

Learning from Self-Generated Responses: Reliability–Uncertainty Self-Distillation for Reasoning

Jinhao Pan, Bowen Wei, Yuqing Zhou, Ziwei Zhu

George Mason University

{jpan23, bwei2, yzhou31, zzhu20}@gmu.edu

Abstract

Reference solutions and question-specific rubrics are not always available to guide the teacher in self-distillation. A large language model’s own solution attempts offer an alternative, but their quality is difficult to assess in open-ended tasks without automatic correctness checks, and even correct attempts can involve uncertain token predictions. We propose Reliability–Uncertainty Self-Distillation (RUSD) to learn from these attempts. The same model serves as teacher and student: the teacher receives the question and completed attempt, and the student matches the teacher’s token-level predictions using a question-only prompt. RUSD weights each attempt’s distillation loss by additively combining judged reliability, estimated from a self-judge’s score and confidence, with student predictive uncertainty. Across eleven reasoning benchmarks, RUSD outperforms all five baselines in average accuracy, reaching 70.47% on Qwen3-4B and 76.69% on Qwen3-8B (8.90 and 2.10 percentage points above the base models). Data and code are available at <https://github.com/JP-25/RUSD>.

1 Introduction

Self-distillation trains large language models (LLMs) by using the model itself as a teacher (Zhao et al., 2026a; Gu et al., 2026). In recent approaches, the teacher receives the question together with additional *teacher context*, such as a reference solution; the student uses a question-only prompt and learns by matching the teacher’s next-token predictions along student-generated responses.

Challenge 1: Reference solutions and rubrics are not always available to guide the teacher. SDFT and OPSD supply demonstration responses (example answers) or reference solutions as teacher context (Shenfeld et al., 2026; Zhao et al., 2026a). RCSD and RGSD use question-specific rubrics,

which specify what a good response should contain (Gu et al., 2026; Rezaei et al., 2026). When these solutions or rubrics are unavailable, preparing them can require expensive human annotation or additional model generation. One alternative is to learn from the model’s own generated responses without supplied reference solutions (Pei et al., 2026; Li et al., 2026). These self-generated *solution attempts*, however, may contain errors.

Challenge 2: Assessing attempt quality is difficult in open-ended reasoning. On open-ended reasoning tasks, reliable automatic verifiers for these self-generated attempts are often unavailable (Ye et al., 2026a). In reinforcement learning with verifiable rewards (RLVR), mathematical answers can be checked against known results, and code can be run against tests (DeepSeek-AI et al., 2025). Without comparable checks or reference answers, it is difficult to determine which attempts are correct and how much to learn from them. An LLM *judge*, a model prompted to assess response quality, can score these attempts, but its ratings can be wrong (Zheng et al., 2023). And using a larger external judge can increase the computation or API fees needed to score training attempts (Verga et al., 2024; Xie et al., 2026).

Our intuition is to learn from correct attempts generated with uncertain token predictions, so the model can produce the reasoning more confidently. We propose *Reliability–Uncertainty Self-Distillation* (RUSD, Figure 1) based on this intuition. For each question, the current model generates one solution attempt. As teacher, the same model receives the question and completed attempt in its prompt to provide token-level guidance. A judge assesses the attempt’s quality: high scores supported by confident judgments indicate greater *judged reliability*. As student, the model learns to match the teacher’s guidance with only the question as its prompt. We weight each attempt’s distilla-

tion loss using judged reliability and the student’s uncertainty in predicting its tokens, giving priority to attempts judged reliable and to attempts with uncertain token predictions.

Our contributions are threefold: **1.** We introduce RUSD, which uses one model-generated attempt as teacher context without supplied solutions or question-specific rubrics. **2.** We design response weights that combine judged reliability and student uncertainty, giving greater learning priority to attempts judged reliable and to attempts with uncertain token predictions. **3.** Across eleven reasoning benchmarks, RUSD outperforms all five baselines in macroaverage accuracy at both scales, improving Qwen3-4B and Qwen3-8B by 8.90 and 2.10 over the base models. External ratings, training analyses, and ablations examine its weighting signals and learning behavior.

2 Related Work

Self-distillation with and without supplied references. Self-distillation trains a model to match its own predictions. In recent on-policy LLM methods, the teacher receives additional context and the student matches the teacher’s token distributions along student-generated responses (Zhao et al., 2026a). SDFT and OPSD supply demonstrations or reference solutions; RCSD and RGSD supply question-specific rubrics (Shenfeld et al., 2026; Zhao et al., 2026a; Gu et al., 2026; Rezaei et al., 2026). Without supplied solutions, NSD constructs negative teacher conditions from model responses, whereas Canon and u-OPSD select a self-generated solution by majority-answer consensus across multiple responses (Pei et al., 2026; Gkountouras et al., 2026; Li et al., 2026). RUSD uses one completed attempt as teacher context, without negative conditions or answer consensus.

Assessing self-generated attempts. SDPO supplies environment feedback or successful responses to the teacher, while SD-Zero trains a reviser with binary correctness feedback (Hübotter et al., 2026; He et al., 2026). Such feedback requires response assessment; consensus requires rules for matching answers. RUSD estimates attempt reliability from judge score and confidence without correctness labels. Reliability controls the response weight; the teacher supplies next-token predictions.

Weighting by confidence and uncertainty. LLKD’s weighted variant combines teacher confi-

dence and student uncertainty for classification (Li et al., 2025). EGRSD weights token updates using teacher entropy, PW-OPSD uses token position, and COSE uses evaluator confidence for policy updates and replay (Ke et al., 2026; Liu et al., 2026; Wei et al., 2026). RUSD combines judged reliability with student predictive uncertainty to weight each attempt’s distillation loss. Appendix A gives fuller comparisons and the scope of our baselines.

3 Method

Overview. Figure 1 illustrates the RUSD workflow, and Algorithm 1 in Appendix B shows the training procedure. RUSD trains on questions $\mathcal{D} = \{\mathbf{x}_i\}$ without supplied reference solutions or question-specific rubrics. The current model π_θ serves as teacher and student under different prompts. The initial model π_{θ_0} is kept frozen throughout training and serves as the judge. The current model generates one attempt, which the teacher uses as additional context to provide token-level guidance (1). The judge’s score and confidence estimate the attempt’s reliability (2). We also measure the student’s uncertainty in predicting the attempt’s tokens (3). These signals determine the attempt’s training weight (4). The student learns to match the teacher’s token distributions, with each attempt’s contribution scaled by its weight (5). Table 4 in Appendix B gives the three prompts.

3.1 Using the attempt as teacher context

For each sampled question $\mathbf{x}_i$, the current model π_θ generates one solution attempt $\mathbf{y}_i$:

$$\mathbf{y}_i \sim \pi_\theta(\cdot \mid \mathcal{P}_S(\mathbf{x}_i)), \quad (1)$$

where $\mathcal{P}_S$ is the question-only prompt. The teacher prompt $\mathcal{P}_T(\mathbf{x}_i, \mathbf{y}_i)$ contains the question and completed attempt. The prompt describes the attempt as potentially useful or misleading and asks the model to solve the problem independently using any useful information.

Following on-policy self-distillation (Zhao et al., 2026a), we evaluate teacher and student next-token distributions at every position within this same sampled response. At position t , both receive the preceding response tokens $\mathbf{y}_{i,<t} = (y_{i,1}, \dots, y_{i,t-1})$. Only the teacher prompt also contains a complete copy of $\mathbf{y}_i$, giving it access to the response’s later

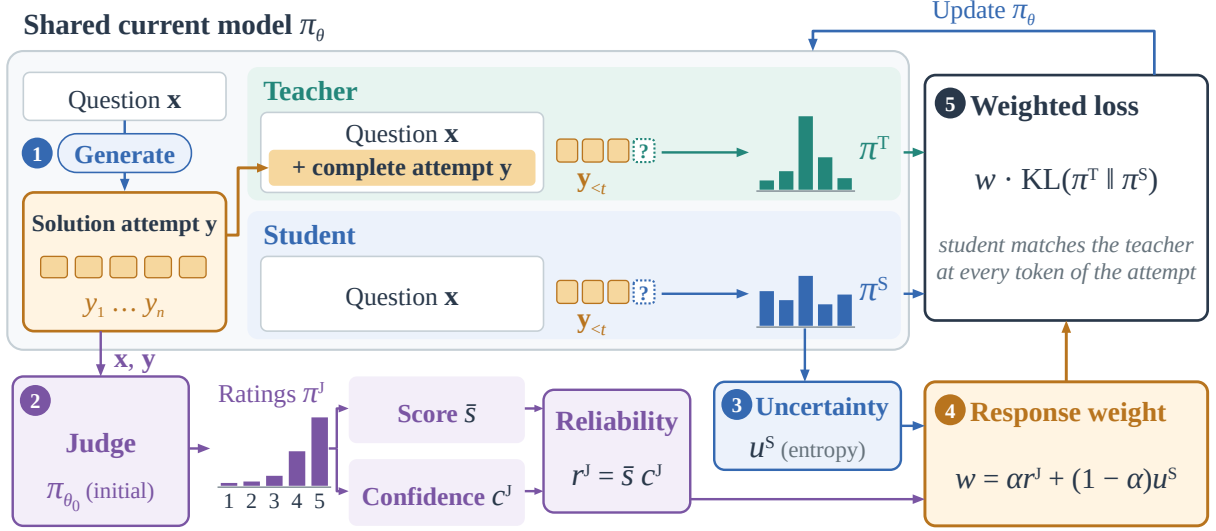


Figure 1: Overview of RUSD. ① π_θ generates attempt y for question x . Teacher and student predict next-token distributions from the same response prefix $y_{<t}$ under different prompts; only the teacher prompt includes the complete attempt. The teacher generates no additional response. ② Judge score s , normalized to $\bar{s}$, and confidence c^J give judged reliability r^J . ③ Normalized student entropy averaged over the response gives uncertainty u^S . ④ A weighted sum of reliability and uncertainty gives response weight w . ⑤ The weighted distillation loss updates the shared model. Numbers match Sections 3.1–3.5; bars are schematic.

reasoning when predicting earlier tokens:

$$\begin{aligned}\pi_{i,t}^S(v) &= \pi_\theta(v \mid \mathcal{P}_S(\mathbf{x}_i), \mathbf{y}_{i,<t}), \\ \pi_{i,t}^T(v) &= \pi_\theta(v \mid \mathcal{P}_T(\mathbf{x}_i, \mathbf{y}_i), \mathbf{y}_{i,<t}),\end{aligned}\quad (2)$$

where $v \in \mathcal{V}$ is a vocabulary token, and $\pi_{i,t}^S$ and $\pi_{i,t}^T$ are the student and teacher next-token distributions, respectively.

3.2 Estimating reliability from self-judgment

The judge π_{θ_0} assesses each completed attempt to provide a quality signal for response weighting; the teacher provides token distributions for the student to match. For efficiency, we reuse the initial model as judge rather than rely on a larger external model. Given the question and attempt, the judge evaluates logical validity, internal consistency, factual plausibility, relevance, and completeness using one five-level rubric shared across questions (Table 4, Appendix B).

For question–attempt pair i , let $\ell_{i,k}^J$ denote the judge’s logit for rating $k \in \{1, \dots, 5\}$; the superscript J labels judge quantities. The judge is prompted to return one rating digit. We read its five rating-token logits and normalize them into a distribution, with j indexing the possible ratings in the denominator:

$$\pi_i^J(k) = \frac{\exp(\ell_{i,k}^J)}{\sum_{j=1}^5 \exp(\ell_{i,j}^J)}. \quad (3)$$

Judge score. We take the judge’s most probable rating as the quality score: 1 means fundamentally incorrect, and 5 means fully correct and well justified. We rescale the score to $[0, 1]$ for use in the response weight:

$$s_i = \arg \max_{k \in \{1, \dots, 5\}} \pi_i^J(k), \quad \bar{s}_i = \frac{s_i - 1}{4}. \quad (4)$$

Judge confidence. Judge confidence distinguishes attempts with the same score: the judge may strongly favor that rating for one attempt but be split between ratings for another. We quantify this distinction using one minus normalized rating entropy:

$$c_i^J = 1 - \frac{H(\pi_i^J)}{\log 5}, \quad (5)$$

where $H(p) = -\sum_j p(j) \log p(j)$ is Shannon entropy and $\log 5$ is the maximum entropy over five ratings. Confidence is zero for equally likely ratings and approaches one as probability concentrates on one rating. It measures decisiveness, not a calibrated probability of correctness.

Judged reliability. We combine the normalized judge score and confidence into judged reliability:

$$r_i^J = \bar{s}_i c_i^J. \quad (6)$$

The product is high only when the judge assigns a high score with high confidence; uncertain high

scores and confident low scores both receive lower reliability. Appendix B gives an example, and Section 4.3 relates judge score and confidence to externally assessed response quality.

3.3 Measuring student uncertainty

We use student uncertainty to prioritize attempts whose tokens the student predicts less confidently, including attempts judged reliable. We measure this uncertainty by averaging normalized next-token entropy over the generated response $\mathbf{y}_i$ of length $n_i = |\mathbf{y}_i|$:

$$u_i^S = \frac{1}{n_i} \sum_{t=1}^{n_i} \frac{H(\pi_{i,t}^S)}{\log |\mathcal{V}|}. \quad (7)$$

We use the full-vocabulary student distribution at the distillation temperature (Appendix B); prompt and padding positions are excluded. Normalization bounds u_i^S in $[0, 1]$, and averaging avoids an automatic increase with response length. A larger value indicates more diffuse token predictions along that response, rather than a direct estimate of whether its reasoning is correct.

3.4 Combining reliability and uncertainty

We combine the two signals to weight each attempt’s distillation loss:

$$w_i = \alpha r_i^J + (1 - \alpha) u_i^S, \quad \alpha \in [0, 1]. \quad (8)$$

The reliability term prioritizes attempts judged to be high quality with high confidence, while the uncertainty term prioritizes attempts for which the student’s token predictions are uncertain.

The additive rule lets either priority contribute independently: uncertainty can contribute when judged reliability is low, and reliability can contribute when the student is already certain. The endpoints $\alpha = 1$ and $\alpha = 0$ use reliability and uncertainty alone, respectively. We use $\alpha = 0.5$, which gives both signals equal coefficients; their realized contributions depend on their magnitudes. Section 4.5 evaluates both components and their balance.

3.5 Training objective

We train the student to match the teacher’s token distributions, scaling each response’s average forward-KL loss by its weight w_i :

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{x}_i, \mathbf{y}_i} \left[\frac{w_i}{n_i} \sum_{t=1}^{n_i} D_{\text{KL}}(\pi_{i,t}^T \parallel \pi_{i,t}^S) \right], \quad (9)$$

where the expectation is over questions from $\mathcal{D}$ and attempts drawn from Equation 1. Following OPSD (Zhao et al., 2026a), we optimize a coordinate-clipped surrogate of this loss, with gradients passing only through student predictions. Appendix B gives the clipping and model-update details.

4 Experiments

Our experiments examine reasoning performance, the judge’s quality signals, training dynamics, and design choices: **RQ1:** Can RUSD improve reasoning accuracy across tasks and model scales? **RQ2:** Do judge scores and confidence reflect response quality? **RQ3:** During training, does the number of responses with high judge scores and confidence increase, and does the percentage with low student uncertainty within this group also increase? **RQ4:** How do the weighting signals, their balance, and the loss type affect performance?

4.1 Experimental Setup

Models and Training Data. We evaluate Qwen3-4B and Qwen3-8B (Yang et al., 2025), with component ablations and hyperparameter studies on 4B. All methods share 30,000 NaturalReasoning questions (Yuan et al., 2025), sampled with seed 42 after filtering for non-empty reference answers so that reference-based baselines can use the same questions. RUSD learns from questions and its own online attempts; the dataset’s reference answers and generated responses are excluded from its teacher and judge prompts.

Baselines. We compare five representative supervision strategies on the same training questions. *Supervised fine-tuning (SFT)* (Ouyang et al., 2022) learns from the dataset’s Llama3.3-70B-Instruct responses. *GRPO* (Shao et al., 2024) uses scalar rewards from an external LLM judge. *RCSD* (Gu et al., 2026) supplies question-specific rubrics as teacher context. *OPSD* (Zhao et al., 2026a) uses the dataset’s Llama3.3-70B-Instruct responses as teacher context and matches teacher and student predictions along student-generated responses. *NSD* (Pei et al., 2026) generates negative conditions from model responses and suppresses behavior encouraged by the negatively conditioned teacher. Thus SFT, OPSD, RCSD, and GRPO rely on supervision beyond the training questions (marked with [†] in Table 1), whereas NSD and RUSD use only the questions and the model’s own outputs.

Method	Science and general reasoning					Mathematical reasoning					Logic	Overall
	GPQA	SciB	MMLU	LiveB	OBQA	GSM8K	MATH	AMC	Minerva	Olymp	Zebra	Average
Qwen3-4B												
Base	41.16	35.77	78.12	46.50	91.40	93.03	81.68	62.35	29.04	46.03	72.13	61.56
SFT [†]	42.30	35.91	78.19	42.00	91.00	93.18	80.50	54.22	29.41	44.51	72.10	60.30
GRPO [†]	43.81	43.89	78.85	47.00	93.00	93.78	81.86	63.86	31.25	47.92	72.58	63.44
RCSD [†]	41.41	42.56	78.66	49.50	92.80	93.25	81.64	61.45	30.15	45.99	71.53	62.63
OPSD [†]	51.14	46.32	77.49	47.50	91.60	94.24	84.48	71.08	29.04	52.67	69.15	64.97
NSD	51.14	61.81	79.22	47.62	93.00	94.22	81.83	62.65	29.78	45.96	72.50	65.43
RUSD	52.78	70.66	79.38	61.25	92.60	94.81	85.87	74.10	35.02	54.27	74.38	70.47
Qwen3-8B												
Base	62.12	75.72	82.04	70.12	93.70	96.17	87.20	80.12	34.10	58.05	81.13	74.59
SFT [†]	61.24	74.49	82.95	67.62	93.85	96.17	87.13	80.42	34.65	58.42	82.70	74.51
GRPO [†]	62.63	75.54	82.95	72.50	94.40	96.10	88.20	81.33	34.28	59.27	82.93	75.47
RCSD [†]	60.61	76.02	83.00	70.62	95.60	95.98	87.95	81.33	34.19	59.79	77.25	74.76
OPSD [†]	59.72	76.59	80.18	70.63	94.10	95.81	88.31	80.12	33.92	60.42	81.25	74.64
NSD	64.65	74.96	83.06	72.50	95.25	96.23	88.06	82.53	34.01	59.53	82.93	75.79
RUSD	64.90	77.13	83.58	73.13	95.70	97.52	88.48	83.13	36.03	60.87	83.12	76.69

Table 1: Accuracy (Avg@4, %) on eleven reasoning benchmarks. RUSD achieves the highest macroaverage at both model sizes. Average is the unweighted mean over benchmarks; bold marks the best score in each column within a model size, including ties. [†]Uses supervision beyond the training questions and the model’s own outputs: dataset responses (SFT, OPSD), generated rubrics (RCSD), or an external LLM judge (GRPO). GPQA: GPQA Diamond; SciB: SciBench; LiveB: LiveBench Reasoning; OBQA: OpenBookQA; Minerva: Minerva Math; Olymp: OlympiadBench; Zebra: ZebraLogic.

Benchmarks. We evaluate eleven benchmarks: GPQA Diamond (Rein et al., 2024), SciBench (Wang et al., 2024), MMLU (Hendrycks et al., 2021a), LiveBench Reasoning (White et al., 2025), and OpenBookQA (Mihaylov et al., 2018) for scientific and general reasoning; GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b), AMC (AI-MO, 2024), Minerva Math (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024) for mathematics; and ZebraLogic (Lin et al., 2025) for logic-grid reasoning.

Evaluation. We report Avg@4: correctness averaged over four generated responses per question and then across questions, expressed as a percentage. *Average* in Table 1 is the unweighted mean of the eleven benchmark scores. All accuracy gains are absolute differences on this 0–100 scale, calculated before rounding. Appendix B gives the decoding and scoring settings.

Implementation. All methods use NVIDIA A100 80GB GPUs and shared training settings. At both model sizes, results are reported after 150 training steps, except GRPO, which uses 500 steps as in RCSD’s GRPO setting. For RUSD, we train LoRA adapters (Hu et al., 2022) on four GPUs with $\alpha = 0.5$ and student and teacher prompts in thinking mode. Appendix B gives the hyperparam-

eters (Table 5), model updates, GRPO judge, and evaluation-step details.

4.2 Reasoning Performance (RQ1)

RUSD outperforms all five baselines in average accuracy at both model scales. In Table 1, RUSD reaches 70.47% macroaverage accuracy at 4B and 76.69% at 8B, exceeding the base models by 8.90 and 2.10 and NSD, the strongest evaluated baseline by average, by 5.04 and 0.90, respectively. It leads on ten of eleven benchmarks at 4B and all eleven at 8B, with gains spanning scientific and general reasoning, mathematics, and logic-grid reasoning. At 4B, notable results include 70.66% on SciBench versus NSD’s 61.81%, and 61.25% on LiveBench Reasoning, 11.75 above the best baseline (RCSD). OpenBookQA is the exception: 92.60% versus 93.00% for GRPO and NSD. At 8B, gains over the best baseline on each benchmark range from 0.10 to 1.38, despite the stronger starting model.

The gains also hold against external supervision: at 4B, RUSD exceeds OPSD by 5.49, while SFT on the same supplied solutions falls below the base model. At 8B, the four externally supervised baselines change the base average by -0.08 to $+0.88$, compared with RUSD’s $+2.10$. Thus, RUSD improves both tested models without supplied solu-

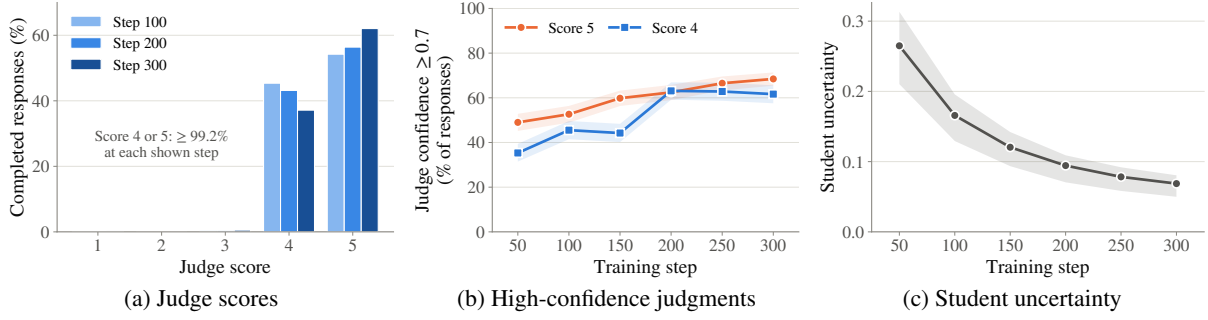


Figure 2: Judge assessments and student uncertainty during Qwen3-4B RUSD training with $\alpha = 0.5$. **(a)** Distribution of judge scores over completed responses in summaries reported at training steps 100, 200, and 300. **(b)** Percentage of high-confidence judgments ($c_i^J \geq 0.7$) within the score-4 and score-5 categories; shading shows 95% Wilson confidence intervals for these proportions. **(c)** Mean student uncertainty over completed score-4/5 responses, with shading spanning the 25th–75th percentiles.

Predictors	Multiple R	R^2 (%)	p -value
Judge score	0.46	21	8.9×10^{-12}
Judge confidence	0.62	39	5.0×10^{-23}
Score + confidence	0.77	59	7.2×10^{-39}

Table 2: Correlation of judge score and confidence with GPT scores (gpt-5.6sol quality ratings) on 200 responses. The multiple correlation coefficient R is the Pearson correlation between fitted and observed GPT scores; R^2 is in-sample explained variation. The p -values are from overall regression F-tests.

tions or question-specific rubrics. Every evaluated 4B training step from 50 to 300 also exceeds the strongest baseline average (Appendix C.2).

4.3 Judge Scores, Confidence, and Response Quality (RQ2)

Higher judge scores and confidence are associated with higher GPT scores. We ask gpt-5.6sol (OpenAI, 2026) to rate 200 randomly sampled saved training responses, one per question, using the same 1–5 quality rubric as the training judge. These ratings, termed *GPT scores*, serve only as an external measure of response quality and are not used to train RUSD. Table 2 reports linear regressions that estimate GPT scores from judge score, judge confidence, or both. In the joint regression, higher judge scores are associated with higher GPT scores at the same confidence, and higher judge confidence is associated with higher GPT scores at the same judge score.

We report the multiple correlation coefficient R , defined as the Pearson correlation between fitted and observed GPT scores. Using judge score and confidence together gives $R = 0.77$ and explains 59% of the in-sample variation in GPT scores,

compared with 21% for judge score alone and 39% for judge confidence alone. Adding confidence to score increases explained variation by $\Delta R^2 = 0.38$. Thus, judge confidence helps distinguish responses with the same judge score but different GPT scores.

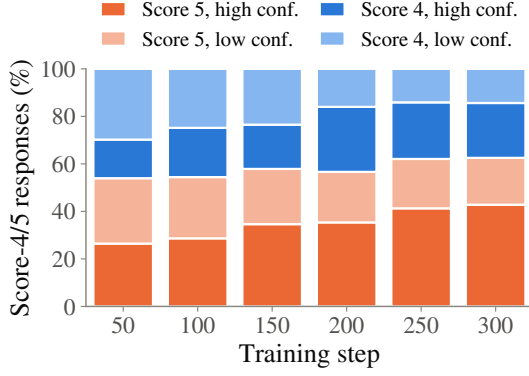
These results support combining judge score and confidence to assess response quality. Section 4.5 tests the product of normalized judge score and judge confidence as a training weight. GPT scores are external quality assessments, not verified correctness labels. Appendix B gives the regression definitions and coefficients.

4.4 Training Dynamics (RQ3)

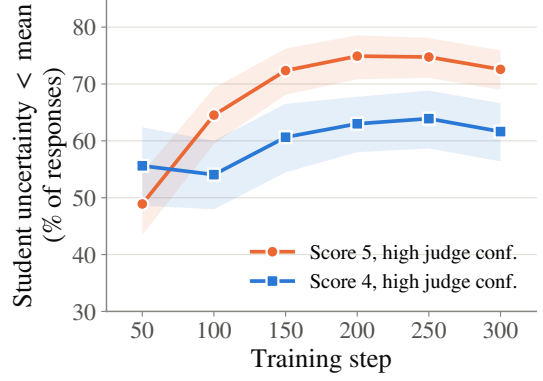
In this experiment, we analyze the Qwen3-4B run with $\alpha = 0.5$. Figures 2 and 3 show changes in judge assessments and student uncertainty during training; Table 8 in Appendix E provides counts and detailed comparisons.

At reported steps 100, 200, and 300, scores 4 and 5 account for at least 99.2% of completed responses, with score 5 becoming more frequent (Figure 2a). Score-only weighting therefore assigns almost all responses either 0.75 or 1.0. Confidence distinguishes attempts with the same score; RQ2 links higher confidence to higher GPT scores at the same judge score.

Both the number and percentage of responses with judge score 5 and high confidence increase. From step 50 to 300, this group grows from 319 to 638 responses, or 26.5% to 42.8% of completed score-4/5 responses (Figure 3a; Table 8). Within score 5, the high-confidence proportion ($c_i^J \geq 0.7$) rises from 49.0% to 68.5% (Figure 2b). The score-4/high-confidence group also grows as a share of



(a) Judge-score/confidence groups



(b) Low student uncertainty by group

Figure 3: Response-group composition and low student uncertainty during Qwen3-4B RUSD training with $\alpha = 0.5$. High judge confidence means $c_i^j \geq 0.7$. **(a)** Shares of completed score-4/5 responses in four groups defined by judge score and confidence. **(b)** Percentage of responses within each high-confidence group whose student uncertainty is below the pooled mean of all completed score-4/5 responses at that reporting step. Shading shows 95% Wilson confidence intervals from group response counts. Both panels use the same response population at each reporting step (Appendix B).

score-4/5 responses; its within-score proportion rises from 35.3% to 61.6%, with fluctuations.

Mean student uncertainty falls 3.86-fold from step 50 to 300. Across completed score-4/5 responses, mean u_i^S falls from 0.2649 to 0.0687, and the middle half of uncertainties also shifts downward (Figure 2c). Student predictions thus become more concentrated.

Within the score-5/high-confidence group, both the number and percentage with low student uncertainty increase overall. Here, low uncertainty means below the mean of all completed score-4/5 responses in the same summary. The count rises from 156 of 319 responses (48.9%) at step 50 to 463 of 638 (72.6%) at step 300 (Figure 3b; Table 8). The percentage peaks at 74.9% at step 200. This threshold measures relative uncertainty because the mean changes across summaries.

From step 100 onward, low student uncertainty is more frequent in high-confidence than low-confidence groups at both scores. At score 5, the percentages are similar at step 50 but reach 72.6% versus 39.8% at step 300 (Table 8). Among high-confidence responses, the below-mean percentage is also higher for score 5 than score 4 from step 100 onward (Figure 3b).

Together, rising judge confidence and falling student uncertainty are consistent with consolidation, but these summaries do not track individual attempts being learned. The two changes affect the response weight in opposite directions: higher judge confidence at a fixed score raises the rela-

bility contribution, while lower student uncertainty reduces the uncertainty contribution.

Further, accuracy reaches 70.47% at step 150, then falls despite declining student uncertainty and increasing high-confidence score-5 judgments (Figure 5, Appendix C.2). Greater predictive certainty and judge confidence therefore do not establish continued improvement in reasoning. A selected heat-calculation case illustrates retained correctness with increasing certainty: both responses are correct and receive score 5, while judge confidence rises from 0.572 to 0.771 and student uncertainty falls from 0.1728 to 0.0388. Table 7 gives the complete final responses; Appendix D gives the calculations and selection details.

4.5 Ablations and Hyperparameters (RQ4)

In this analysis, we study response weighting, loss type, and the weighting coefficient α on Qwen3-4B using the eleven-benchmark macroaverage. To assess what judge confidence and student uncertainty add beyond the judge score, our weighting ablations compare against score-only weighting rather than uniform weighting, which assigns the same loss weight to every attempt. Teacher context and forward KL remain fixed across the weighting variants. The loss ablations retain the full weighting rule and change the divergence. Table 3 summarizes the ablations; Table 6 in Appendix C.1 gives the per-benchmark comparison with score-only weighting.

Ablation studies. Adding judge confidence im-

Response weighting	Average
Score only: $w_i = \bar{s}_i$	68.44
Uncertainty only: $w_i = u_i^S$	69.93
Reliability only: $w_i = r_i^J$	70.12
RUSD: $w_i = 0.5r_i^J + 0.5u_i^S$	70.47
Loss type	Average
Forward KL	70.47
Reverse KL	69.85
Jensen–Shannon divergence	69.12

Table 3: Response-weighting and loss ablations on Qwen3-4B, reporting average accuracy (Avg@4, %) across eleven benchmarks. Weighting variants use the forward-KL configuration; loss variants use the full weighting rule at $\alpha = 0.5$. Jensen–Shannon divergence uses mixture coefficient 0.5.

proves average accuracy by 1.68. Score-only weighting, $w_i = \bar{s}_i$, reaches 68.44%; multiplying the normalized score by judge confidence, which gives reliability-only weighting $w_i = r_i^J = \bar{s}_i c_i^J$, raises accuracy to 70.12%. Score-only weighting assigns the same weight to all attempts with the same rating; confidence differentiates attempts within each score level. The gain is consistent with the quality audit, in which confidence explains additional variation in external ratings.

Combining reliability and uncertainty gives the highest average accuracy among all the weighting variants. Reliability-only weighting prioritizes assessed quality, while uncertainty-only weighting prioritizes diffuse student predictions. The full rule combines both priorities and reaches 70.47%, exceeding reliability-only weighting by 0.35 and uncertainty-only weighting by 0.54. It is also higher than score-only weighting on all eleven benchmarks, with a macroaverage gain of 2.03. Adding confidence gives the larger observed gain (1.68), while adding student uncertainty to reliability gives a smaller further gain in this single-run comparison. Both single-signal variants exceed the strongest baseline average, while their combination gives a further improvement.

Forward KL gives the highest average among the tested loss types. Response weights determine each attempt’s contribution, while the loss type determines how teacher and student token distributions are matched. With the full weighting rule and $\alpha = 0.5$, the forward-KL configuration reaches 70.47%, compared with 69.85% for reverse KL and 69.12% for Jensen–Shannon divergence with mixture coefficient 0.5.

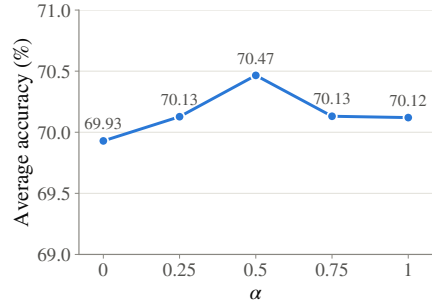


Figure 4: Effect of the mixing coefficient α on Qwen3-4B RUSD, measured by average accuracy across eleven benchmarks. All settings use the dynamic teacher. The endpoints, $\alpha = 0$ (uncertainty only) and $\alpha = 1$ (reliability only), match the corresponding runs in Table 3.

Hyperparameter study. $\alpha = 0.5$ gives the highest observed average accuracy. In Equation 8, $\alpha = 0$ gives uncertainty-only weighting and $\alpha = 1$ gives reliability-only weighting, matching the corresponding ablations in Table 3. Larger α assigns a higher coefficient to judged reliability, emphasizing quality supported by judge confidence; smaller α assigns a higher coefficient to remaining student uncertainty. Figure 4 reports 69.93%, 70.13%, 70.47%, 70.13%, and 70.12% for $\alpha = 0, 0.25, 0.5, 0.75$, and 1, respectively. The full range is 0.54, and all three intermediate settings reach at least 70.13%. The highest observed average favors using both signals in this configuration.

5 Conclusion

In this work, we address two challenges in self-distillation: obtaining teacher guidance without supplied reference solutions, and assessing self-generated attempts without reliable correctness feedback. RUSD uses the attempts as teacher context and weights distillation by judged reliability and student uncertainty. Across eleven benchmarks, it improves Qwen3-4B and Qwen3-8B macroaverage accuracy by 8.90 and 2.10 over the base models, respectively. At 4B, adding judge confidence improves average accuracy, and combining reliability with uncertainty gives the highest observed average among the weighting variants. External ratings support judge confidence as a quality signal, while training summaries show more high-confidence judgments alongside lower student uncertainty.

Limitations

Model coverage. Our experiments use Qwen3-4B and Qwen3-8B. Further evaluation across model families and sizes is needed to establish how well RUSD generalizes beyond these backbones.

Judge design. For efficiency, the judge assesses completed reasoning responses with thinking mode disabled, directly reading rating-token logits without generating an evaluation rationale. Whether enabling judge thinking or updating the judge improves assessment quality remains untested. Enabling thinking would add decoding cost and latency, depending on the reasoning budget; judge-update costs depend on whether updates reuse student parameters or require separate optimization. We also do not compare larger judges or judges from other model families within RUSD, so their effects on accuracy and cost remain untested.

Self-judgment and learning dynamics. The judge can confidently endorse incorrect responses and share biases with the generating model. Its confidence measures concentration over ratings, rather than a calibrated probability of correctness. The additive rule also assigns uncertainty-based weight when judged reliability is low. The 200-response audit measures in-sample agreement with automated quality ratings, rather than verified correctness. Training summaries follow changing response populations and describe associations rather than causal learning mechanisms. The later decline in accuracy despite falling entropy shows that greater predictive certainty does not guarantee better reasoning.

Component evidence. Component studies use Qwen3-4B. Judged reliability and student uncertainty both lie in $[0, 1]$; their interaction and relative influence on learning across training stages warrant deeper analysis.

Cost and scope. Each attempt requires a teacher forward pass conditioned on the complete attempt and a separate judge forward pass. The teacher does not generate a second response. We do not report total compute across all experiments or compare methods at matched compute budgets, and we do not analyze response length or truncation rates. Experiments use one NaturalReasoning training selection, English questions, and thinking mode for response generation. All benchmarks have automatically checkable answers, so performance on

open-ended tasks remains untested.

Ethical Considerations

This study uses publicly available reasoning resources and model-generated responses. NaturalReasoning is released under CC BY-NC 4.0, and the Qwen3-4B and Qwen3-8B backbones under Apache 2.0.¹ The dataset’s Llama3.3-70B-Instruct responses, which only the SFT and OPSD baselines use, carry additional obligations under the [Llama 3.3 Community License](#) when used to train distributed models. Reuse and redistribution should respect these licenses and the terms of individual evaluation resources, including the training dataset’s noncommercial restriction.

Self-distillation can reinforce factual errors and social biases inherited from the model or its training data. The teacher and judge share a pretrained backbone, so errors may persist even when judge confidence increases and student uncertainty decreases. The proposed weighting signals are fallible assessments, so more confident responses should not be treated as verified answers.

The external quality check uses automated gpt-5.6sol ratings of 200 responses. It provides evidence about assessed response quality, rather than human verification or a safety evaluation. Our experiments do not measure fairness, privacy, or robustness to harmful requests. Deployment, especially in high-stakes settings, would require separate assessment and safeguards appropriate to the intended use.

References

- Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. 2024. [On-policy distillation of language models: Learning from self-generated mistakes](#). In *The Twelfth International Conference on Learning Representations*.
- Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. 2025. [The unreasonable effectiveness of entropy minimization in LLM reasoning](#). In *Advances in Neural Information Processing Systems*, volume 38, pages 107150–107180. Curran Associates, Inc.
- AI-MO. 2024. [AIMO Validation AMC](#). Hugging Face dataset.

¹See the official [NaturalReasoning dataset card](#) and [Qwen3-4B/Qwen3-8B model cards](#).

- Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, and 3 others. 2021. [A general language assistant as a laboratory for alignment](#). *Preprint*, arXiv:2112.00861.
- Qiye Cai, Yichuan Ma, Peiji Li, Yongkang Chen, Qipeng Guo, Yicheng Zou, Linyang Li, Xiaocheng Feng, and Bing Qin. 2026. [H²SD: Hybrid hindsight self-distillation](#). *Preprint*, arXiv:2607.18955.
- Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. 2023. [SoftMatch: Addressing the quantity-quality trade-off in semi-supervised learning](#). In *The Eleventh International Conference on Learning Representations*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, et al. 2025. [DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning](#). *Preprint*, arXiv:2501.12948.
- Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, KaShun Shum, and Tong Zhang. 2023. [RAFT: Reward ranked finetuning for generative foundation model alignment](#). *Transactions on Machine Learning Research*.
- Yangyang Feng, Zhuoyan Feng, and Junlan Chen. 2026. [PAST: Privileged adaptation from complete student trajectories for on-policy self-distillation](#). *Preprint*, arXiv:2608.08726.
- John Gkountouras, Josip Jukić, and Ivan Titov. 2026. [Consensus as privileged context for label-free self-distillation](#). *Preprint*, arXiv:2607.13643.
- Siye Gu, Jialin Chen, Sophia Zhou, Arman Cohan, and Rex Ying. 2026. [Rethinking reward supervision: Rubric-conditioned self-distillation](#). In *Third Conference on Language Modeling*.
- Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. 2023. [Reinforced self-training \(ReST\) for language modeling](#). *Preprint*, arXiv:2308.08998.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. [On calibration of modern neural networks](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.
- Jiaxin Guo, Yanwei Yue, Xuanbo Fan, Chunyu Yang, and Yan Zhang. 2026. [Learning from consensus and disagreement: Unsupervised on-policy self-distillation with minority-trajectory contrast](#). *Preprint*, arXiv:2608.08764.
- Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. [OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand. Association for Computational Linguistics.
- Yinghui He, Simran Kaur, Adithya Bhaskar, Yongjin Yang, Jiarui Liu, Narutatsu Ri, Liam Fowl, Abhishek Panigrahi, Danqi Chen, and Sanjeev Arora. 2026. [Self-distillation zero: Self-revision turns binary rewards into dense supervision](#). *Preprint*, arXiv:2604.12002.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. [Measuring massive multitask language understanding](#). In *International Conference on Learning Representations*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. [Measuring mathematical problem solving with the MATH dataset](#). In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. [Distilling the knowledge in a neural network](#). *Preprint*, arXiv:1503.02531.
- Md. Ismail Hossain, Humaira Kousar, and Isidora Chara Tourni. 2026. [Overcoming scaling limits in on-policy self-distillation for LLM reasoning](#). *Preprint*, arXiv:2609.37915.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Di Huang, Hao Li, Yixin Chen, and Fuhai Li. 2026. [When clipping reverses correction: Failure dynamics of pointwise forward-KL on-policy self-distillation](#). *Preprint*, arXiv:2609.38995.
- Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. [Large language models can self-improve](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, Singapore. Association for Computational Linguistics.

- Jonas Hübner, Frederike Lübeck, Lejs Deen Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. 2026. [Reinforcement learning via self-distillation](#). In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 48677–48721. PMLR.
- Yuki Ichihara, Naoto Iwase, Mohammad Atif Quamar, and Junpei Komiyama. 2026. [Privileged solutions or context-induced teacher behavior? dissecting on-policy self-distillation](#). *Preprint*, arXiv:2608.09228.
- Junlong Ke, Zichen Wen, Weijia Li, Conghui He, and Linfeng Zhang. 2026. [Respecting self-uncertainty in on-policy self-distillation for efficient LLM reasoning](#). *Preprint*, arXiv:2605.13255.
- Yoon Kim and Alexander M. Rush. 2016. [Sequence-level knowledge distillation](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.
- Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. [Solving quantitative reasoning problems with language models](#). In *Advances in Neural Information Processing Systems*, volume 35.
- Juanhui Li, Sreyashi Nag, Hui Liu, Xianfeng Tang, Sheikh Muhammad Sarwar, Limeng Cui, Hansu Gu, Suhang Wang, Qi He, and Jiliang Tang. 2025. [Learning with less: Knowledge distillation from large language models via unlabeled data](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2627–2641, Albuquerque, New Mexico. Association for Computational Linguistics.
- Yijiang Li, Bingyang Wang, Yijun Liang, Yunjie Tian, Di Fu, and Nuno Vasconcelos. 2026. [On-policy self-distillation without any supervision](#). *Preprint*, arXiv:2608.06296.
- Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. 2025. [ZebraLogic: On the scaling limits of LLMs for logical reasoning](#). In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 37889–37905. PMLR.
- Xiaogeng Liu, Xinyan Wang, Yingzi Ma, Yechao Zhang, and Chaowei Xiao. 2026. [When are teacher tokens reliable? position-weighted on-policy self-distillation for reasoning](#). *Preprint*, arXiv:2605.21606.
- David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. 2016. [Unifying distillation and privileged information](#). In *International Conference on Learning Representations*.
- Zhexi Lu, Subhajit Chaudhury, Tejaswini Pedapati, Keerthiram Murugesan, and Lei Yu. 2026. [Activation-conditioned self-distillation](#). *Preprint*, arXiv:2609.38342.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. [Self-Refine: Iterative refinement with self-feedback](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 46534–46594. Curran Associates, Inc.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. [Can a suit of armor conduct electricity? a new dataset for open book question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.
- Subhabrata Mukherjee and Ahmed Awadallah. 2020. [Uncertainty-aware self-training for few-shot text classification](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 21199–21212. Curran Associates, Inc.
- Andrei Liviu Nicolicioiu, Mohammad Pezeshki, and Aaron Courville. 2026. [On-policy self-distillation with sampled demonstrations reduces output diversity](#). *Preprint*, arXiv:2606.26091.
- OpenAI. 2026. [GPT-5.6 system card](#).
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.
- Arjun Panickssery, Samuel R. Bowman, and Shi Feng. 2024. [LLM evaluators recognize and favor their own generations](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 68772–68802. Curran Associates, Inc.
- Rongcan Pei, Zhepei Wei, Shuyao Xu, Xinyu Zhu, Wei-Lin Chen, and Yu Meng. 2026. [Negative self-distillation: Learning to reason by avoiding flaws](#). *Preprint*, arXiv:2609.11699.
- Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. 2025. [Maximizing confidence alone improves reasoning](#). *Preprint*, arXiv:2505.22660.
- Qwen Team. 2026. [Qwen3.8-27B model card](#).

- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. [GPQA: A graduate-level Google-proof Q&A benchmark](#). In *First Conference on Language Modeling*.
- MohammadHossein Rezaei, Anas Mahmoud, Zihao Wang, Utkarsh Tyagi, Advait Gosai, Razvan-Gabriel Dumitru, Aakash Sabharwal, Bing Liu, and Yunzhong He. 2026. [Rubric-guided self-distillation: Post-training without rubric verifiers](#). *Preprint*, arXiv:2606.12507.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [DeepSeekMath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.
- Idan Shenfeld, Mehul Damani, Jonas Hübner, and Pulkit Agrawal. 2026. [Self-distillation enables continual learning](#). In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 111542–111558. PMLR.
- Samyak Shrestha and Alexander Tessier. 2026. [Re-thinking privileged information in on-policy self-distillation](#). *Preprint*, arXiv:2608.18271.
- Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron T Parisi, Abhishek Kumar, Alexander A Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Fathy Elsayed, Hanie Sedghi, and 22 others. 2024. [Beyond human data: Scaling self-training for problem-solving with language models](#). *Transactions on Machine Learning Research*. Expert Certification.
- Charlie Snell, Dan Klein, and Ruiqi Zhong. 2022. [Learning by distilling context](#). *Preprint*, arXiv:2209.15189.
- Vladimir Vapnik and Akshay Vashist. 2009. [A new learning paradigm: Learning using privileged information](#). *Neural Networks*, 22(5-6):544–557.
- Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. [Replacing judges with juries: Evaluating LLM generations with a panel of diverse models](#). *Preprint*, arXiv:2404.18796.
- Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2024. [SciBench: Evaluating college-level scientific problem-solving abilities of large language models](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 50622–50649. PMLR.
- Bowen Wei, Nan Wang, Yuqing Zhou, Jinhao Pan, and Ziwei Zhu. 2026. [COSE: Confidence-orchestrated self-evolution for effective LLM reasoning](#). In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. To appear, main conference.
- Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddhartha Venkat Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. 2025. [LiveBench: A challenging, contamination-limited LLM benchmark](#). In *The Thirteenth International Conference on Learning Representations*.
- Fengyu Xie, Yilun Zhao, Bingsen Chen, Arman Cohan, and Chen Zhao. 2026. [Small language models as judges for rubric-based reinforcement learning](#). *Preprint*, arXiv:2608.30005.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Tianzhu Ye, Li Dong, Guanheng Chen, He Zhu, Xun Wu, Shaohan Huang, and Furu Wei. 2026a. [LLM-as-a-coach: Experiential learning for non-verifiable tasks](#). *Preprint*, arXiv:2607.18110.
- Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. 2026b. [On-policy context distillation for language models](#). *Preprint*, arXiv:2602.12275.
- Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. 2024. [Self-rewarding language models](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 57905–57923. PMLR.
- Weizhe Yuan, Jane Yu, Song Jiang, Karthik Padthe, Yang Li, Dong Wang, Ilya Kulikov, Kyunghyun Cho, Yuandong Tian, Jason Weston, and Xian Li. 2025. [NaturalReasoning: Reasoning in the wild with 2.8M challenging questions](#). In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc. Datasets and Benchmarks Track.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. [STaR: Bootstrapping reasoning with reasoning](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488. Curran Associates, Inc.
- Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. 2026a. [Self-distilled reasoner: On-policy self-distillation for large language models](#). In *Proceedings of the 43rd*

International Conference on Machine Learning, volume 306 of *Proceedings of Machine Learning Research*, pages 162433–162448. PMLR.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. 2026b. [Learning to reason without external rewards](#). In *The Fourteenth International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. 2023. [Judging LLM-as-a-judge with MT-Bench and Chatbot Arena](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc. Datasets and Benchmarks Track.

A Related Work

Distillation and self-training. Knowledge distillation transfers teacher predictions to a student (Hinton et al., 2015; Kim and Rush, 2016); on-policy distillation applies this guidance along student-generated responses (Agarwal et al., 2024). Context distillation transfers behavior elicited by additional teacher context (Askell et al., 2021; Snell et al., 2022; Ye et al., 2026b), connecting to learning with privileged information (Vapnik and Vashist, 2009; Lopez-Paz et al., 2016). Self-training selects generated responses using consistency, correctness checks, or rewards (Huang et al., 2023; Zelikman et al., 2022; Gulcehre et al., 2023; Dong et al., 2023; Singh et al., 2024). Self-Refine instead revises responses at inference time, whereas self-rewarding models construct preferences for training (Madaan et al., 2023; Yuan et al., 2024).

Sources of teacher guidance. SDFT uses demonstrations for continual learning, and OPSD uses reference solutions for reasoning (Shenfeld et al., 2026; Zhao et al., 2026a). RCSD first trains a rubric generator through a teacher given reference answers, then supplies its rubrics to the reasoning teacher (Gu et al., 2026). RGSD provides rubrics to a frozen teacher and trains a question-only student (Rezaei et al., 2026). Experiential Learning obtains textual coaching feedback for non-verifiable tasks (Ye et al., 2026a). Recent analyses find that OPSD’s gains need not depend on the paired reference solution, because solutions to other problems can provide comparable teacher context (Ichihara et al., 2026; Shrestha and Tessier, 2026). These sources of teacher guidance differ from RUSD’s use of the completed attempt itself as additional teacher context.

Attempts, feedback, and reference-free supervision. SDPO constructs teacher context from textual feedback, or successful responses when only scalar rewards are available (Hübotter et al., 2026). H²SD combines verified successful attempts with reference hints for failures, while PAST uses complete attempts, reference solutions, and correctness-based routing (Cai et al., 2026; Feng et al., 2026). SD-Zero trains a reviser using binary correctness feedback in mathematics and coding, then distills its predictions into the generator (He et al., 2026). Activation-Conditioned Self-Distillation uses verified outcomes to construct teacher-steering activation contrasts (Lu et al., 2026). OASIS conditions the teacher on another unverified attempt and uses final-answer labels (Hossain et al., 2026), and conditioning the teacher on a sampled correct response can reduce output diversity (Nicoliciu et al., 2026). Without supplied solutions, Canon and u-OPSD condition the teacher on a self-generated solution that reaches the majority answer over multiple samples; CoDA additionally penalizes minority trajectories (Gkountouras et al., 2026; Li et al., 2026; Guo et al., 2026). NSD suppresses tokens favored by negative conditions generated from its own responses (Pei et al., 2026). Unlike these methods, RUSD uses the completed attempt itself as additional teacher context, without correctness feedback, a reference solution, or answer consensus, and weights that attempt by judged reliability and student uncertainty.

Confidence and uncertainty. Confidence and uncertainty guide pseudo-label selection and weighting (Chen et al., 2023; Mukherjee and Awadallah, 2020). LLKD’s weighted variant adds normalized teacher confidence and student uncertainty to scale selected classification losses (Li et al., 2025). For reasoning, EGRSD attenuates token updates using teacher entropy, while COSE weights PPO updates and prioritizes replay using evaluator confidence (Ke et al., 2026; Wei et al., 2026). PW-OPSD weights reference-conditioned distillation by token position, motivated by diagnostics of teacher-token reliability (Liu et al., 2026). Other approaches optimize predictive confidence through entropy minimization or confidence rewards (Agarwal et al., 2025; Prabhudesai et al., 2025; Zhao et al., 2026b). RUSD distinguishes confidence in judging an attempt’s quality from uncertainty in predicting its tokens; evaluator bias and calibration remain relevant to interpreting both sig-

nals (Zheng et al., 2023; Panickssery et al., 2024; Guo et al., 2017).

Scope of comparison. Our experiments compare representative supervision strategies on fixed NaturalReasoning questions. RGSD shares OPSD’s teacher-conditioning principle, using rubrics in place of reference solutions (Rezaei et al., 2026). We include OPSD with NaturalReasoning’s Llama-generated responses as teacher context and rubric-conditioned RCSD as representative baselines, without separately evaluating RGSD. Verification-based methods require correctness feedback, while answer consensus requires rules for matching free-form answers. LLKD selects and weights pseudo-labels in a fixed classification label space, and COSE jointly generates tasks and updates its agents. Entropy-minimization and self-certainty methods also require only questions (Agarwal et al., 2025; Zhao et al., 2026b); because student entropy decreases during RUSD training, they are a natural control that we leave to future comparison. Other demonstration-, rubric-, feedback-, and uncertainty-based methods remain applicable or adaptable alternatives outside the present comparison.

B Implementation Details

Algorithm 1 RUSD training procedure

Require: Questions $\mathcal{D}$, initial model π_{θ_0} , coefficient α
Ensure: Trained reasoning model π_{θ}

- 1: Initialize $\theta \leftarrow \theta_0$; retain π_{θ_0} as judge
- 2: **for** each sampled question $\mathbf{x}_i \in \mathcal{D}$ **do**
- 3: Generate attempt $\mathbf{y}_i$ using current π_{θ} (Eq. 1)
- 4: Compute student predictions $\pi_{i,t}^S$ (Eq. 2)
- 5: Compute teacher predictions $\pi_{i,t}^T$ (Eq. 2)
- 6: Obtain $\bar{s}_i$ and c_i^J (Eqs. 3–5)
- 7: Compute judged reliability r_i^J (Eq. 6)
- 8: Compute student uncertainty u_i^S (Eq. 7)
- 9: Form weight w_i (Eq. 8)
- 10: Update θ (Section 3.5)
- 11: **end for**
- 12: **return** π_{θ}

Prompts. Table 4 lists the three prompts. Student and teacher prompts use the model’s thinking mode, and the judge prompt does not. The teacher and judge receive the same decoded attempt.

Teacher and judge. Generation, teacher prediction, and student prediction use the same current LoRA adapters (Hu et al., 2022). Gradients through student predictions update these shared adapters, so the teacher also uses the updated parameters on the next step. We refer to this as a dynamic teacher;

Student (user turn)

Question: {question}

Please reason step by step, and put your final answer within \boxed{ }.

Teacher (user turn)

Question: {question}

Here is a candidate analysis that may contain useful or misleading information:

=== Candidate Analysis Begin === {attempt} ===
Candidate Analysis End ===

Solve the problem independently using any useful information from the candidate. Please reason step by step, and put your final answer within \boxed{ }.

Judge (system turn)

You are an impartial evaluator of reasoning responses. Evaluate logical validity, internal consistency, factual plausibility, relevance, and completeness. Use this rubric strictly: 1 = fundamentally incorrect, irrelevant, or nonsensical; 2 = contains major errors that make the conclusion unreliable; 3 = partially correct but contains an important error, omission, or incomplete conclusion; 4 = mostly correct and complete, with only minor issues that do not invalidate the conclusion; 5 = fully correct, complete, relevant, and well justified. A response that omits a requested final answer, conclusion, or subpart is incomplete and must not receive a score above 3.

Judge (user turn)

Question: {question} Response: {attempt}

Apply the 1-to-5 rubric above. Return exactly one digit: 1, 2, 3, 4, or 5.

Table 4: Student, attempt-conditioned teacher, and judge prompts used by RUSD. The judge applies a common five-level quality rubric.

a fixed teacher retains the initial parameters. The judge uses the initial backbone with adapters disabled and reads the five score-token logits after its prompt, without generating text. For each update, teacher predictions and response weights are computed without gradient tracking; generation and judging are also outside the differentiated loss.

Judged reliability example. A judge that splits its probability between ratings 4 and 5 as 0.44/0.56 assigns score 5 ($\bar{s}_i = 1$) but has confidence of only about 0.57, so $r_i^J \approx 0.57$; a decisive rating of 5 gives reliability close to 1.

Prediction settings. The generation notation in Equation 1 denotes one attempt from π_{θ} using the temperature and top- p /top- k settings in Table 5. Student and teacher predictions in Equation 2 instead use a full-vocabulary softmax with logits divided by the distillation temperature. Student entropy uses this same full-vocabulary distribution,

Parameter	Setting
Optimizer	AdamW
Learning rate	5×10^{-6}
Epoch budget (maximum)	3
Effective batch size	32
Training steps for reported results	150
Attempts per sampled question	1
Generation temperature	1.2
Sampling top- p / top- k	0.95 / 20
Max. response length (tokens)	8,192
Max. sequence length (tokens)	32,768
Distillation temperature τ_D	1.2
Vocabulary-coordinate cap κ	0.05

Table 5: Key training hyperparameters for the main Qwen3-4B and Qwen3-8B RUSD runs.

without the generation filters. The judge’s scoring logits in Equation 3 are its five rating-token logits divided by the judge temperature, which is 1.0 in the main configuration.

Loss implementation. We use the OPSD implementation (Zhao et al., 2026a), which caps each teacher-weighted log-probability-ratio contribution at $\kappa = 0.05$ before summing over vocabulary coordinates. This is a coordinate-clipped surrogate of the forward-KL objective in Equation 9, rather than exact KL; the cap can change the update direction (Hossain et al., 2026; Huang et al., 2026). Distillation and student entropy use the full output vocabulary without generation sampling filters.

Training configuration. Table 5 collects the main RUSD settings for both model sizes. We use linear learning-rate decay. The effective batch size is 32 questions, combining one question per GPU with eight gradient accumulation steps across four GPUs. The epoch budget is a configured maximum; the main results are reported after 150 training steps. The coefficient study varies the response-weight setting with the same dynamic teacher used in the main configuration. One RUSD training run took approximately 2 days on four NVIDIA A100 80 GB GPUs.

Masks and reduction. The implementation excludes prompt and padding tokens from entropy and loss calculations. Equations 7 and 9 sum over the unpadded response positions of retained responses. Responses that reach the 8,192-token generation limit are excluded from distillation and receive zero weight. A response with no valid tokens contributes no loss and has $u_i^S = 0$. The main configuration uses one question per device micro-

batch, so the valid-token-normalized loss is the per-response average in Equation 9. The implementation sums weighted token losses and divides by the microbatch’s valid-token count before gradient accumulation and distributed averaging. With larger microbatches, this is token-count weighting rather than an equal average of response means. Algorithm 1 presents the per-question computation; optimizer updates follow the accumulation schedule described above. No gradient passes through generation, judging, teacher predictions, or weights.

Data. We use the public 1.1M-question release of NaturalReasoning (Yuan et al., 2025). Each row pairs a question with a reference answer derived from its source document, when one exists, and a response generated by Llama3.3-70B-Instruct. All methods we train share one selection: we keep rows whose reference-answer field is non-empty, shuffle with seed 42, and take the first 30,000 questions. In RUSD, the reference field is used only for this selection, and neither dataset answer field is used in training prompts. The student prompt contains the question and task instructions; teacher and judge prompts also include the model’s own completed attempt.

GRPO reward. For GRPO, scalar response-quality rewards are supplied by a frozen external Qwen3.8-27B judge (Qwen Team, 2026).

Training steps for reported results. At both Qwen3-4B and Qwen3-8B, the main SFT, RCSD, OPSD, NSD, and RUSD results are reported after 150 training steps, and GRPO results after 500 training steps. Training steps denote optimizer updates. For RUSD, the reporting step was fixed at 150 independently of the checkpoint evaluations. The additional evaluations in Appendix C.2 characterize performance over training.

Evaluation decoding. We use temperature 1.0, top- $p = 0.95$, top- $k = -1$ (disabled), min- $p = 0$, and presence penalty 0. We enable thinking mode and generate four responses per question. These evaluation settings differ from the training generation settings in Table 5.

Benchmark scoring. Benchmark-specific automatic scoring compares parsed multiple-choice answers with annotated options and extracted mathematical answers with references using numerical or symbolic comparison. ZebraLogic requires every grid assignment to be correct. These scores

use neither the training judge nor an external LLM judge.

Training-dynamics summaries and intervals.

Figures 2(b–c) and 3 use the same reported summaries of completed score-4/5 responses. Figure 2(a) uses the corresponding saved summaries at steps 100, 200, and 300, including all completed responses with scores 1–5. For Figure 2(b), the denominator is all responses with the corresponding judge score, and the numerator counts high-confidence judgments. For Figure 3(b), n is the judge-group response count and k counts responses with student entropy below the pooled mean of the corresponding summary. Both proportion plots use 95% Wilson confidence intervals with $z = 1.96$. For example, the score-5/high-confidence group at step 100 has $k = 231$ and $n = 358$, giving 64.5% and an interval of 59.4–69.3%; at step 300, 463/638 gives 72.6% with an interval of 69.0–75.9%. These intervals describe response proportions, rather than variation across independent training runs. The band in Figure 2(c) instead spans the 25th–75th percentiles of response entropy; it is not a confidence interval.

External quality audit. We randomly sample 200 saved question–response pairs and ask gpt-5.6sol (OpenAI, 2026) to act as an external judge using the same five-level quality rubric in Table 4. Each of the 200 questions contributes one response and one observation. The three regressions include an intercept and predict the external GPT rating from the training judge’s score, confidence, or both. We report descriptive in-sample fit and coefficient estimates. The multiple correlation coefficient $R = \sqrt{R^2}$ is the correlation between fitted and observed GPT ratings. Its square, R^2 , is the fraction of external-rating variation explained within the fitted sample. R is nonnegative; the signs of the regression coefficients give the direction of the fitted conditional associations. The joint coefficients are $b_{\text{score}} = 0.84$ per rating level and $b_{\text{confidence}} = 3.78$. Thus, a 0.1 increase in confidence is associated with a 0.378 increase in the external rating at the same judge score; an increase of 1 in the judge score is associated with a 0.84 increase in the rating at the same confidence. These estimates describe response-quality ratings, rather than verified correctness or calibrated probabilities.

C Additional Experimental Results

C.1 Extended weighting results

Benchmark	Score only	RUSD	Δ
GPQA Diamond	52.40	52.78	+0.38
SciBench	66.11	70.66	+4.55
MMLU	78.12	79.38	+1.26
LiveBench Reasoning	58.50	61.25	+2.75
OpenBookQA	91.75	92.60	+0.85
GSM8K	94.56	94.81	+0.25
MATH	83.47	85.87	+2.40
AMC	71.39	74.10	+2.71
Minerva Math	32.08	35.02	+2.94
OlympiadBench	51.41	54.27	+2.86
ZebraLogic	73.02	74.38	+1.36
Macroaverage	68.44	70.47	+2.03

Table 6: Qwen3-4B accuracy (Avg@4, %) under score-only and RUSD weighting. Δ is RUSD’s absolute gain over score-only weighting on the same 0–100 scale. RUSD improves all eleven benchmarks.

Table 6 reports the complete benchmark comparison between score-only weighting and RUSD. The full rule is higher on each benchmark, including scientific reasoning, mathematics, and logic-grid reasoning, with single-run differences below 1 on several benchmarks. This comparison changes confidence and student uncertainty together; the endpoint comparisons in Section 4.5 examine reliability-only and uncertainty-only weighting.

Sensitivity to SciBench. The 4B gain is largest on SciBench, where the base model starts at 35.77% and RUSD reaches 70.66%. Removing this benchmark from the macroaverage still leaves RUSD ahead of the base model by 6.30 and the strongest remaining baseline, OPSD, by 3.61. Thus, the overall advantage is not confined to SciBench.

C.2 Performance across training steps

Accuracy rises from 67.78% at step 50 to 70.47% at step 150, remains close at step 200 (70.34%), and falls to 67.96% at step 300 (Figure 5). Figure 2(c) shows that student entropy continues to fall as this accuracy curve declines. Entropy measures predictive uncertainty, whereas benchmark accuracy measures task performance. Their divergence shows why the learning-priority signals should be interpreted alongside the benchmark curve. Training step 150 gives the highest observed reasoning performance, rather than the lowest student entropy, and every evaluated training step exceeds the strongest baseline, NSD (65.43%).

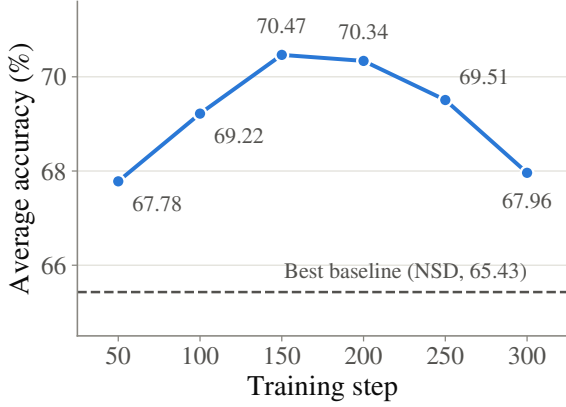


Figure 5: Average accuracy (Ave@4, %) across eleven benchmarks for Qwen3-4B RUSD with $\alpha = 0.5$ at six evaluated training steps from 50 to 300. The highest observed value is 70.47% at step 150. The dashed line marks the strongest Qwen3-4B baseline average (NSD, 65.43%; Table 1).

D Retaining a Correct Solution with Increasing Certainty

Table 7 reproduces the complete saved final responses to the same heat-calculation question with $\alpha = 0.5$. They are observations at training steps 30 and 105, rather than responses regenerated for separate evaluations at those training steps. Both responses contain the correct two-stage calculation. Their shared correctness makes this case useful for examining changes that the discrete judge score alone does not capture. We compare the logged training signals and verify the saved final calculations below.

Checking both solutions. The solid must first reach its melting point and then undergo the phase change. Heating requires $mc\Delta T = 2.5 \times 390 \times (90 - 40) = 48,750$ J. Melting requires $mL_f = 2.5 \times 4,000 = 10,000$ J. The total is therefore 58,750 J, matching the supplied reference answer. Both saved final responses explicitly apply these formulas and obtain these values. A temperature difference of 50°C equals 50 K, so their temperature conversion is also correct. This check concerns the saved final explanations shown in the case record.

What changes when correctness is retained?

The judge score remains 5, while the rating distribution becomes more concentrated: the probabilities assigned to scores 4 and 5 change from approximately 0.438/0.562 to 0.119/0.879. Judge confidence consequently rises from 0.572 to 0.771.

Meanwhile, mean normalized student entropy falls from 0.1728 to 0.0388, approximately a 4.5-fold reduction. The later response preserves the same correct reasoning with less predictive uncertainty; it also states the energy unit next to its final boxed value. The fixed score thus conceals two changes with different effects on the additive weight: greater judged reliability raises its reliability contribution, while lower student uncertainty reduces its uncertainty contribution. This is the response-level pattern motivating learning from a correct attempt whose generation and assessment can still become more certain.

Selection and interpretation. The case was selected post hoc through a non-blinded review of repeated training observations. Reference answers were used for this selection and the subsequent correctness check, not for training supervision. The displayed case was chosen after checking its full saved final explanations for correctness at both observations and requiring an unchanged judge score.

The case illustrates retained correctness with increasing certainty; it neither identifies an initial random guess nor isolates the cause of the change.

E Additional Training-Dynamics Analysis

Table 8 gives the shared counts used in the training-summary analysis. Figure 2(b) divides the high-confidence group size by all responses with the same judge score. Figure 3(a) divides each group size by all completed score-4/5 responses, while Figure 3(b) reports k/n for the two high-confidence groups. For the score-5/high-confidence group at step 300, these proportions are $638/932 = 68.5\%$, $638/1490 = 42.8\%$, and $463/638 = 72.6\%$, respectively. Thus, the panels describe different proportions of the same response population. Figure 2(a) uses the same saved summaries at steps 100, 200, and 300 but includes all completed responses: 1,254, 1,369, and 1,502, respectively. Their score-4/5 subsets contain 1,249, 1,363, and 1,490 responses, matching Table 8.

Score distributions and group composition.

The score-5 share of completed responses rises from 54.2% at reported step 100 to 62.1% at step 300 (Figure 2a). Among completed score-4/5 responses, the score-5/high-confidence share grows from 26.5% at reported step 50 to 42.8% at step 300. The score-4/high-confidence share grows from 16.3% to 23.1%, reaching its maximum of

Question. A solid has a melting point of 90.0°C . Calculate the total heat required to change 2.5 kg of this solid from 40.0°C to a liquid, given that the specific heat of the solid is $390\text{ J/kg}\cdot\text{K}$ and its heat of fusion is 4000 J/kg . Show your work and explain your reasoning.

Earlier response (step 30)

To determine the total heat required to change 2.5 kg of a solid from 40.0°C to a liquid, we need to account for two distinct processes:

1. Heating the solid from its initial temperature to its melting point
2. Melting the solid at its melting point into a liquid

Step 1: Heating the solid to its melting point

The temperature difference between the initial temperature and the melting point is:

$$\Delta T = 90.0^{\circ}\text{C} - 40.0^{\circ}\text{C} = 50.0^{\circ}\text{C}$$

The heat required to raise the temperature of the solid is calculated using the specific heat formula:

$$Q_1 = m \cdot c \cdot \Delta T$$

Where:

- $m = 2.5\text{ kg}$
- $c = 390\text{ J/kg}\cdot\text{K}$
- $\Delta T = 50.0\text{ K}$ (since a change of $1^{\circ}\text{C} = 1\text{ K}$)

$$\begin{aligned} Q_1 &= 2.5\text{ kg} \cdot 390\text{ J/kg}\cdot\text{K} \cdot 50.0\text{ K} \\ &= 48750\text{ J} \end{aligned}$$

Step 2: Melting the solid at its melting point

The heat required to melt the solid is calculated using the heat of fusion formula:

$$Q_2 = m \cdot L_f$$

Where:

- $m = 2.5\text{ kg}$
- $L_f = 4000\text{ J/kg}$

$$\begin{aligned} Q_2 &= 2.5\text{ kg} \cdot 4000\text{ J/kg} \\ &= 10000\text{ J} \end{aligned}$$

Step 3: Total Heat Required

$$\begin{aligned} Q_{\text{total}} &= Q_1 + Q_2 \\ &= 48750\text{ J} + 10000\text{ J} \\ &= 58750\text{ J} \end{aligned}$$

Final Answer

58750

Later response (step 105)

To calculate the total heat required to change 2.5 kg of a solid from 40.0°C to a liquid, we need to consider two distinct processes:

1. Heating the solid from its initial temperature (40.0°C) to its melting point (90.0°C).
2. Melting the solid at its melting point (90.0°C) into a liquid.

Step 1: Heat Required to Raise the Temperature of the Solid

The formula for the heat required to change the temperature of a substance is:

$$Q_1 = m \cdot c \cdot \Delta T$$

Where:

- $m = 2.5\text{ kg}$ (mass of the solid),
- $c = 390\text{ J/kg}\cdot\text{K}$ (specific heat of the solid),
- $\Delta T = T_{\text{melt}} - T_{\text{initial}}$
 $= 90.0^{\circ}\text{C} - 40.0^{\circ}\text{C} = 50.0^{\circ}\text{C}$.

$$\begin{aligned} Q_1 &= 2.5\text{ kg} \times 390\text{ J/kg}\cdot\text{K} \times 50.0\text{ K} \\ &= 48,750\text{ J} \end{aligned}$$

Step 2: Heat Required to Melt the Solid

The formula for the heat required to melt a substance is:

$$Q_2 = m \cdot L_f$$

Where:

- $m = 2.5\text{ kg}$ (mass of the solid),
- $L_f = 4000\text{ J/kg}$ (heat of fusion).

$$\begin{aligned} Q_2 &= 2.5\text{ kg} \times 4000\text{ J/kg} \\ &= 10,000\text{ J} \end{aligned}$$

Step 3: Total Heat Required

$$\begin{aligned} Q_{\text{total}} &= Q_1 + Q_2 \\ &= 48,750\text{ J} + 10,000\text{ J} \\ &= \boxed{58750}\text{ J} \end{aligned}$$

Final Answer

58750 J

Recorded training signal	Step 30	Step 105
Judge score s_i	5	5
Judge probabilities $\pi_i^J(4) / \pi_i^J(5)$	0.438 / 0.562	0.119 / 0.879
Judge confidence c_i^J	0.572	0.771
Student uncertainty u_i^S	0.1728	0.0388

Table 7: Complete saved final responses (after thinking) and training signals for a selected heat-calculation case from an auxiliary fixed-teacher run. Both answers are correct; judge confidence increases and student uncertainty decreases between steps 30 and 105. Appendix D gives the calculations and case-selection details.

Step	Judge score 5		Judge score 4	
	High conf.	Low conf.	High conf.	Low conf.
0	135/314	103/334	116/137	215/351
50	156/319	165/332	109/196	204/359
100	231/358	158/322	140/259	146/310
150	335/463	107/311	151/249	135/314
200	361/482	115/290	235/373	95/218
250	438/586	124/295	216/338	98/200
300	463/638	117/294	212/344	106/214

Table 8: Response counts underlying Figures 2(b), 3, and 6. Each entry is k/n : n is the group size and k counts responses with student entropy below the pooled mean of completed score-4/5 responses in the same reported summary. High judge confidence means $c_i^J \geq 0.7$; low confidence means $c_i^J < 0.7$. The figures show reported steps 50–300; step 0 is included here for completeness.

27.4% at step 200 (Figure 3a). Within score 5, the high-confidence proportion continues to rise after the accuracy peak, from 59.8% at step 150 to 68.5% at step 300.

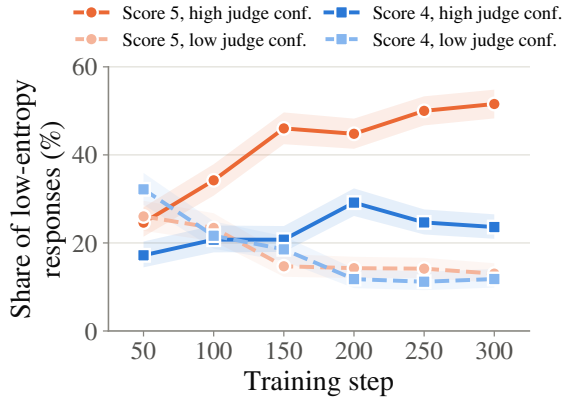


Figure 6: Judge-group composition among low-entropy responses during Qwen3-4B RUSD training with $\alpha = 0.5$. Only completed score-4/5 responses are included; low entropy means student entropy below their pooled mean at that reporting step. Each curve shows a group’s share of this subset; the four shares sum to 100%. High judge confidence means $c_i^J \geq 0.7$. Shading shows 95% Wilson confidence intervals.

Composition of low-entropy responses. Among responses below the pooled mean student entropy, the score-5/high-confidence group’s share rises from 24.6% at step 50 to 51.6% at step 300 (Figure 6). At step 300, this share is 463/898, whereas the within-group fraction in Figure 3(b) is $463/638 = 72.6\%$. The composition depends on both group size and the fraction of each group below the mean.

Spread of student uncertainty. The 25th–75th percentile range of response uncertainty shifts from 0.21032–0.31335 at reported step 50 to 0.04992–0.08053 at step 300 (Figure 2c). These quantiles describe variation across completed score-4/5 responses in each summary.

Detailed relative-certainty comparisons. At step 50, the fractions below the pooled mean student uncertainty are 48.9% for score-5/high-confidence responses and 52.6% overall; at step 100, they are 64.5% and 54.0%. The group’s fraction peaks at 74.9% at step 200 before decreasing to 72.6% at step 300, when the pooled fraction is 60.3%. It exceeds the pooled percentage from step 100 onward. The score-4/high-confidence group’s fraction remains at or above the pooled percentage throughout, with almost no difference at step 100. High- and low-confidence score-5 responses have similar fractions at step 50 (48.9% versus 49.7%), but differ at step 300 (72.6% versus 39.8%).