectation retains information from the full quality distribution. Consequently, responses with the same most likely level can still receive different scores according to how probability mass is distributed across all levels.

3.3 Preference Learning

RSV produces a response-level quality score for each self-generated candidate, making preference-based optimization a natural downstream learning objective. We adopt Direct Preference Optimization (DPO) (Rafailov et al., 2023), which directly optimizes the policy from preferred and rejected response pairs without requiring a separately trained reward model. This provides a clean interface with RSV: the recursive verification module is responsible for constructing reliable preference supervision, while the downstream optimization objective remains unchanged. Such a formulation is also consistent with recent self-rewarding and self-evolving language models that learn from model-generated preference data (Yuan et al., 2024; Chen et al., 2025a; Wang et al., 2025).

Figure 2(c) illustrates how the response-level scores are converted into preference pairs for subsequent preference optimization. For each input x_i, we independently score all candidate responses in $\mathcal{Y}_i$ and construct a preference pair by selecting the highest-scoring response as preferred and the lowest-scoring response as rejected:

$$\begin{aligned} y_i^+ &= \arg\max_{y \in \mathcal{Y}_i} S_{\mathrm{RSV}}(x_i, y), \\ y_i^- &= \arg\min_{y \in \mathcal{Y}_i} S_{\mathrm{RSV}}(x_i, y). \end{aligned} \tag{5}$$

Repeating this procedure over all inputs yields the self-generated preference dataset

$$\mathcal{D}_{\mathrm{pref}} = \left\{(x_i, y_i^+, y_i^-)\right\}_{i=1}^{M}. \tag{6}$$

Importantly, RSV evaluates each response independently before introducing any pairwise preference relation. Thus, the preferred and rejected responses are determined only after response-level quality estimation, rather than through direct pairwise comparison during verification.

We then use $\mathcal{D}_{\mathrm{pref}}$ to optimize the policy with DPO. Let π_θ denote the trainable policy and π_{ref} the fixed reference policy, and define the reference-adjusted log-probability as

$$g_\theta(x, y) = \log \pi_\theta(y \mid x) - \log \pi_{\mathrm{ref}}(y \mid x). \tag{7}$$

The DPO objective is then

$$\mathcal{L}_{\mathrm{DPO}} = -\mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}_{\mathrm{pref}}}\left[\log\sigma\left(\beta\left[g_\theta(x,y^+) - g_\theta(x,y^-)\right]\right)\right], \tag{8}$$

where $\sigma(\cdot)$ denotes the sigmoid function and β controls the strength of preference optimization relative to the reference policy.

During preference-data construction, the self-verifier is kept fixed. Once $\mathcal{D}_{\mathrm{pref}}$ is constructed, DPO is applied using its standard optimization objective without further modification. Therefore, RSV changes *how preference supervision is constructed*: fine-grained response quality is obtained through recursive binary self-verification, converted into response-level scores, and only then transformed into preference pairs for standard DPO optimization.

4 EXPERIMENTS

We evaluate Recursive Self-Verification (RSV) on three benchmarks using three models from two model families. Our main experiments examine whether recursive binary verification improves self-generated preference supervision across model scales and reasoning tasks, while subsequent diagnostic analyses focus on Qwen3-1.7B and Gemma-3-4B-IT. Unless otherwise specified, we use one round of preference construction followed by DPO training.

4.1 SETTINGS

Datasets and Metrics. We evaluate on **Knights-and-Knaves (K&K)**, **TruthQuest**, and **MATH-Hard**. K&K (Xie et al., 2025) contains automatically generated Knights-and-Knaves puzzles; we group instances into 2–3, 4–5, and 6–8 people. TruthQuest (Mondorf & Plank, 2024) evaluates suppositional reasoning over truthfulness; we generate **Set S** with its released generator and group instances into 3–4, 5–6, and 7–8 people. Larger groups generally involve more possible assignments and longer logical dependencies. **MATH-Hard** is the Level-5 subset of MATH (Hendrycks et al., 2021), covering challenging problems across seven mathematical categories, including algebra, geometry, number theory, and probability. We report **accuracy (%)** based on the exact final assignment for K&K and TruthQuest, and the final mathematical answer for MATH-Hard.

Models, Baselines, and Training. We evaluate three open-weight models of different scales: **Qwen3-1.7B** (Qwen Team, 2025), **Gemma-3-4B-IT** (Gemma Team, 2025), and **Qwen2.5-7B-Instruct** (Yang et al., 2024a). Qwen3-1.7B is used in non-thinking mode throughout. For each backbone, the same frozen model is used for candidate generation and self-verification during preference-data construction, without any stronger external judge. For K&K and TruthQuest, we train on the 2–3- and 3–4-person splits, respectively, and evaluate all reported difficulty groups. For MATH-Hard, we independently sample 20% from each problem category in the original training and test splits, pooling the sampled categories within each split to form mixed training and evaluation sets. We compare RSV with the **Base** model, **Self-Rewarding** (Yuan et al., 2024), **SimpleSE** (Qi et al., 2026), **GenRM** (Zhang et al., 2025a), and **IPO** (Garg et al., 2025). For SimpleSE, we follow Qi et al. (2026) with thresholds of 0.3 and 0.7. Let L denote the number of quality levels; we evaluate $\mathrm{RSV}(L)$ with $L \in \{3, 4, 5\}$ and $K = L - 1$ binary criteria. **Multi-Level Verification (MLV)**, denoted $\mathrm{MLV}(L)$, directly predicts one of the same L levels in a single multiclass judgment. For each prompt, the highest- and lowest-scoring candidates form a preference pair for one round of DPO. Each trained model is evaluated with four random seeds, and we report mean accuracy across runs.

4.2 RESULTS

Overall Results. Table 1 summarizes the main results. RSV ($L = 5$) achieves the best performance across all reported model, benchmark, and difficulty settings. For Qwen3-1.7B, it reaches **63.1%** average accuracy on K&K and **57.4%** on TruthQuest, compared with **54.8%** and **49.5%** for the strongest baselines, respectively. On MATH-Hard, RSV ($L = 5$) achieves **45.3%**, **55.2%**, and **46.3%** accuracy on Qwen3-1.7B, Gemma-3-4B-IT, and Qwen2.5-7B-Instruct, respectively, consistently outperforming the strongest baselines. Moreover, RSV with $L = 3$ and $L = 4$ outperforms the strongest baselines in nearly all reported settings. Compared with Self-Rewarding and threshold-based SimpleSE, RSV retains step-wise binary verification confidences and recursively composes

Table 1: Main results on K&K, TruthQuest, and MATH-Hard. All values are accuracy (%). For RSV, L denotes the number of ordered quality levels. For each exact number of people, the test set contains 100 examples. Avg. denotes the accuracy over all test examples in the corresponding benchmark, i.e., an example-count-weighted average across the reported difficulty groups. The best result for each model and evaluation setting is shown in bold.

		K&K				TruthQuest				MATH-Hard
Model	**Method**	**2–3**	**4–5**	**6–8**	**Avg.**	**3–4**	**5–6**	**7–8**	**Avg.**	**Acc.**
Qwen3-1.7B	Base	57.0	36.1	19.5	35.0	35.8	23.9	11.4	23.7	39.1
	Self-Rewarding	71.0	51.2	31.8	48.5	63.8	51.0	33.8	49.5	43.0
	SimpleSE	58.5	37.5	21.6	36.7	37.1	24.3	11.6	24.3	41.7
	GenRM	71.3	45.1	11.4	38.1	43.5	28.9	12.0	28.1	40.0
	IPO	76.0	62.6	35.5	54.8	64.0	50.6	30.6	48.4	42.2
	RSV ($L = 3$)	83.5	68.0	39.1	60.0	72.1	52.8	35.7	53.5	43.1
	RSV ($L = 4$)	86.0	69.9	40.6	61.9	74.3	54.1	38.6	55.7	44.4
	RSV ($\mathbf{L = 5}$)	**87.3**	**71.2**	**41.5**	**63.1**	**76.1**	**55.9**	**40.1**	**57.4**	**45.3**
Gemma-3-4B	Base	66.7	30.8	8.7	31.6	54.1	25.0	11.7	30.3	50.5
	Self-Rewarding	73.3	37.5	15.3	38.2	55.8	28.3	11.9	32.0	53.6
	SimpleSE	72.9	32.0	13.6	35.8	56.1	27.4	15.8	33.1	52.9
	GenRM	67.3	28.9	11.8	32.5	54.4	25.6	13.8	31.3	52.2
	IPO	72.3	36.9	12.8	36.7	58.5	28.8	14.1	33.8	53.3
	RSV ($L = 3$)	74.6	39.7	15.2	39.2	59.7	30.3	15.2	35.1	54.2
	RSV ($L = 4$)	77.1	41.3	16.9	41.1	61.7	31.2	16.0	36.3	54.6
	RSV ($\mathbf{L = 5}$)	**78.3**	**42.2**	**18.1**	**42.2**	**62.4**	**32.5**	**16.9**	**37.3**	**55.2**
Qwen2.5-7B Instruct	Base	42.5	16.5	5.2	19.1	29.2	13.0	5.5	15.9	39.9
	Self-Rewarding	41.9	16.6	5.6	19.1	33.6	14.5	5.9	18.0	44.9
	SimpleSE	43.2	16.8	5.5	19.5	30.3	13.3	5.3	16.3	42.7
	GenRM	42.7	16.4	4.3	18.7	29.8	11.5	6.0	15.8	41.2
	IPO	45.3	18.3	5.8	20.7	30.5	13.2	6.3	16.7	43.4
	RSV ($L = 3$)	48.6	19.5	6.1	22.1	33.1	14.3	6.4	17.9	45.6
	RSV ($L = 4$)	50.2	20.6	6.3	22.9	34.2	14.7	6.8	18.6	45.9
	RSV ($\mathbf{L = 5}$)	**52.4**	**21.3**	**6.9**	**24.0**	**34.9**	**15.2**	**7.0**	**19.0**	**46.3**

them into an ordered quality distribution, yielding a finer-grained preference signal. The results on Gemma-3-4B-IT and Qwen2.5-7B-Instruct show the same overall trend, as detailed in Table 1. Overall, RSV provides consistent gains across model scales, benchmarks, and reasoning tasks.

Effect of Recursive Verification Granularity. Table 1 shows that increasing the number of ordered quality levels consistently improves RSV across all models and benchmarks. For Qwen3-1.7B, the average accuracy increases from 60.0%/53.5% with $L = 3$ to 61.9%/55.7% with $L = 4$ and 63.1%/57.4% with $L = 5$ on K&K/TruthQuest, respectively. A similar monotonic trend is observed for Gemma-3-4B-IT, where the corresponding averages improve from 39.2%/35.1% to 42.2%/37.3%, and for Qwen2.5-7B-Instruct, from 22.1%/17.9% to 24.0%/19.0%. MATH-Hard shows the same monotonic improvement across all three backbones. These results indicate that finer recursive verification granularity generally provides more informative quality estimates and leads to stronger self-evolution performance.

Effect of Recursive Verification Decomposition. To examine verification decomposition at matched granularity, Figure 3 compares RSV and MLV for $L \in \{3, 4, 5\}$. MLV predicts probabilities over L levels in one multiclass judgment and, like RSV, computes its score as the probability-weighted sum of level scores. Across Qwen3-1.7B and Gemma-3-4B-IT, RSV achieves higher accuracy at every L and difficulty. For Qwen3-1.7B, RSV ($L = 3$) obtains 83.5%/68.0%/39.1%, versus 50.1%/35.8%/21.6% for MLV. At $L = 5$, MLV reaches 72.0%/55.1%/31.7%, while RSV achieves **87.3%/71.2%/41.5%**. Results suggest that recursive verification is substantially more effective.

Scaling with Inference Passes. Figure 4 compares RSV ($L = 5$) with the Qwen3-1.7B base model as the number of inference passes increases. RSV consistently achieves higher accuracy across tested budgets and difficulty levels, while both methods benefit from additional inference.

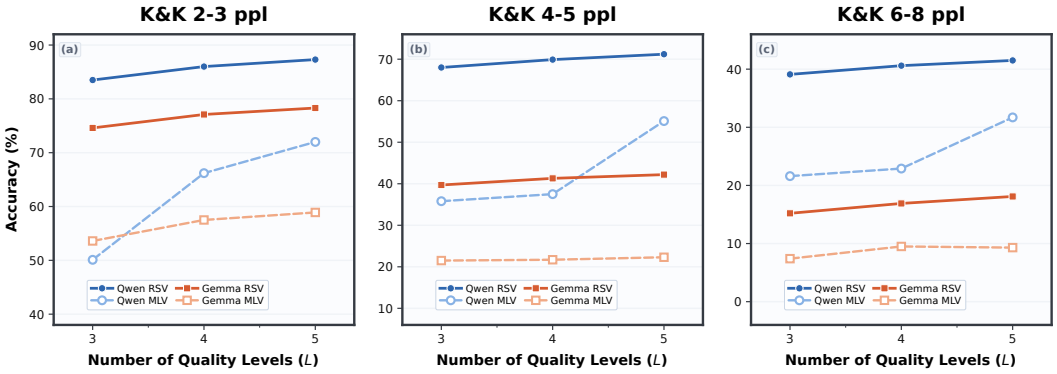

Figure 3: Effect of the number of quality levels on K&K. We compare RSV and Multi-Level Verification (MLV) with $L \in \{3, 4, 5\}$ using Qwen3-1.7B and Gemma-3-4B-IT across three difficulty splits.

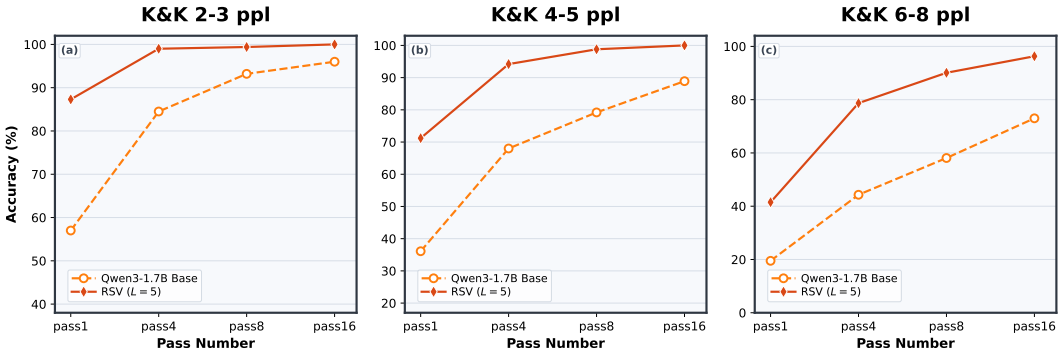

Figure 4: Performance across different numbers of passes on K&K using Qwen3-1.7B. We compare the Base model with RSV ($L = 5$) using 1, 4, 8, and 16 passes across three difficulty splits. For pass-N, each problem is attempted N times and counted correct if any attempt is correct.

Notably, with four passes, RSV reaches **99.0%, 94.2%, and 78.7%** on the three difficulty splits, already surpassing the base model with sixteen passes (96.0%, 88.9%, and 73.0%). With sixteen passes, RSV further reaches **100%, 100%, and 96.3%**, showing RSV substantially raises the likelihood of sampling correct trajectories under repeated inference, particularly on challenging instances.

Why Does Recursive Verification Help? To analyze RSV, Table 2 compares matched-granularity RSV and MLV statistics on K&K for two models, covering preference quality, selection quality, within-prompt ranking, and length dependence. On **Gemma-3-4B-IT**, RSV reduces incorrect-over-correct pairs from 22.83% to 19.17% ($p = 0.0018$), while increasing Correct in top-4 from 76.50% to 83.58% and chosen assignment accuracy from 69.99% to 73.65%. On **Qwen3-1.7B**, preference precision improves from 70.13% to 74.12%, while Correct-at-top increases from 55.11% to 57.89%. RSV shows weaker length dependence on Qwen, with absolute character-length correlation decreasing from 0.3612 to 0.0583. These results suggest recursive decomposition yields cleaner preference supervision, more reliable candidate ordering, and less reliance on superficial length cues, strengthening self-evolution. Further analysis of RSV and MLV score distributions appears in Appendix D.3.

Results with On-Policy Optimization. To examine whether RSV generalizes to on-policy optimization, we evaluate it with GRPO on K&K and MATH-Hard using **Qwen3-1.7B**, with results reported in Table 3. On K&K, RSV ($L = 3$) achieves **82.6%, 64.7%, and 39.8%** on the 2–3, 4–5, and 6–8 people splits, respectively, with **59.1%** overall accuracy. It outperforms IPO by **4.5, 2.8, and**

Table 2: Diagnostic comparison of matched-granularity **MLV** and **RSV** on K&K ($L = 3$ for Qwen3-1.7B; $L = 4$ for Gemma-3-4B-IT). Preference precision measures correctly oriented informative pairs; Correct-at-top is top-1 accuracy when a correct candidate exists; Both-wrong quality gap compares assignment quality when both responses are incorrect. $|\rho|$ denotes absolute within-prompt Spearman correlation with response length.

	Qwen3-1.7B ($L = 3$)			**Gemma-3-4B-IT** ($L = 4$)		
Metric	**MLV**	**RSV**	Δ	**MLV**	**RSV**	Δ
Preference quality						
Incorrect-over-correct (%) ↓	11.75	**9.75**	−2.00	22.83	**19.17**	−3.66*
Chosen accuracy (%) ↑	53.00	**55.67**	+2.67	51.58	**54.17**	+2.59
Preference precision (%) ↑	70.13	**74.12**	+3.99	58.55	**60.75**	+2.20
Selection quality						
Correct-at-top (%) ↑	55.11	**57.89**	+2.78	52.37	**54.99**	+2.62
Correctness gap ↑	0.1583	**0.1817**	+0.0234	0.0942	**0.1050**	+0.0108
Chosen assignment acc. (%) ↑	72.21	**73.88**	+1.67	69.99	**73.65**	+3.66
Both-wrong quality gap ↑	0.0051	**0.0173**	+0.0122	−0.0162	−0.0013	+0.0149
Within-prompt ranking						
MRR ↑	0.6881	**0.7100**	+0.0219	0.6530	**0.6898**	+0.0368
Correct in top-4 (%) ↑	80.58	**81.83**	+1.25	76.50	**83.58**	+7.08
NDCG@4 ↑	0.5509	**0.5601**	+0.0092	0.5527	**0.5778**	+0.0251
Pairwise ranking acc. ↑	0.5878	**0.5927**	+0.0049	0.5421	**0.5564**	+0.0143
Length dependence						
Char-length corr. $\lvert\rho\rvert$ ↓	0.3612	**0.0583**	−0.3029	0.4967	**0.2772**	−0.2195
Word-length corr. $\lvert\rho\rvert$ ↓	0.3630	**0.0676**	−0.2954	0.5057	**0.2612**	−0.2445

*Paired sign-flip test, $p = 0.0018$.

Table 3: On-policy GRPO accuracy on K&K and MATH-Hard with Qwen3-1.7B.

Method	**K&K**				**MATH-Hard**
	2–3 ppl	**4–5 ppl**	**6–8 ppl**	**Avg.**	**Acc.**
Base	57.0	36.1	19.5	35.0	39.1
SimpleSE	67.0	52.5	27.6	46.0	43.0
IPO	78.1	61.9	32.2	53.8	43.8
GenRM	70.1	47.2	20.7	42.4	41.2
MLV ($L = 3$)	55.4	36.3	22.1	35.7	40.5
RSV ($L = 3$)	**82.6**	**64.7**	**39.8**	**59.1**	**45.5**

7.6 points across the splits and matched-granularity MLV ($L = 3$), which obtains 55.4%, 36.3%, and 22.1%. On MATH-Hard, RSV reaches **45.5%**, exceeding IPO (43.8%) and MLV (40.5%). These results suggest that RSV benefits both objectives: DPO uses only the two score extremes, whereas GRPO exploits verification scores from all sampled responses through relative rewards.

5 Conclusion

We introduced Recursive Self-Verification (RSV), which reformulates fine-grained self-verification into ordered binary verification problems and recursively allocates probability mass using their step-wise verification confidences, inducing an ordered quality distribution for preference construction. Experiments on K&K, TruthQuest, and MATH-Hard show consistent improvements across model families and reasoning tasks. Matched-granularity comparisons further indicate that these gains are not attributable to finer quality granularity alone, while verification-stage diagnostics provide evidence of more reliable within-prompt candidate ordering and fewer harmful preference reversals. Results with on-policy GRPO additionally show that the benefits of RSV extend beyond DPO. Overall, our findings highlight verification structure as an important design choice for constructing reliable self-generated supervision in self-evolving language models.

AI USE STATEMENT

In this work, we used generative AI tools for assisting with literature search and locating potentially relevant prior work. We have not used generative AI tools for other substantive research tasks, and the other listed AI-use categories are not applicable to this work. Additionally, we used generative AI tools for language editing and polishing of the manuscript, including improving grammar, wording, clarity, readability, and LaTeX formatting. We have reviewed all AI-assisted work. References located with AI assistance were checked against the original sources, and all AI-assisted language edits were reviewed before inclusion in the manuscript. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies self-verification and self-evolution of language models using existing reasoning benchmarks and open-weight models. It does not involve human subjects or personally identifiable information. We follow standard research practices in reporting experimental results and model behavior.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the RSV formulation, datasets and evaluation protocols, model and training settings, and principal generation, verification, and optimization hyperparameters in the main paper and Appendix. The Appendix also includes complete theoretical derivations and proofs of the stated properties, together with additional experimental analyses. We additionally provide the implementation code in the supplementary material to facilitate reproduction and verification of our results.

REFERENCES

Changyu Chen, Zichen Liu, Chao Du, Tianyu Pang, Qian Liu, Arunesh Sinha, Pradeep Varakantham, and Min Lin. Bootstrapping language models with DPO implicit rewards. In *International Conference on Learning Representations*, 2025a.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Ziye Chen, Chengwei Qin, and Yao Shu. RIMO: An easy-to-evaluate, hard-to-solve olympiad benchmark for advanced mathematical reasoning. *arXiv preprint arXiv:2509.07711*, 2025b.

Shivank Garg, Ayush Singh, Shweta Singh, and Paras Chopra. IPO: Your language model is secretly a preference classifier. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025.

Gemma Team. Gemma 3 Technical Report. *arXiv preprint arXiv:2503.19786*, 2025.

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. *NeurIPS*, 2021.

Ruixin Hong, Hongming Zhang, Xinyu Pang, Dong Yu, and Changshui Zhang. A closer look at the self-verification abilities of large language models in logical reasoning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-STaR: Training verifiers for self-taught reasoners. In *Proceedings of the 2024 Conference on Language Modeling*, 2024.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. In *International Conference on Learning Representations*, 2024a.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024b.

Lei Li, Yekun Chai, Shuohuan Wang, Yu Sun, Hao Tian, Ningyu Zhang, and Hua Wu. Tool-augmented reward modeling. 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.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023.

Mathematical Association of America. American invitational mathematics examination (AIME). MAA American Mathematics Competitions, 2026.

Ning Miao, Yee Whye Teh, and Tom Rainforth. SelfCheck: Using LLMs to zero-shot check their own step-by-step reasoning. In *International Conference on Learning Representations*, 2024.

Philipp Mondorf and Barbara Plank. Liar, liar, logical mire: A benchmark for suppositional reasoning in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization. In *Advances in Neural Information Processing Systems*, 2024.

Archiki Prasad, Weizhe Yuan, Richard Yuanzhe Pang, Jing Xu, Maryam Fazel-Zarandi, Mohit Bansal, Sainbayar Sukhbaatar, Jason E. Weston, and Jane Yu. Self-consistency preference optimization. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Zhenting Qi, Susanna Maria Baby, Stefanie Anna Baby, Kan Yuan, Andrew Tomkins, Tu Vu, Da-Cheng Juan, and Cyrus Rashtchian. On the generalization gap in self-evolving language model reasoning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Qwen Team. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshitiej Mahajan, Laura Culp, Lechao Xiao, Maxwell L. Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yundi Qian, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024.

Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. On the self-verification limitations of large language models on reasoning and planning tasks. In *International Conference on Learning Representations*, 2025.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-Instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.

Zhaoyang Wang, Weilei He, Zhiyuan Liang, Xuchao Zhang, Chetan Bansal, Ying Wei, Weitong Zhang, and Huaxiu Yao. CREAM: Consistency regularized self-rewarding language models. In *International Conference on Learning Representations*, 2025.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason E. Weston, and Sainbayar Sukhbaatar. Meta-rewarding language models: Self-improving alignment with LLM-as-a-meta-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025.

Chulin Xie, Yangsibo Huang, Chiyuan Zhang, Da Yu, Xinyun Chen, Bill Yuchen Lin, Bo Li, Badih Ghazi, and Ravi Kumar. On memorization of large language models in logical reasoning. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024a.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. Regularizing hidden states enables learning generalizable reward model for LLMs. In *Advances in Neural Information Processing Systems*, 2024b.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. LIMO: Less is more for reasoning. In *Second Conference on Language Modeling*, 2025a.

Zihuiwen Ye, Fraser David Greenlee, Max Bartolo, Phil Blunsom, Jon Ander Campos, and Matthias Gallé. Improving reward models with synthetic critiques. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025b.

Ziyi Ye, Xiangsheng Li, Qiuchi Li, Qingyao Ai, Yujia Zhou, Wei Shen, Dong Yan, and Yiqun Liu. Learning LLM-as-a-judge for preference alignment. In *International Conference on Learning Representations*, 2025c.

Yue Yu, Zhengxing Chen, Aston Zhang, Liang Tan, Chenguang Zhu, Richard Yuanzhe Pang, Yundi Qian, Xuewei Wang, Suchin Gururangan, Chao Zhang, Melanie Kambadur, Dhruv Mahajan, and Rui Hou. Self-generated critiques boost reward modeling for language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2025.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E. Weston. Self-rewarding language models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, 2022.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Seyed Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction. In *International Conference on Learning Representations*, 2025a.

Shimao Zhang, Xiao Liu, Xin Zhang, Junxiao Liu, Zheheng Luo, Shujian Huang, and Yeyun Gong. Process-based self-rewarding language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025b.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, 2023.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. JudgeLM: Fine-tuned large language models are scalable judges. In *International Conference on Learning Representations*, 2025.

A THEORETICAL PROPERTIES OF RECURSIVE SELF-VERIFICATION

We provide a compact analysis of the properties central to Recursive Self-Verification (RSV). We first formalize the relation between progressively relaxed binary criteria and ordered quality levels. We then introduce the residual probability-mass allocation rule and show that it is consistent with exact-level classification under ideal hard verification, always defines a valid quality distribution, and can represent any ordered categorical distribution. Finally, we analyze the resulting RSV score, establish its boundedness, cumulative representation, and monotonic progressive influence, and justify expected quality as a scalar summary.

Throughout, all K verification criteria are evaluated separately on the same candidate response. The recursive structure is introduced only when their soft Yes/No confidences are subsequently composed into probability mass over ordered quality levels.

A.1 ORDERED CRITERIA AND QUALITY LEVELS

For a fixed input, let Ω denote the space of possible candidate responses. We consider $K+1$ ordered quality levels

$$L_1 \succ L_2 \succ \cdots \succ L_{K+1},$$

where L_1 denotes the highest quality and L_{K+1} the lowest. For each level L_k, let $\mathcal{B}_k \subseteq \Omega$ be the set of responses whose exact semantic quality is L_k. These exact-level regions are mutually exclusive and collectively exhaustive:

$$\mathcal{B}_i \cap \mathcal{B}_j = \varnothing \quad (i \neq j), \qquad \bigcup_{k=1}^{K+1} \mathcal{B}_k = \Omega.$$

Thus, each response belongs to exactly one quality-level region $\mathcal{B}_k$.

RSV does not ask the verifier to directly identify this exact level. Instead, it uses K cumulative binary criteria $\mathcal{C}_1, \ldots, \mathcal{C}_K$, ordered from strict to progressively relaxed. For criterion $\mathcal{C}_k$, define

$$\mathcal{A}_k := \{y \in \Omega : y \text{ satisfies } \mathcal{C}_k\}.$$

Criterion $\mathcal{C}_k$ asks whether a response reaches L_k or any higher-quality level, so

$$\mathcal{A}_k = \bigcup_{i=1}^{k} \mathcal{B}_i, \qquad \mathcal{A}_1 \subseteq \mathcal{A}_2 \subseteq \cdots \subseteq \mathcal{A}_K.$$

Hence, the criterion regions are nested: any response satisfying a stricter criterion necessarily satisfies every more permissive criterion.

Conversely, the mutually exclusive exact-level regions can be recovered from the nested criterion regions:

$$\mathcal{B}_1 = \mathcal{A}_1, \qquad \mathcal{B}_k = \mathcal{A}_k \setminus \mathcal{A}_{k-1} \quad (2 \leq k \leq K), \qquad \mathcal{B}_{K+1} = \Omega \setminus \mathcal{A}_K.$$

Thus, cumulative binary criteria and mutually exclusive quality levels describe the same ordered semantic structure from two perspectives: $\mathcal{C}_k$ specifies a quality threshold, whereas $\mathcal{B}_k$ specifies an exact quality region.

Importantly, each criterion induces only a single binary partition:

$$\Omega = \underbrace{\mathcal{A}_k}_{\text{reaches } L_k \text{ or better}} \cup \underbrace{(\Omega \setminus \mathcal{A}_k)}_{\text{below } L_k},$$

where

$$\mathcal{A}_k = \bigcup_{i=1}^{k} \mathcal{B}_i, \qquad \Omega \setminus \mathcal{A}_k = \bigcup_{i=k+1}^{K+1} \mathcal{B}_i.$$

Accordingly, verification step k only needs to determine on which side of the quality boundary associated with L_k a response lies, rather than identifying its exact quality level. All responses at L_k or any higher level are grouped together on the positive side of this binary decision.

For example, suppose

$$L_1 = \text{fully correct}, \quad L_2 = \text{mostly correct}, \quad L_3 = \text{partially correct}, \quad L_4 = \text{incorrect}.$$

The criterion associated with L_3 asks only whether a response is *at least partially correct*. A partially correct, mostly correct, and fully correct response all satisfy this criterion; the verifier need not distinguish among these three exact levels at this step. More generally, the nested construction isolates one ordered quality boundary in each verification query, rather than requiring all quality boundaries to be resolved jointly in a single multi-level prediction.

For later reference, define the ideal hard satisfaction indicator

$$c_k(y) := \mathbf{1}\{y \in \mathcal{A}_k\}.$$

If $y \in \mathcal{B}_r$ for some $r \le K$, then

$$c_k(y) = \begin{cases} 0, & k < r, \\ 1, & k \ge r, \end{cases}$$

whereas if $y \in \mathcal{B}_{K+1}$, then $c_k(y) = 0$ for all $k = 1, \dots, K$. Thus, exact quality levels correspond to cumulative hard criterion-satisfaction patterns.

A.2 Recursive Probability-Mass Allocation

For criterion $\mathcal{C}_k$, let

$$q_k := P_k^{\text{Yes}} \in [0,1], \qquad 1 - q_k := P_k^{\text{No}},$$

where q_k is the separately elicited soft Yes confidence for the k-th criterion.

RSV recursively allocates a unit probability mass by setting $R_0 := 1$ and, for $k = 1, \dots, K$,

$$P_{L_k} := R_{k-1} q_k, \qquad R_k := R_{k-1}(1 - q_k). \tag{9}$$

Equivalently,

$$P_{L_k} = q_k \prod_{j<k} (1 - q_j), \quad k = 1, \dots, K, \qquad P_{L_{K+1}} = \prod_{j=1}^{K} (1 - q_j). \tag{10}$$

Here and below, an empty product is defined as 1. These products define the RSV composition rule itself and do not assume statistical independence among the semantic criterion events.

Proposition 1 (Consistency with exact-level classification under ideal hard verification). Suppose a response y has exact semantic quality L_r and let $q_k = c_k(y)$ be the ideal hard outcome of criterion $\mathcal{C}_k$. Then

$$P_{L_r} = 1, \qquad P_{L_k} = 0 \quad (k \ne r).$$

Hence, under ideal hard verification, RSV recovers exactly the same one-hot quality-level assignment as direct exact-level classification.

Proof. If $r \le K$, then $q_k = 0$ for $k < r$ and $q_k = 1$ for $k \ge r$. Therefore,

$$P_{L_k} = 0 \quad (k < r), \qquad P_{L_r} = q_r \prod_{j<r} (1 - q_j) = 1.$$

For every $r < k \le K$, the product defining P_{L_k} contains the factor $1 - q_r = 0$, and hence

$$P_{L_k} = 0 \qquad (r < k \le K).$$

Similarly,

$$P_{L_{K+1}} = \prod_{j=1}^{K} (1 - q_j) = 0,$$

since this product also contains the factor $1 - q_r = 0$.

If $r = K + 1$, then $q_k = 0$ for all k, so

$$P_{L_k} = 0 \qquad (k \le K),$$

and

$$P_{L_{K+1}} = \prod_{j=1}^{K} (1 - q_j) = 1.$$

Thus, RSV recovers the exact one-hot level assignment in all cases. *Q.E.D.*

Corollary (Order preservation under ideal verification). Let $y_a \in \mathcal{B}_r$ and $y_b \in \mathcal{B}_s$ with $r < s$, so that y_a has higher semantic quality than y_b. Under ideal hard verification,

$$S_{\mathrm{RSV}}(y_a) = S_{L_r}, \qquad S_{\mathrm{RSV}}(y_b) = S_{L_s}.$$

Since $S_{L_1} > \cdots > S_{L_{K+1}}$, we have

$$S_{\mathrm{RSV}}(y_a) > S_{\mathrm{RSV}}(y_b).$$

Therefore, under ideal verification, RSV preserves the semantic ordering of response quality.

This result shows that the cumulative criterion representation and the mutually exclusive quality-level representation are semantically consistent in the ideal hard-verification limit. RSV therefore does not change the underlying target notion of quality; its distinction from direct multi-level verification lies in how quality information is elicited and subsequently composed under imperfect self-verification.

Importantly, the criterion confidence q_k and the induced level mass P_{L_k} have different semantics. The quantity q_k represents soft evidence that a response reaches L_k or better, whereas P_{L_k} is the residual probability mass assigned specifically to exact level L_k after stricter levels have been accounted for. Therefore, a higher-quality response need not receive a larger P_{L_k} at every lower-quality level, even if it receives an equal or larger Yes confidence for the corresponding relaxed criterion.

For example, under ideal hard verification, an L_1 response satisfies the criterion associated with L_2, so $q_2 = 1$; but because $q_1 = 1$, its mass has already been assigned to L_1, giving

$$P_{L_2} = (1 - q_1)q_2 = 0.$$

By contrast, an exact L_2 response has $q_1 = 0$ and $q_2 = 1$, hence

$$P_{L_2} = (1 - q_1)q_2 = 1.$$

Thus, cumulative criterion satisfaction and mutually exclusive level assignment are fully compatible under ideal hard verification.

For soft model-produced confidences, however, P_{L_k} should be interpreted as an induced level mass under the RSV composition rule, rather than as a calibrated posterior probability that the response semantically belongs exactly to L_k. RSV does not require these exact-level masses to be coordinate-wise calibrated across responses; its downstream use relies on the ordering induced by the complete quality distribution and its resulting scalar score.

Proposition 2 (Validity of residual allocation). For every $\mathbf{q} = (q_1, \ldots, q_K) \in [0, 1]^K$, Eq. equation 10 defines a valid probability distribution:

$$P_{L_k} \geq 0, \qquad \sum_{k=1}^{K+1} P_{L_k} = 1.$$

Proof. Nonnegativity follows from $q_k, 1 - q_k \in [0, 1]$. From Eq. equation 9,

$$P_{L_k} = R_{k-1}q_k = R_{k-1} - R_k.$$

Hence the total mass telescopes:

$$\sum_{k=1}^{K+1} P_{L_k} = \sum_{k=1}^{K}(R_{k-1} - R_k) + R_K = R_0 = 1.$$

Thus, the recursively allocated masses form a valid probability distribution. *Q.E.D.*

This property is important because separately elicited confidences need not obey the monotonic ordering

$$q_1 \leq q_2 \leq \cdots \leq q_K$$

that exact probabilities of nested semantic events would require. Residual allocation nevertheless yields a valid level distribution for arbitrary verifier confidences.

Proposition 3 (Residual form parameterizes any ordered categorical distribution). Let $p_1, \ldots, p_{K+1}$ be any categorical distribution satisfying

$$p_k \geq 0, \qquad \sum_{k=1}^{K+1} p_k = 1.$$

Define the remaining target mass before level k as

$$R^*_{k-1} := 1 - \sum_{j<k} p_j.$$

Whenever $R^*_{k-1} > 0$, define

$$\widetilde{q}_k := \frac{p_k}{R^*_{k-1}}. \tag{11}$$

Then $\widetilde{q}_k \in [0, 1]$, and the recursive allocation in Eq. equation 10 exactly recovers $P_{L_k} = p_k$ for all k. If $R^*_{k-1} = 0$, all remaining target masses are zero and the corresponding remaining $\widetilde{q}_k$ may be chosen arbitrarily.

Proof. Since

$$p_k \leq 1 - \sum_{j<k} p_j = R^*_{k-1},$$

we have $\widetilde{q}_k \in [0, 1]$. Moreover,

$$R^*_{k-1}\widetilde{q}_k = p_k, \qquad R^*_k = R^*_{k-1} - p_k = R^*_{k-1}(1 - \widetilde{q}_k).$$

Thus, the target residual masses obey the same recursion as Eq. equation 9, yielding

$$P_{L_k} = p_k \quad (k = 1, \ldots, K), \qquad P_{L_{K+1}} = R^*_K = 1 - \sum_{k=1}^{K} p_k = p_{K+1}.$$

Hence, the recursive allocation recovers the entire target distribution. *Q.E.D.*

Thus, the residual form is not restrictive as a parameterization of an ordered categorical distribution. This property concerns the expressive capacity of the composition rule and does not require the model-produced q_k values to be calibrated conditional probabilities.

In particular, Proposition 3 should be distinguished from Proposition 1. Proposition 1 concerns semantic consistency in the ideal hard-verification limit, where cumulative criterion outcomes recover the same one-hot exact-level assignment as direct classification. Proposition 3 instead states that, as a mathematical parameterization, the residual form can represent any soft categorical distribution when its residual parameters are chosen appropriately. The actual model-produced q_k values need not coincide with these constructed parameters $\widetilde{q}_k$.

A.3 RSV Score and Progressive Influence

Associate the ordered levels with scalar values

$$S_{L_1} > S_{L_2} > \cdots > S_{L_{K+1}},$$

and define the RSV score as

$$S_{\mathrm{RSV}} := \sum_{k=1}^{K+1} S_{L_k} P_{L_k}. \tag{12}$$

Proposition 4 (Boundedness and attainability of the RSV score). For every candidate response and every $\mathbf{q} = (q_1, \ldots, q_K) \in [0, 1]^K$,

$$S_{L_{K+1}} \leq S_{\mathrm{RSV}} \leq S_{L_1}.$$

Moreover, both bounds are attainable. In particular,

$$S_{\mathrm{RSV}} = S_{L_1} \iff q_1 = 1, \qquad S_{\mathrm{RSV}} = S_{L_{K+1}} \iff q_1 = \cdots = q_K = 0.$$

Proof. By Proposition 2, $\{P_{L_k}\}_{k=1}^{K+1}$ is a probability distribution, so S_{RSV} is a convex combination of the ordered values $S_{L_1} > \cdots > S_{L_{K+1}}$. Therefore,

$$S_{L_{K+1}} \le S_{\mathrm{RSV}} \le S_{L_1}.$$

If $q_1 = 1$, then $P_{L_1} = 1$ and every later mass contains the factor $1 - q_1 = 0$, so $S_{\mathrm{RSV}} = S_{L_1}$. Conversely, equality with the strict maximum requires all mass on L_1, hence $P_{L_1} = q_1 = 1$.

Similarly, if $q_1 = \cdots = q_K = 0$, then $P_{L_{K+1}} = 1$ and $S_{\mathrm{RSV}} = S_{L_{K+1}}$. Conversely, equality with the strict minimum requires

$$P_{L_{K+1}} = \prod_{j=1}^{K}(1 - q_j) = 1,$$

which, because every factor lies in $[0, 1]$, holds iff $q_j = 0$ for every j. Thus both endpoints are attainable under exactly the stated conditions. *Q.E.D.*

Define

$$Q_m := \sum_{k=1}^{m} P_{L_k}$$

as the cumulative probability mass on the first m highest-quality levels, and let

$$\Delta_m := S_{L_m} - S_{L_{m+1}} > 0$$

denote the score difference between adjacent quality levels.

Proposition 5 (Cumulative representation). For every $m \in \{1, \ldots, K\}$,

$$Q_m = 1 - \prod_{j=1}^{m}(1 - q_j),$$

and the RSV score admits the equivalent representation

$$\begin{aligned} S_{\mathrm{RSV}} &= S_{L_{K+1}} + \sum_{m=1}^{K} \Delta_m Q_m \\ &= S_{L_{K+1}} + \sum_{m=1}^{K} \Delta_m \left[1 - \prod_{j=1}^{m}(1 - q_j)\right]. \end{aligned} \tag{13}$$

Proof. Using $P_{L_k} = R_{k-1} - R_k$,

$$Q_m = \sum_{k=1}^{m}(R_{k-1} - R_k) = R_0 - R_m = 1 - \prod_{j=1}^{m}(1 - q_j).$$

Since

$$S_{L_r} = S_{L_{K+1}} + \sum_{m=r}^{K} \Delta_m,$$

substitution into Eq. equation 12 and exchange of the finite sums give

$$\begin{aligned} S_{\mathrm{RSV}} &= \sum_{r=1}^{K+1} P_{L_r} \left(S_{L_{K+1}} + \sum_{m=r}^{K} \Delta_m \right) \\ &= S_{L_{K+1}} \sum_{r=1}^{K+1} P_{L_r} + \sum_{m=1}^{K} \Delta_m \sum_{r=1}^{m} P_{L_r} \\ &= S_{L_{K+1}} + \sum_{m=1}^{K} \Delta_m Q_m. \end{aligned}$$

Combining this identity with the expression for Q_m yields Eq. equation 13. *Q.E.D.*

The cumulative form shows that RSV can be interpreted as starting from the lowest-quality value and adding successive quality increments weighted by cumulative high-quality probability mass.

Proposition 6 (Monotonicity and progressive residual influence). For every criterion $m \in \{1, \ldots, K\}$,

$$\begin{aligned} \frac{\partial S_{\mathrm{RSV}}}{\partial q_m} &= \sum_{k=m}^{K} \Delta_k \prod_{\substack{1 \le j \le k \\ j \neq m}} (1 - q_j) \\ &= R_{m-1} \sum_{k=m}^{K} \Delta_k \prod_{j=m+1}^{k} (1 - q_j) \ge 0. \end{aligned} \tag{14}$$

Moreover,

$$\frac{\partial S_{\mathrm{RSV}}}{\partial q_m} > 0 \quad \Longleftrightarrow \quad R_{m-1} > 0.$$

Proof. Differentiating Eq. equation 13 with respect to q_m, only terms with $k \ge m$ remain, because

$$\frac{\partial}{\partial q_m} \left[1 - \prod_{j=1}^{k} (1 - q_j) \right] = \prod_{\substack{1 \le j \le k \\ j \neq m}} (1 - q_j).$$

Therefore,

$$\frac{\partial S_{\mathrm{RSV}}}{\partial q_m} = R_{m-1} \sum_{k=m}^{K} \Delta_k \prod_{j=m+1}^{k} (1 - q_j), \qquad R_{m-1} = \prod_{j<m} (1 - q_j).$$

All factors are nonnegative. The inner sum is strictly positive because its $k = m$ term is $\Delta_m > 0$ (the corresponding product is empty and equals 1). Hence the derivative is nonnegative and is strictly positive iff $R_{m-1} > 0$. *Q.E.D.*

Therefore, increasing the Yes confidence of any criterion cannot decrease the final RSV score. More precisely, if positive residual probability mass remains after the preceding stricter criteria, then increasing the Yes confidence of criterion m strictly increases the RSV score.

Equivalently, let $q'_m > q_m$ while all other criterion confidences are held fixed. Since S_{RSV} is affine in q_m when the remaining q_j values are fixed,

$$\begin{aligned} &S_{\mathrm{RSV}}(q_1, \ldots, q'_m, \ldots, q_K) - S_{\mathrm{RSV}}(q_1, \ldots, q_m, \ldots, q_K) \\ &\quad = (q'_m - q_m) R_{m-1} \sum_{k=m}^{K} \Delta_k \prod_{j=m+1}^{k} (1 - q_j) \ge 0, \end{aligned}$$

with strict inequality whenever $R_{m-1} > 0$.

More importantly, the factor

$$R_{m-1} = \prod_{j<m} (1 - q_j)$$

formalizes the progressive nature of RSV: after all binary verification confidences have been obtained, the contribution of a relaxed criterion is modulated by the probability mass remaining after the preceding stricter criteria. This modulation concerns only probability composition and does not determine whether later criteria are evaluated.

A.4 Expected Quality as a Scalar Summary

For a fixed response, define an auxiliary discrete quality variable U by

$$\Pr(U = S_{L_k}) = P_{L_k}, \qquad k = 1, \ldots, K + 1.$$

The RSV score is exactly its expectation:

$$S_{\mathrm{RSV}} = \mathbb{E}[U].$$

Corollary 1 (Optimal scalar summary under squared loss). Among all scalar summaries $a \in \mathbb{R}$,

$$S_{\mathrm{RSV}} = \arg\min_{a \in \mathbb{R}} \mathbb{E}\big[(U-a)^2\big] .$$

Proof. The standard variance decomposition gives

$$\mathbb{E}\big[(U-a)^2\big] = \mathrm{Var}(U) + (a - \mathbb{E}[U])^2 .$$

The first term is independent of a, while the second is uniquely minimized at

$$a = \mathbb{E}[U] = S_{\mathrm{RSV}}.$$

Therefore,

$$S_{\mathrm{RSV}} = \arg\min_{a \in \mathbb{R}} \mathbb{E}\big[(U-a)^2\big] .$$

Q.E.D.

Hence, under squared loss, expected quality is the optimal scalar summary of the induced discrete quality distribution. Because the expectation depends on all level masses $\{P_{L_k}\}_{k=1}^{K+1}$, it incorporates the complete induced distribution when forming this scalar summary, rather than collapsing the evaluation to a single most-probable level.

B Algorithmic Details of Recursive Self-Verification

This section provides pseudocode for RSV scoring and preference-data construction. The implementation follows the formulation in the main paper. In particular, all verification criteria are evaluated separately on the same complete candidate response before recursive probability-mass composition. No hard verification decision is used to control the execution of subsequent criteria.

B.1 RSV Scoring for a Single Candidate

Algorithm 1 summarizes the complete scoring procedure for one candidate response. It consists of two distinct stages. First, the verifier evaluates all K ordered criteria and obtains their normalized Yes confidences. Second, these confidences are composed through residual probability-mass allocation to obtain the ordered quality distribution and its expected quality score.

Importantly, the first loop is completed for all K criteria before the second loop performs probability-mass allocation. Thus, the recursive structure is entirely contained in the composition of the soft verification confidences and does not correspond to conditional execution or hard routing between verification steps.

B.2 Preference-Data Construction

Algorithm 2 describes how response-level RSV scores are converted into preference supervision. Every candidate is scored independently before any pairwise relation is introduced. The highest-scoring response is selected as preferred and the lowest-scoring response as rejected.

The resulting preference dataset is subsequently used by the standard preference-optimization objective. RSV therefore modifies preference supervision construction rather than the downstream optimization objective itself.

C Verification Prompts

We present the verification prompts using the Knights-and-Knaves (K&K) benchmark with the $L = 4$ configuration as an illustrative example. This setting contains four ordered quality levels and therefore uses $K = L - 1 = 3$ binary verification steps. The three criteria correspond to progressively relaxed notions of response quality, ranging from a fully correct solution, to a mostly correct solution, and finally to a partially correct solution containing meaningful valid reasoning. Each step independently receives the same original puzzle and complete model-generated response as input and evaluates the response only against its corresponding criterion. The verifier outputs exactly `Yes` or `No`, and RSV uses the resulting verbalizer probabilities for subsequent recursive probability-mass composition.

Algorithm 1 Recursive Self-Verification for a Single Candidate

Require: Input x; candidate response y; ordered criteria $\mathcal{C}_{1:K}$; verifier $\mathcal{V}$; ordered level values $S_{L_1} > \cdots > S_{L_{K+1}}$
Ensure: Quality distribution $\mathbf{P} = [P_{L_1}, \ldots, P_{L_{K+1}}]$ and score S_{RSV}

// Stage 1: evaluate all binary criteria
for $k = 1, \ldots, K$ **do**
  $\phi_k \leftarrow \mathrm{BuildPrompt}(x, y, \mathcal{C}_k)$
  $(\ell_k^{\mathrm{Yes}}, \ell_k^{\mathrm{No}}) \leftarrow \mathcal{V}(\phi_k)$
  $q_k \leftarrow \dfrac{\exp(\ell_k^{\mathrm{Yes}})}{\exp(\ell_k^{\mathrm{Yes}}) + \exp(\ell_k^{\mathrm{No}})}$
end for

// Stage 2: recursively allocate probability mass
$R \leftarrow 1$
for $k = 1, \ldots, K$ **do**
  $P_{L_k} \leftarrow R q_k$
  $R \leftarrow R(1 - q_k)$
end for
$P_{L_{K+1}} \leftarrow R$

$S_{\mathrm{RSV}} \leftarrow \sum_{k=1}^{K+1} S_{L_k} P_{L_k}$
return $\mathbf{P}, S_{\mathrm{RSV}}$

Algorithm 2 RSV Preference-Data Construction

Require: Prompt–candidate collections $\{(x_i, \mathcal{Y}_i)\}_{i=1}^{M}$; ordered criteria $\mathcal{C}_{1:K}$; verifier $\mathcal{V}$; level values $S_{L_1} > \cdots > S_{L_{K+1}}$
Ensure: Preference dataset $\mathcal{D}_{\mathrm{pref}}$

$\mathcal{D}_{\mathrm{pref}} \leftarrow \varnothing$
for $i = 1, \ldots, M$ **do**
  for all $y \in \mathcal{Y}_i$ **do**
    $(\mathbf{P}_{i,y}, s_{i,y}) \leftarrow \mathrm{RSVScore}(x_i, y, \mathcal{C}_{1:K}, \mathcal{V}, S_{L_{1:K+1}})$
  end for
  $y_i^+ \leftarrow \arg\max_{y \in \mathcal{Y}_i} s_{i,y}$
  $y_i^- \leftarrow \arg\min_{y \in \mathcal{Y}_i} s_{i,y}$
  $\mathcal{D}_{\mathrm{pref}} \leftarrow \mathcal{D}_{\mathrm{pref}} \cup \{(x_i, y_i^+, y_i^-)\}$
end for
return $\mathcal{D}_{\mathrm{pref}}$

K&K Verification Step 1: Fully Correct

Original Puzzle: {`problem`}
Candidate Response: {`candidate_response`}
Verification Prompt: You are an expert evaluator of logical reasoning for Knights-and-Knaves puzzles, in which every character is either a knight who always tells the truth or a knave who always lies. Carefully examine the original puzzle and the complete candidate response, including both its reasoning process and its final character assignment. Determine whether the response is *fully correct*. Output `"Yes"` only if the entire solution is logically and factually correct: every character is assigned the correct identity; all relevant statements are interpreted consistently with the knight/knave truthfulness rules; all important logical constraints are correctly identified and applied; every key deduction required to derive the final assignment follows validly from the puzzle or from previously established valid conclusions; the complete reasoning chain is internally consistent; and the final assignment is fully supported by the reasoning. The response must contain no substantive contradiction, unsupported assumption, invalid implication, incorrect character assignment, or missing essential logical step that affects

the validity of the solution. Minor stylistic, formatting, or wording issues may be ignored only when they have no effect on logical correctness or completeness. If all of these requirements are satisfied, output `"Yes"`; otherwise, output `"No"`. Answer with just `"Yes"` or `"No"`.

Emphasis. This step requires complete correctness in both the final character assignment and the full chain of key logical deductions, with no substantive reasoning error.

K&K Verification Step 2: Mostly Correct

Original Puzzle: {`problem`}
Candidate Response: {`candidate_response`}
Verification Prompt: You are an expert evaluator of logical reasoning for Knights-and-Knaves puzzles, in which every character is either a knight who always tells the truth or a knave who always lies. Carefully examine the original puzzle and the complete candidate response, including its reasoning process and final character assignment. Determine whether the response is *at least mostly correct*. A fully correct response also satisfies this criterion. Output `"Yes"` if most of the solution is correct overall: the response should correctly understand and apply most of the important knight/knave truthfulness constraints, most of the main reasoning steps should be logically valid, and most character identities should be correctly determined or supported by valid reasoning. A small number of local errors are allowed, such as a minor reasoning mistake, a missing or unclear step, or a small number of incorrect or unresolved character identities, as long as these errors affect only a limited part of the solution and do not invalidate the main reasoning. The main approach should already be correct, and the response should be close to a fully correct solution without requiring major changes to its central reasoning. Output `"No"` if the response contains major logical errors, incorrectly handles several important truthfulness constraints, gets a substantial part of the character assignments wrong, or would require major changes to the main reasoning process to become correct. Answer with just `"Yes"` or `"No"`.

Emphasis. This step focuses on whether the central solution structure and most key deductions are correct, allowing only limited errors that can be repaired locally without reconstructing the main reasoning process.

K&K Verification Step 3: Partially Correct

Original Puzzle: {`problem`}
Candidate Response: {`candidate_response`}
Verification Prompt: You are an expert evaluator of logical reasoning for Knights-and-Knaves puzzles, in which every character is either a knight who always tells the truth or a knave who always lies. Carefully examine the original puzzle and the complete candidate response, including its reasoning process and final character assignment. Determine whether the response is *at least partially correct*. A fully correct or mostly correct response also satisfies this criterion. Output `"Yes"` if the response contains some meaningful and logically valid progress toward solving the puzzle. For example, the response may correctly apply at least one important knight/knave truthfulness constraint, make at least one valid logical deduction, correctly determine the identity of one or more characters with valid reasoning, correctly rule out an impossible assignment, or derive another useful constraint that helps narrow down the solution. The overall response does not need to be complete and may contain substantial mistakes, missing reasoning steps, unresolved characters, or even an incorrect final assignment, as long as there is a clear and nontrivial part of the reasoning that is genuinely correct and useful for solving the puzzle. Output `"No"` if the response only restates the puzzle, makes unsupported guesses, gives character identities without valid reasoning, contains only trivial observations, or does not include any meaningful correct logical progress. Answer with just `"Yes"` or `"No"`.

Emphasis. This step requires only a limited but genuine amount of correct logical progress, while excluding guesses, trivial observations, and unsupported claims that do not validly advance the solution.

Table 4: Incremental contribution of progressively relaxed verification criteria on K&K and TruthQuest. L denotes the number of induced quality levels and $K = L - 1$ the number of binary criteria. I→C denotes incorrect-over-correct pairs; Pref. Prec. denotes preference precision; Correct-at-top is top-1 accuracy when a correct candidate exists; Assign. Acc. denotes chosen assignment accuracy; BW Gap denotes the both-wrong quality gap; and $|\rho|$ denotes absolute within-prompt Spearman correlation with character length.

Benchmark	Model	L	I→C (%) ↓	Pref. Prec. (%) ↑	Correct at top (%) ↑	Assign. Acc. (%) ↑	BW Gap ↑	MRR ↑	NDCG@4 ↑	Pairwise Acc. ↑	Char. $\lvert\rho\rvert$ ↓
K&K	Qwen3-1.7B	2	11.38	71.84	55.96	72.91	0.0086	0.6974	0.5534	0.5871	0.1124
		3	9.75	74.12	57.89	73.88	0.0173	0.7100	0.5601	0.5927	0.0583
		4	9.42	74.46	58.22	74.11	0.0192	0.7138	0.5624	0.5936	0.0615
		5	9.20	74.81	58.39	74.24	0.0200	0.7160	0.5641	0.5941	0.0567
	Gemma-3-4B-IT	2	21.96	59.01	52.95	70.82	−0.0118	0.6609	0.5589	0.5451	0.3454
		3	20.38	59.92	53.86	72.18	−0.0058	0.6762	0.5687	0.5512	0.3098
		4	19.17	60.75	54.99	73.65	−0.0013	0.6898	0.5778	0.5564	0.2772
		5	18.83	61.04	55.28	73.91	0.0007	0.6941	0.5802	0.5580	0.2706
TruthQuest	Qwen3-1.7B	2	13.42	68.95	51.83	70.14	0.0047	0.6712	0.5318	0.5734	0.1287
		3	11.96	70.87	53.26	71.08	0.0108	0.6835	0.5396	0.5791	0.0876
		4	11.34	71.63	54.05	71.72	0.0131	0.6908	0.5442	0.5826	0.0794
		5	10.98	72.18	54.47	72.05	0.0145	0.6949	0.5471	0.5844	0.0821
	Gemma-3-4B-IT	2	24.35	56.72	49.61	69.03	−0.0154	0.6387	0.5421	0.5338	0.3821
		3	22.81	57.62	50.58	70.25	−0.0101	0.6519	0.5508	0.5394	0.3497
		4	21.47	58.36	51.42	71.31	−0.0060	0.6638	0.5592	0.5446	0.3268
		5	20.96	58.88	51.96	71.84	−0.0037	0.6701	0.5635	0.5472	0.3194

D Additional Experiments and Analyses

D.1 Incremental Contribution of Relaxed Criteria

We next isolate the contribution of verification granularity by progressively adding relaxed criteria to RSV. For a given L, RSV uses the first $K = L - 1$ ordered binary criteria while retaining the same residual probability-mass composition. This ablation therefore examines whether the additional, semantically weaker criteria provide useful discrimination beyond the strictest correctness-oriented criterion. We report results on K&K and TruthQuest with Qwen3-1.7B and Gemma-3-4B-IT.

The results show a consistent benefit from progressively enriching the verification hierarchy. Across both benchmarks and both backbones, increasing L generally reduces incorrect-over-correct selections while improving preference precision, correct-at-top accuracy, assignment accuracy, and within-prompt ranking metrics. For example, on K&K, increasing L from 2 to 5 reduces I→C from 11.38% to 9.20% for Qwen3-1.7B and from 21.96% to 18.83% for Gemma-3-4B-IT, while MRR increases from 0.6974 to 0.7160 and from 0.6609 to 0.6941, respectively. The both-wrong gap also improves as additional criteria are introduced, indicating that the relaxed criteria provide useful distinctions even when neither candidate is fully correct. A similar pattern holds on TruthQuest. Meanwhile, the absolute correlation with response length is substantially reduced relative to $L = 2$, suggesting that the additional criteria improve discrimination without increasing reliance on superficial length cues. Although the marginal gains become smaller at higher granularities and a few diagnostic metrics are not strictly monotonic, the overall pattern is stable: the relaxed criteria contribute complementary information beyond the strictest binary judgment, yielding progressively more informative preference and ranking signals.

D.2 Ablation of Distribution-to-Score Reduction

We examine how the induced ordinal distribution should be reduced to a scalar score for preference construction. All variants use the same candidate responses, verifier outputs, and $L = 4$ quality distribution, differing only in the final score reduction. *MAP Level* uses the score of the most probable quality level; *Ordinal Median* selects the first level whose cumulative probability reaches 0.5; *Sampled Level* draws one level from $\mathrm{Categorical}(P_{L_1}, \ldots, P_{L_L})$ and uses its associated score; and *Expected Quality*, used by RSV, computes $S_{\mathrm{Exp}} = \sum_{k=1}^{L} P_{L_k} S_{L_k}$. Sampled Level is averaged over

Table 5: Comparison of distribution-to-score reduction rules on K&K and TruthQuest with $L = 4$. MAP uses the most probable quality level; Ordinal Median uses the first level whose cumulative probability reaches 0.5; Sampled Level draws one level from the induced distribution; and Expected Quality uses the probability-weighted quality score. Sampled Level results are averaged over multiple sampling seeds, with each seed evaluated separately.

(a) Preference and selection quality

Benchmark	Model	Scoring	I→C (%) ↓	Pref. Prec. (%) ↑	Correct-at-top (%) ↑	Assign. Acc. (%) ↑	BW Gap ↑
K&K	Qwen3-1.7B	MAP Level	10.31	72.93	57.36	73.42	0.0121
		Ordinal Median	9.88	73.54	57.18	73.77	0.0157
		Sampled Level (1 draw)	10.02	73.70	57.72	73.84	0.0165
		Expected Quality (RSV)	**9.42**	**74.46**	**58.22**	**74.11**	**0.0192**
	Gemma-3-4B-IT	MAP Level	20.72	59.47	53.71	71.86	−0.0076
		Ordinal Median	20.04	60.08	54.28	72.71	−0.0040
		Sampled Level (1 draw)	19.76	60.16	54.53	72.98	−0.0034
		Expected Quality (RSV)	**19.17**	**60.75**	**54.99**	**73.65**	−0.0013
TruthQuest	Qwen3-1.7B	MAP Level	12.17	70.31	52.81	70.76	0.0070
		Ordinal Median	11.56	70.94	53.42	71.29	0.0106
		Sampled Level (1 draw)	11.74	71.10	53.70	71.41	0.0115
		Expected Quality (RSV)	**11.34**	**71.63**	**54.05**	**71.72**	**0.0131**
	Gemma-3-4B-IT	MAP Level	22.76	57.18	50.07	69.67	−0.0127
		Ordinal Median	21.93	57.91	50.89	70.61	−0.0090
		Sampled Level (1 draw)	22.05	58.03	51.16	70.83	−0.0081
		Expected Quality (RSV)	**21.47**	**58.36**	**51.42**	**71.31**	−0.0060

(b) Within-prompt ranking and score behavior

Benchmark	Model	Scoring	MRR ↑	Top-4 (%) ↑	NDCG@4 ↑	Pairwise Acc. ↑	Tie Rate (%) ↓
K&K	Qwen3-1.7B	MAP Level	0.7025	80.92	0.5542	0.5885	36.41
		Ordinal Median	0.7069	81.46	0.5578	0.5907	31.83
		Sampled Level (1 draw)	0.7086	81.63	0.5589	0.5916	27.94
		Expected Quality (RSV)	**0.7138**	**82.08**	**0.5624**	**0.5936**	**0.18**
	Gemma-3-4B-IT	MAP Level	0.6718	80.47	0.5640	0.5488	42.16
		Ordinal Median	0.6809	81.92	0.5712	0.5527	37.64
		Sampled Level (1 draw)	0.6831	82.44	0.5731	0.5538	32.85
		Expected Quality (RSV)	**0.6898**	**83.58**	**0.5778**	**0.5564**	**0.21**
TruthQuest	Qwen3-1.7B	MAP Level	0.6781	78.10	0.5348	0.5751	34.57
		Ordinal Median	0.6856	79.01	0.5403	0.5794	30.41
		Sampled Level (1 draw)	0.6874	79.28	0.5417	0.5806	26.73
		Expected Quality (RSV)	**0.6908**	**79.66**	**0.5442**	**0.5826**	**0.19**
	Gemma-3-4B-IT	MAP Level	0.6455	74.91	0.5441	0.5341	44.28
		Ordinal Median	0.6558	75.87	0.5519	0.5402	39.36
		Sampled Level (1 draw)	0.6581	76.24	0.5544	0.5421	34.11
		Expected Quality (RSV)	**0.6638**	**76.69**	**0.5592**	**0.5446**	**0.23**

I→C: incorrect-over-correct pairs; Pref. Prec.: preference precision; Correct-at-top: top-1 accuracy when a correct candidate exists; Assign. Acc.: chosen assignment accuracy; BW Gap: both-wrong quality gap. Tie Rate is the percentage of within-prompt candidate pairs receiving identical scalar scores. For Sampled Level, one level is drawn per candidate for each sampling seed; ranking metrics are computed separately for each seed and then averaged.

multiple sampling seeds, whereas Expected Quality deterministically integrates probability mass across all quality levels.

Expected Quality performs consistently best across both benchmarks and backbones. For example, on K&K with Qwen3-1.7B, it reduces I→C from 10.31% under MAP to 9.42%, while increasing preference precision from 72.93% to 74.46% and MRR from 0.7025 to 0.7138. The same trend is observed for Gemma-3-4B-IT and on TruthQuest. The difference is particularly pronounced in score resolution: Expected Quality reduces the within-prompt tie rate to only 0.18–0.23%, compared with 34–44% for MAP, while simultaneously improving MRR, NDCG@4, and pairwise accuracy. Sampled Level generally improves over discrete MAP-based scoring but remains below Expected Quality, consistent with the additional variance introduced by using a single stochastic realization. Since all variants share the same induced distribution, these results indicate that using probabil-

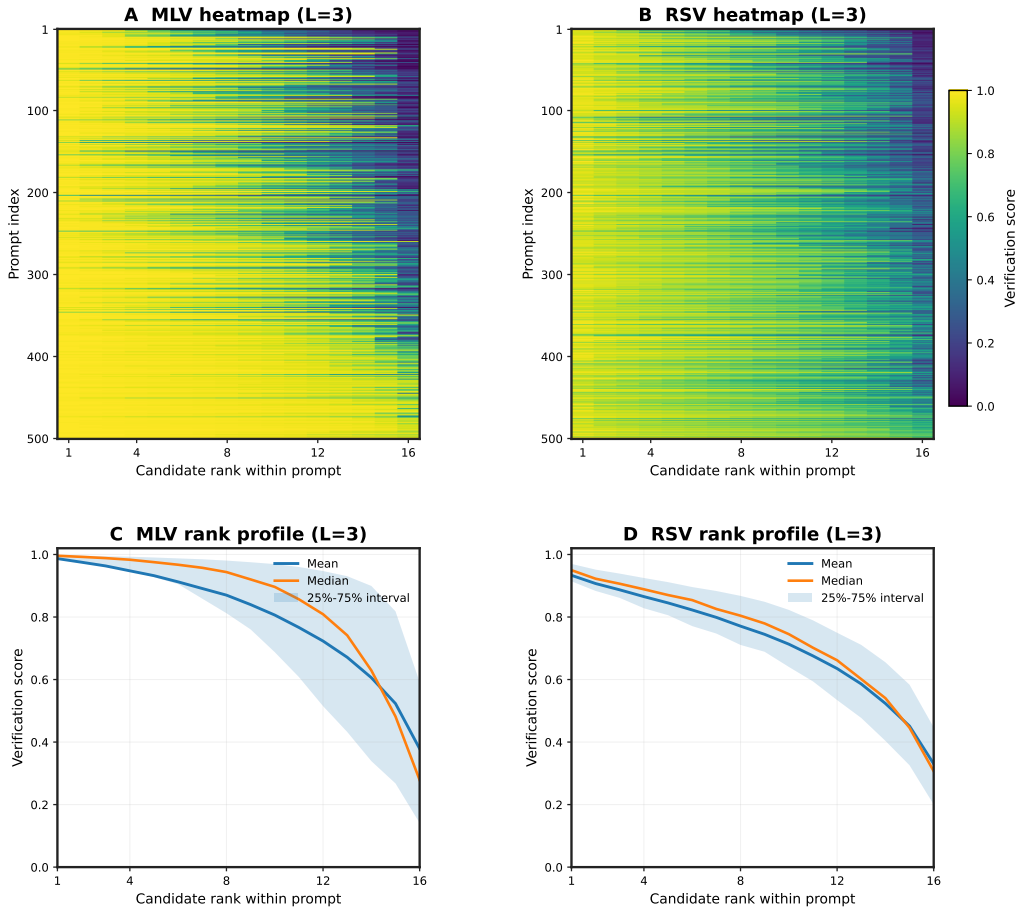

Figure 5: Within-prompt verification-score distributions for MLV and RSV with $L = 3$ on K&K using Gemma-3-4B-IT.

ity mass from all quality levels provides a more informative and stable scalar ranking signal than collapsing the distribution to a single discrete level.

D.3 Visualization of Verification Score Distributions

In the main text, we analyze why RSV substantially outperforms direct multi-level verification (MLV) under the same number of quality levels, primarily through statistics of the resulting preference construction and within-prompt candidate ranking. Here, we complement that analysis from the perspective of the verifier scores themselves. Specifically, we examine how MLV and RSV distribute verification confidence across quality levels and how these distributions are converted into scalar scores for individual candidate responses.

We use Gemma-3-4B-IT on K&K and consider $L \in \{3, 4, 5\}$. For each of the first 500 prompts, we evaluate 16 candidate responses and sort them in descending order according to the verification score produced by each method. Figures 5–7 visualize the resulting within-prompt score structure. The heatmaps show candidate-level scores across prompts, while the accompanying rank profiles summarize the mean, median, and interquartile range at each candidate rank.

At $L = 3$, MLV already exhibits a strong concentration of high verification scores. Consequently, several highly ranked candidates within the same prompt receive very similar scalar scores. RSV instead produces a more gradual score profile, creating clearer separation among candidates that MLV tends to place in a similar high-quality region.

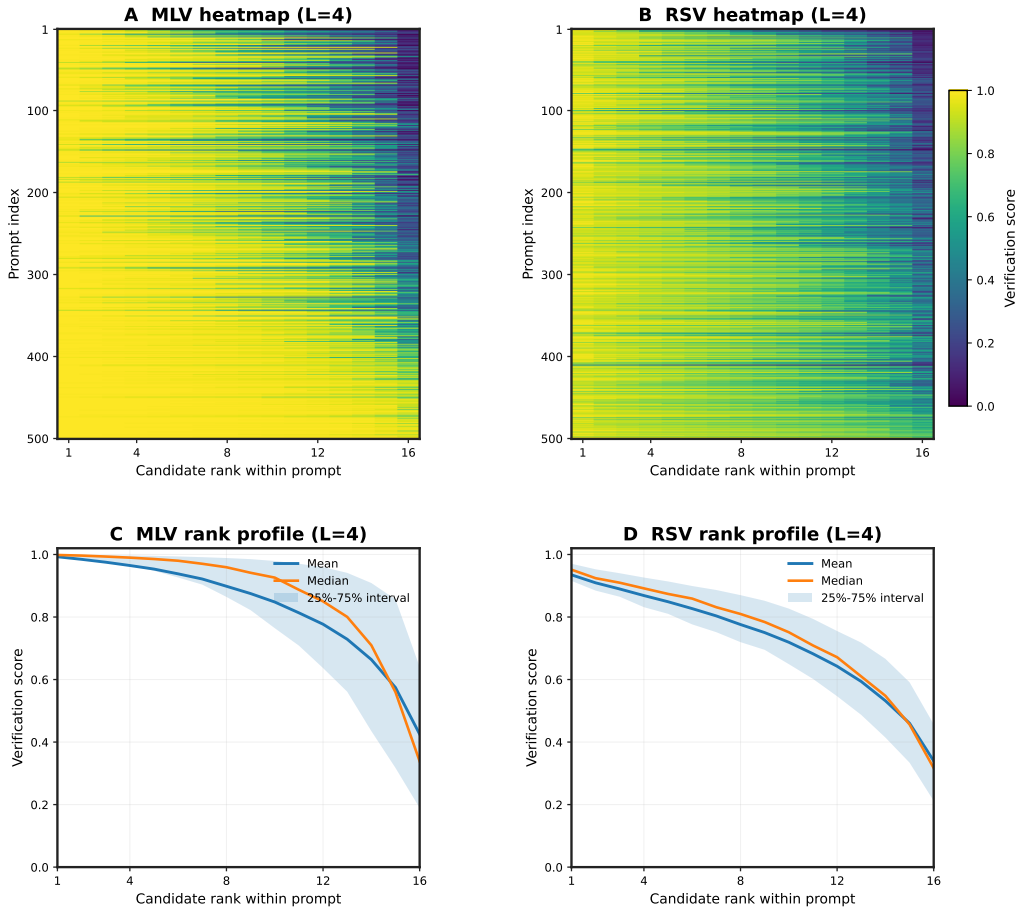

Figure 6: Within-prompt verification-score distributions for MLV and RSV with $L = 4$ on K&K using Gemma-3-4B-IT.

The difference becomes more pronounced at $L = 4$. MLV produces a broad high-score region in which many candidates are assigned nearly indistinguishable scores, whereas RSV maintains a smoother transition from highly rated to weaker responses. The corresponding rank profiles indicate that this behavior is systematic across prompts rather than being driven by a small number of outliers.

A similar pattern persists at $L = 5$. Thus, the distinction between RSV and MLV is not simply a consequence of the number of available quality levels. Even under matched L, the two verification mechanisms induce substantially different probability and scalar-score structures.

To quantify these observations, Table 6 reports two complementary groups of statistics. For the induced level distribution $\mathbf{p} = (p_1, \ldots, p_L)$, we measure normalized entropy, the maximum probability mass $\max_k p_k$, the fraction of responses with $\max_k p_k > 0.9$, and the mean adjacent-level probability gap $\frac{1}{L-1} \sum_{k=1}^{L-1} |p_{k+1} - p_k|$. These quantities characterize how concentrated the verification distribution is.

We additionally measure the resolution of the resulting scalar scores within each prompt. *Near-Tie* denotes the fraction of candidate pairs satisfying $|S_i - S_j| < 0.01$, while *Top-4 Near-Tie* applies the same criterion to the four highest-scoring candidates. We also report the mean and median margin between the top-ranked and second-ranked candidates. Lower near-tie rates and larger top-1–top-2 margins indicate finer candidate-level discrimination.

Table 6 quantitatively supports the patterns visible in the heatmaps. Across all three values of L, MLV produces substantially more concentrated level distributions: approximately 56–61% of its evaluations assign more than 0.9 probability mass to a single level, compared with about 18% for

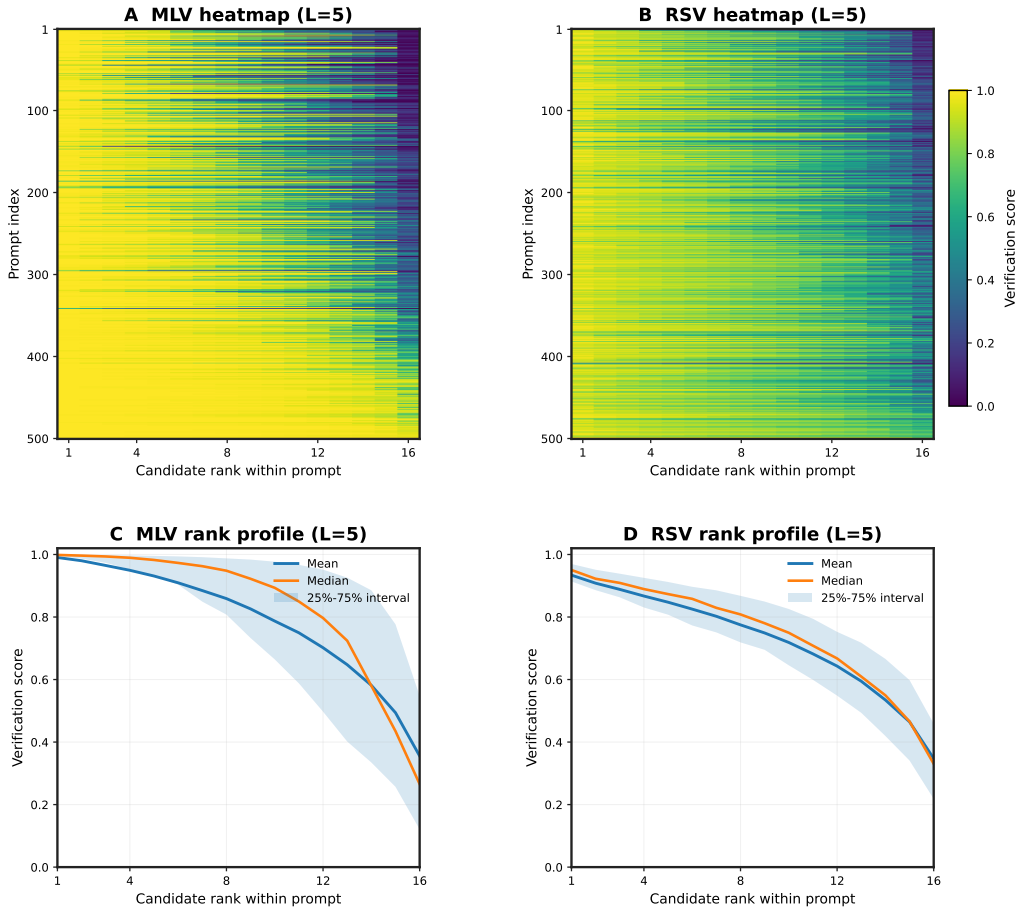

Figure 7: Within-prompt verification-score distributions for MLV and RSV with $L = 5$ on K&K using Gemma-3-4B-IT.

RSV. Correspondingly, MLV exhibits lower normalized entropy and larger adjacent-level probability gaps.

More importantly, this concentration is accompanied by markedly stronger score saturation within individual prompts. At $L = 4$, for example, 20.73% of all MLV candidate pairs are near-ties, compared with only 4.59% for RSV; among the top four candidates, the rates are 72.07% and 23.87%, respectively. The median top-1–top-2 margin also increases from 0.0013 under MLV to 0.0173 under RSV. The same pattern holds at $L = 3$ and $L = 5$.

These results do not imply that higher-entropy distributions are inherently better. Rather, they show that the two verification mechanisms induce different score geometries under the same quality granularity: MLV tends to make sharper level assignments but produces more saturated scalar scores, whereas RSV yields finer distinctions among candidates within the same prompt. This score-level behavior provides a complementary explanation for the stronger preference construction and ranking quality of RSV observed in the main text.

D.4 RSV and Baselines on Additional Challenging Mathematical Reasoning Benchmarks

The main experiments evaluate RSV on two logical reasoning benchmarks, K&K and TruthQuest, together with the mathematical reasoning benchmark MATH-Hard. To further examine whether the observed improvements extend to more challenging mathematical reasoning settings, we conduct additional experiments on LIMO (Ye et al., 2025a), AIME 2025/2026 (Mathematical Association of America, 2026), and RIMO-N (Chen et al., 2025b).

Table 6: Quantitative characterization of the verification-score distributions in Figures 5–7. Statistics are computed over 8,000 candidate responses from 500 K&K prompts.

(a) Distributional concentration across quality levels

L	Method	Norm. Entropy	Mean Max Mass	$p_{\max} > 0.9$ (%)	Adjacent Gap
3	MLV	0.3516	0.8514	57.73	0.4400
3	**RSV**	0.5961	0.7342	17.66	0.3681
4	MLV	0.2807	0.8586	60.56	0.3057
4	**RSV**	0.5542	0.7166	17.59	0.2523
5	MLV	0.2760	0.8382	55.98	0.2300
5	**RSV**	0.5097	0.7132	17.59	0.2028

(b) Within-prompt score resolution

L	Method	Near-Tie (%) ↓	Top-4 Near-Tie (%) ↓	Mean Top-1–Top-2↑	Median Top-1–Top-2↑
3	MLV	16.23	64.57	0.0115	0.0024
3	**RSV**	**4.49**	**23.63**	**0.0263**	**0.0173**
4	MLV	20.73	72.07	0.0082	0.0013
4	**RSV**	**4.59**	**23.87**	**0.0256**	**0.0173**
5	MLV	18.58	66.93	0.0104	0.0014
5	**RSV**	**4.64**	**24.17**	**0.0257**	**0.0172**

For each backbone, we split LIMO into 70% training data and a 30% held-out evaluation split. All self-evolution methods are trained using only the 70% LIMO training split. The resulting models are then evaluated on the held-out LIMO split as well as on AIME 2025/2026 and RIMO-N without additional adaptation. For AIME, we report accuracy pooled over the AIME 2025 and AIME 2026 evaluation sets. All results are reported as accuracy (%).

Across all three backbones, RSV achieves consistent gains on the held-out LIMO split and on the two additional mathematical reasoning benchmarks. RSV with $L = 5$ obtains the highest accuracy in every backbone–benchmark combination. Compared with the strongest non-RSV baseline, RSV ($L = 5$) improves LIMO/AIME/RIMO-N by $1.2/3.3/2.1$ percentage points for Gemma-3-4B-IT, $1.1/3.4/1.9$ points for Qwen3-1.7B, and $1.4/3.4/0.7$ points for Qwen2.5-7B-Instruct.

Increasing the verification granularity from $L = 3$ to $L = 5$ also produces a consistent non-decreasing trend across all three evaluation sets and model families. These additional results extend the findings of the main experiments to more challenging mathematical reasoning settings, showing that the advantage of RSV over alternative self-evolution methods is preserved beyond the benchmarks considered in the main text.

D.5 Verifier-Call-Matched Comparison

Standard MLV evaluates each candidate with a single multiclass judgment, whereas RSV uses $K = L - 1$ separately elicited binary criteria. To control for the number of verifier evaluations, we construct *MLV-Ensemble-K*, which applies K semantically equivalent multiclass prompts with the same quality-level definitions and averages their predicted level distributions. Let $\mathbf{p}^{(m)}$ denote the distribution produced by the m-th MLV judgment. We compute

$$\bar{p}_\ell = \frac{1}{K} \sum_{m=1}^{K} p_\ell^{(m)}, \qquad S_{\text{MLV-Ens}} = \sum_{\ell=1}^{L} \bar{p}_\ell S_{L_\ell}. \tag{15}$$

Table 7: Performance of RSV and baseline self-evolution methods on additional challenging mathematical reasoning benchmarks. All self-evolution methods are trained using the 70% LIMO training split and evaluated on the held-out 30% LIMO split, AIME 2025/2026, and RIMO-N. All values are accuracy (%). The best result for each backbone and evaluation set is shown in **bold**.

Model	Method	LIMO	AIME 2025/2026	RIMO-N
Gemma-3-4B-IT	Base	18.1	10.0	33.7
	Self-Rewarding	21.1	13.3	36.8
	SimpleSE	19.2	11.7	35.2
	GenRM	19.0	11.7	33.9
	IPO	21.3	15.0	37.3
	RSV ($L = 3$)	21.7	16.7	38.1
	RSV ($L = 4$)	22.1	16.7	38.5
	RSV ($L = 5$)	**22.5**	**18.3**	**39.4**
Qwen3-1.7B	Base	22.1	11.7	36.6
	Self-Rewarding	24.7	13.3	37.5
	SimpleSE	23.2	11.7	37.2
	GenRM	22.6	10.0	36.9
	IPO	24.5	13.3	38.8
	RSV ($L = 3$)	25.1	13.3	39.2
	RSV ($L = 4$)	25.4	15.0	40.3
	RSV ($L = 5$)	**25.8**	**16.7**	**40.7**
Qwen2.5-7B-Instruct	Base	24.3	6.7	35.5
	Self-Rewarding	25.9	8.3	34.8
	SimpleSE	25.6	5.0	35.7
	GenRM	24.6	6.7	34.2
	IPO	26.0	8.3	36.9
	RSV ($L = 3$)	26.6	8.3	37.1
	RSV ($L = 4$)	27.1	10.0	37.4
	RSV ($L = 5$)	**27.4**	**11.7**	**37.6**

All other components, including candidate responses, level scores, preference construction, and DPO training, are kept unchanged. We evaluate $L \in \{3, 4\}$, corresponding to two and three verifier calls per candidate, respectively.

Table 8 shows that increasing the number of multiclass verification calls yields only modest downstream changes. Across both benchmarks, MLV-Ensemble-K remains close to the corresponding single-call MLV baseline. For Qwen3-1.7B, the average accuracy changes from 33.8% to 33.9% and from 39.4% to 40.3% on K&K at $L = 3$ and $L = 4$, respectively; on TruthQuest, the corresponding changes are 31.3% to 31.6% and 35.4% to 36.0%. A similar pattern holds for Gemma-3-4B-IT, and several individual difficulty groups show small decreases rather than consistent gains.

In contrast, RSV remains substantially stronger under the same verifier-call budget across both benchmarks, granularities, and backbones. These results indicate that RSV's advantage is not primarily explained by additional verifier evaluations. Rather, separately eliciting distinct ordered binary criteria provides a substantially stronger signal for preference construction than repeatedly eliciting semantically equivalent multiclass judgments.

E Implementation Details

For all experiments, we generate 16 candidate responses per question and evaluate each candidate using the ordered binary criteria of RSV. The resulting RSV scores are used to select the highest- and lowest-scoring responses as the preferred and rejected samples for full-parameter DPO training. For GRPO, RSV scores are instead used directly as on-policy reward signals. Qwen3-1.7B is run with native thinking disabled, while Gemma-3-4B-IT and Qwen2.5-7B-Instruct use their standard instruction-tuned inference protocols. All experiments are conducted on eight NVIDIA H20 GPUs.

Table 8: Verifier-call-matched comparison on K&K and TruthQuest. MLV-Ensemble-K averages $K = L - 1$ semantically equivalent multiclass verification judgments, matching the number of verifier calls used by RSV. All values are downstream accuracy (%).

				K&K				TruthQuest			
Model	L	**Verification**	**Calls / cand.**	**2–3**	**4–5**	**6–8**	**Avg.**	**3–4**	**5–6**	**7–8**	**Avg.**
Qwen3-1.7B	3	MLV	1	50.1	35.8	21.6	33.8	45.8	30.6	17.4	31.3
	3	MLV-Ensemble-2	2	50.8	36.2	21.2	33.9	46.9	29.9	18.0	31.6
	3	**RSV**	2	**83.5**	**68.0**	**39.1**	**60.0**	**72.1**	**52.8**	**35.7**	**53.5**
	4	MLV	1	66.2	37.5	22.9	39.4	51.4	34.7	20.1	35.4
	4	MLV-Ensemble-3	3	67.8	39.4	22.5	40.3	53.6	36.5	18.0	36.0
	4	**RSV**	3	**86.0**	**69.9**	**40.6**	**61.9**	**74.3**	**54.1**	**38.6**	**55.7**
Gemma-3-4B-IT	3	MLV	1	53.6	21.5	7.4	24.6	46.8	21.9	10.2	26.3
	3	MLV-Ensemble-2	2	53.0	22.1	7.9	24.8	47.7	21.1	10.8	26.5
	3	**RSV**	2	**74.6**	**39.7**	**15.2**	**39.2**	**59.7**	**30.3**	**15.2**	**35.1**
	4	MLV	1	57.5	21.7	9.5	26.7	49.2	23.4	11.6	28.1
	4	MLV-Ensemble-3	3	58.8	23.2	9.3	27.4	51.5	25.0	9.5	28.7
	4	**RSV**	3	**77.1**	**41.3**	**16.9**	**41.1**	**61.7**	**31.2**	**16.0**	**36.3**

Table 9: Key hyperparameters for RSV self-evolution on K&K. Verification settings are shown for the representative $L = 4$ configuration.

Stage	**Setting**	**Value**
Generation	Candidates / question	16
	Temperature / Top-p	0.7 / 1.0
	Max. generation / context	8192 / 32768
Verification	Quality levels / criteria (L/K)	4 / 3
	Level scores $(L = 4)$	$\{1, 0.3, 0.1, 0\}$
	Temperature	0.6
	Max. verifier input	30000
DPO	Gradient accumulation	4
	DPO β / cutoff	0.1 / 4096
	Weight decay / grad. clip	0 / 1.0
GRPO	Questions / policy update	50
	Rollouts / question	16
	PPO epochs / learning rate	$1 / 1 \times 10^{-6}$
	Weight decay / grad. clip	0 / 1.0
Backbone-specific DPO training settings		
Gemma-3-4B-IT	Epochs / learning rate	$1 / 5 \times 10^{-7}$
Qwen3-1.7B	Epochs / learning rate	$3 / 5 \times 10^{-7}$
Qwen2.5-7B-Instruct	Epochs / learning rate	$1 / 3 \times 10^{-7}$

Table 9 summarizes the principal hyperparameters. For verification, we report the representative $L = 4$ configuration. Experiments with $L \in \{3, 4, 5\}$ follow the same pipeline with $K = L - 1$ ordered binary criteria and the corresponding fixed monotonic ordinal score mapping. Unless otherwise specified, the remaining generation, verification, and optimization settings are shared across verification granularities and model backbones.