QuantumQA: Enhancing Scientific Reasoning via Physics-Consistent Dataset and Verification-Aware Reinforcement Learning

Songxin Qu[1,2], Tai-Ping Sun[3], Yun-Jie Wang[1], Huan-Yu Liu[2], Cheng Xue[2], Xiao-Fan Xu [3], Han Fang [5], Yang Yang[4], Yu-Chun Wu[3], Guo-Ping Guo[3], Zhao-Yun Chen[2]

[1]Institute of Advanced Technology, University of Science and Technology of China
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
[3]School of Physics, University of Science and Technology of China
[4]School of Electronics and Information Engineering, Anhui University
[5]School of Computing, National University of Singapore

Correspondence: gpguo@ustc.edu.cn, chenzhaoyun@iai.ustc.edu.cn

Abstract

Large language models (LLMs) show strong capabilities in general reasoning but typically lack reliability in scientific domains like quantum mechanics, which demand strict adherence to physical constraints. This limitation arises from the scarcity of verifiable training resources and the inadequacy of coarse feedback signals in standard alignment paradigms. To address the data challenge, we introduce QUANTUMQA, a large-scale dataset constructed via a task-adaptive strategy and a hybrid verification protocol that combines deterministic solvers with semantic auditing to guarantee scientific rigor. Building on this foundation, we propose the verification-aware reward model (VRM) tailored for Reinforcement Learning with Verifiable Rewards (RLVR), which employs an adaptive reward fusion (ARF) mechanism to dynamically integrate deterministic signals from a scientific execution suite (SES) with multi-dimensional semantic evaluations for precise supervision. Experimental results demonstrate that our method consistently outperforms baselines and general-purpose preference models. Notably, our optimized 8B model achieves performance competitive with proprietary models, validating that incorporating verifiable, rule-based feedback into the reinforcement learning loop offers a parameter-efficient alternative to pure scaling.

1 Introduction

Large language models (LLMs) have shown strong capabilities in general reasoning and mathematical problem solving (Wei et al., 2022; Shao et al., 2024b; Glazer et al., 2025). However, their reliability in scientific domains that require strict adherence to axioms and physical constraints remains inconsistent (Wang et al., 2024c; Taylor et al., 2022). This gap arises less from an absence of general reasoning ability than from limited access to domain-specific constraints and systematic verification that guide scientific reasoning beyond purely mathematical validity (Yao et al., 2023). Quantum mechanics, with a compact axiomatic foundation and well-defined constraint structure across common approximation regimes (Von Neumann, 2018; Nielsen and Chuang, 2010), therefore offers a stringent and controlled testbed for studying constrained scientific reasoning in LLMs (Guo et al., 2025b; Minami et al., 2025).

Applying LLMs to this rigorous domain faces two primary challenges. The first is the scarcity of high-quality, verifiable training resources. Existing resources typically bifurcate into two extremes: they are either small-scale, multiple-choice benchmarks inadequate for training complex reasoning (Minami et al., 2025), or large-scale synthetic corpora that lack physical verification mechanisms (Kashani, 2024). While recent datasets address quantum algorithm implementation and code generation (Yang et al., 2025a; Vishwakarma et al., 2024; Paltenghi and Pradel, 2024), and low-level quantum circuit design and compilation tasks (Foderà et al., 2024; Fu et al., 2025; Zhang et al., 2024), they primarily focus on algorithmic implementation and circuit-level tasks. The lack of scalable, process-supervised data constrains models' mathematical reasoning capabilities in quantum theory (Liu et al., 2024; Vishwakarma et al., 2024).

The second critical challenge lies in the limitations of existing alignment methods within scientific domains. Standard Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) suffers from reward over-optimization (Gao et al., 2023; Skalse et al., 2022; Taylor et al., 2022; Miao et al., 2025) and often yielding plausible-sounding hallucinations that violate physical constraints (Chua et al., 2025; Taylor et al., 2022). Although Process Reward Models (PRMs) provide dense supervision for formal domains like math (Setlur et al., 2025; Yang et al., 2025b) and

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 30821–30845
July 2-7, 2026

coding (Li et al., 2025; Zhang et al., 2025) by leveraging step-wise verification, such verifiers are often scarce in scientific settings. While RLVR incorporates execution signals (Gou et al., 2024b,a), it typically relies on sparse, outcome-based supervision (Lightman et al., 2023; Uesato et al., 2022), which is insufficient for complex scientific problems. Furthermore, existing multi-objective frameworks prioritize balancing general-purpose human preferences, such as helpfulness and safety (Wang et al., 2024a; Bai et al., 2022; Dai et al., 2024), rather than enforcing verifiable correctness or physical consistency.

To systematically address the challenges of data scarcity and reasoning hallucination, we build upon the Seed-Evolve paradigm (Wang et al., 2023b; Zeng et al., 2024) to expand topical breadth, while introducing two rigor-focused enhancements to ensure scientific validity. First, we implement a task-adaptive data construction strategy, which effectively mitigates hallucination by tailoring response structures to task complexity (Wang et al., 2023a). Specifically, this approach enforces conciseness for straightforward tasks while mandating detailed Chain-of-Thought (CoT) derivations for complex reasoning (Magister et al., 2023; Wei et al., 2022). Second, to guarantee scientific rigor in synthetic data, we devise a hybrid verification protocol that integrates deterministic verification tools via our scientific execution suite (SES), a deterministic toolkit consisting of automated verification scripts and symbolic solvers (Meurer et al., 2017), with semantic auditing (Zheng et al., 2023), culminating in a Human-in-the-Loop (HITL) review process to ensure the quality of dataset (Cobbe et al., 2021; Ouyang et al., 2022). By strictly enforcing physical constraints, this mechanism effectively purges plausible yet fallacious derivations that evade standard self-evaluation (Ji et al., 2023).

To address the challenge of feedback reliability, we introduce the verification-aware reward model (VRM), a unified reward model tailored for RLVR. VRM combines deterministic verification signals from the SES (Lightman et al., 2023) with semantic evaluation provided by an LLM-as-a-Judge framework (Zheng et al., 2023). Specifically, the semantic evaluation encompasses three key dimensions: Mathematical Correctness (*Corr*), Physical Consistency (*Phys*), and Instruction Following (*Inst*). Additionally, we employ an adaptive reward fusion (ARF) mechanism that dynamically modulates supervision strength based on verifiability (Uesato et al., 2022).

This dynamic calibration mitigates the reward sparsity often found in rigid execution environments, effectively bridging the gap between general preference optimization (Schulman et al., 2017) and the rigorous supervision required for complex scientific reasoning (Schick et al., 2023).

We conduct comprehensive evaluations on the test subset from QUANTUMQA and external scientific benchmarks (SUPERGPQA (Du et al., 2025) and PHYSICS) (Feng et al., 2025). The experimental results demonstrate that our method consistently outperforms supervised fine-tuning (SFT) (Zhang et al., 2023; Dong et al., 2024) across diverse open-source backbones. Notably, our VRM-optimized 8B model achieves performance competitive with proprietary systems like ChatGPT and DeepSeek-R1 (Guo et al., 2025a) in certain metrics. These findings suggest that incorporating verifiable, rule-based feedback into the reinforcement learning loop offers a parameter-efficient alternative to pure scaling. To elucidate the source of these gains, we perform fine-grained error analysis and component ablations. We observe that generic preference models struggle to distinguish subtle calculation errors from plausible hallucinations, whereas our verification-aware signals effectively reduce logical violations and mitigate the length-bias often associated with RLHF. Furthermore, ablation studies indicate that our ARF plays a crucial role in stabilizing optimization. These results suggest that fine-grained, multi-dimensional supervision significantly enhances the reliability of automated scientific reasoning.

In summary, our contributions are as follows:

1. We construct QUANTUMQA, a large-scale dataset of 92,749 samples for verifiable scientific reasoning. By leveraging a task-adaptive data construction pipeline and a hybrid verification protocol, we guarantee scientific rigor and high data validity. QUANTUMQA will be released upon publication.

2. We propose the VRM, which combines verification feedback and multidimensional semantic evaluation to enable precise optimization of scientific reasoning.

3. Experiments demonstrate that our method achieves superior performance and parameter efficiency, surpassing proprietary models like ChatGPT-5. Furthermore, it significantly outperforms SFT baselines across several base models as well as generic preference models.

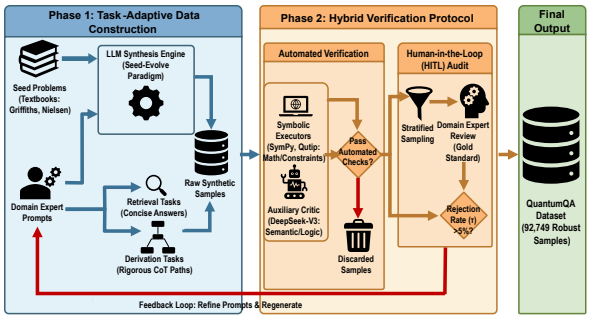

Figure 1: Dataset construction and verification pipeline.

2 The QuantumQA Dataset

To facilitate robust alignment and evaluation in physics-constrained reasoning, we introduce QUANTUMQA, a verified, multi-task dataset comprising 92,749 samples. Designed to overcome the limitations of previous works, QUANTUMQA supports both SFT and Reinforcement Learning pipelines, as well as serving as a rigorous evaluation benchmark.

2.1 Dataset Construction and Verification

A core challenge in synthesizing scientific data lies in mitigating reasoning hallucination. As shown in Fig. 1, we refine the standard synthetic data generation pipeline (Yu et al., 2024; Shao et al., 2024a) by incorporating two rigor-focused enhancements: a task-adaptive data construction strategy and a hybrid verification protocol. This approach ensures that the dataset aligns with strict physical constraints while maintaining the diversity inherent in large-scale synthesis. Comprehensive implementation details are provided in Appendix D.

Task-Adaptive Data Construction. We build upon the established Seed-Evolve paradigm (Wang et al., 2023b) to expand topical breadth from authoritative seed problems (Griffiths and Schroeter, 2018; Nielsen and Chuang, 2010). However, to address the distinct cognitive demands of different scientific tasks, we introduce a task-adaptive construction strategy. Specifically, retrieval-heavy tasks are constrained to concise answers to prevent verbose hallucinations (Zhou et al., 2023), whereas complex tasks are augmented with rigorous CoT paths to facilitate precise process supervision (Wei et al., 2022; Lewkowycz et al., 2022).

Hybrid Verification Protocol. To guarantee scientific rigor, we implement a dual-stage validation mechanism integrating automated verification with HITL auditing. Specifically, we employ SES to enforce physical constraints while leveraging an independent LLM as a semantic critic (Meurer et al., 2017; Johansson et al., 2012; Zheng et al., 2023). Following this, we conduct a HITL audit where a dynamic rejection threshold ($\tau > 5\%$) triggers iterative refinement to establish a high-reliability gold standard (Cobbe et al., 2021).

2.2 Dataset Analysis

We analyze the characteristics of QUANTUMQA to demonstrate its diversity and detail the data splits tailored for our optimization.

Diversity and Coverage. Instead of relying on simple pattern matching, QUANTUMQA encompasses five distinct types of tasks, including Short Answer, Fill-in-the-Blank, True/False, Multiple Choice, and Problem Solving, where the Problem Solving task is emphasized for the requirement of multi-step derivation. Furthermore, the dataset spans a broad spectrum of topics, from theoretical foundations to physical implementations, while maintaining a balanced difficulty distribution to assess varying cognitive loads. Details are provided in Appendix I.1.

Data Splits. We stratify QUANTUMQA into train ($\approx$95%), dev ($\approx$100), and a strictly held-out test set ($\approx$5%, $N = 4{,}675$). Within the training split, we allocate $\approx$70% for SFT training and construct the RLVR training set from the remaining challenging prompts, augmented with a small pool ($\approx$5%) of high-complexity prompts mined during SFT.

3 Verification-Aware Reward Model

In this section, we propose VRM, a verification-driven reward mechanism designed to align language model generation. As illustrated in Fig. 2, VRM integrates intrinsic semantic assessment with extrinsic tool verification. Concretely, the model operates through three coordinated stages. First, the SES performs code-based validation to yield deterministic feedback vector $\mathbf{v}$. Simultaneously, the Scoring head evaluates the response across multiple semantic dimensions to produce semantic scores $\mathbf{s}$. To synthesize these heterogeneous signals, we introduce the Dynamic Weight Allocation (DWA) head, a gating network that dynamically estimates reliability weights based on the execution results. Finally, these components are aggregated into a robust scalar reward, thus enabling the model to

effectively distinguish superior responses even in the presence of partial verification noise.

3.1 Verification Signals from SES

Given an input context x and a candidate response y, we evaluate the pair (x, y) along a set of dimensions $\mathcal{K}$. To formalize the verification signals, we partition the evaluation dimensions $\mathcal{K}$ into verifiable ($\mathcal{K}_{\text{ver}} = \{\text{Corr}, \text{Phys}\}$) and purely semantic ($\mathcal{K}_{\text{sem}} = \{\text{Inst}\}$) subsets. Specifically, dimensions in $\mathcal{K}_{\text{ver}}$ leverage the SES to yield deterministic execution feedback, whereas $\mathcal{K}_{\text{sem}}$ relies on semantic judgments due to the absence of formal solvers.

Accordingly, the feedback $\mathbf{v}$ composed of trinary verification indicators v_k for each $k \in \mathcal{K}$. The indicator $v_k \in \{1, 0, -1\}$ captures the detailed execution status:

$$v_k = \begin{cases} 1 & \text{constraint satisfied,} \\ -1 & \text{violation detected,} \\ 0 & \text{execution unavailable.} \end{cases}$$

Finally, the vector $\mathbf{v}$ is concatenated with the textual representation of (x, y) to serve as inputs for the VRM.

3.2 Verification-Aware Reward Architecture

Our architecture is founded on a pre-trained Transformer backbone $\mathcal{M}$ (instantiated via Qwen3-4B) that serves as a shared encoder, where the standard causal head is replaced by two specialized output modules. Given (x, y), the model first extracts a unified contextual representation:

$$\mathbf{h} = \mathcal{M}(x, y). \tag{1}$$

Based on this shared embedding $\mathbf{h}$, we design two parallel prediction heads, structured as 3-layer MLPs with 1,024 hidden units and GeLU activations, to jointly estimate generation quality and reliability weights.

Scoring Head. First, to assess the semantic plausibility of the generation, especially when SES is inapplicable, we employ a Scoring head to project the hidden state $\mathbf{h}$ into a semantic evaluation score:

$$\mathbf{s} = \sigma(\mathbf{W}_s \mathbf{h} + \mathbf{b}_s) \in [0, 1]^{|\mathcal{K}|}, \tag{2}$$

where $\mathbf{W}_s$ and $\mathbf{b}_s$ are learnable parameters. This score serves as an intrinsic quality estimation, providing a dense supervision signal regardless of tool usage.

DWA Head. Simultaneously, it dynamically estimates the reliability weight of each dimension. By concatenating the semantic embedding $\mathbf{h}$ with $\mathbf{v}$, this head computes a dimension-wise weight vector:

$$\mathbf{w} = \sigma(\mathbf{W}_g[\mathbf{h}\,;\,\mathbf{v}] + \mathbf{b}_g) \in [0, 1]^{|\mathcal{K}|}, \tag{3}$$

where $\mathbf{W}_g$ and $\mathbf{b}_g$ are learnable parameters. This head learns to adjust the corresponding weights with reliable SES feedback, facilitating robust reward estimation.

3.3 Adaptive Reward Fusion

Given the verification indicator $\mathbf{v}$, the semantic evaluation score $\mathbf{s}$, and the reliability weights $\mathbf{w}$, the VRM produces a scalar reward used by the downstream reinforcement learning algorithm.

Signal Fusion. For each dimension k, we compute a fused score $\tilde{s}_k$:

$$\tilde{s}_k = \lambda(v_k) + \big(1 - \lambda(v_k)\big) \cdot s_k, \tag{4}$$

where s_k denotes the k-th scalar component of $\mathbf{s}$. We set $\lambda(1) = 1.0$ to fully trust successful verification and $\lambda(0) = 0.0$ to fall back entirely on the semantic evaluation score when a reliable execution result is unavailable. For failed verification ($v_k = -1$), we use a small constant $0 < \lambda(-1) \ll 1$ to weaken the influence of SES, while preserving the contribution of the fine-grained semantic assessment s_k to ensure the stability of optimization.

Reliability-Weighted Aggregation. The final scalar reward is obtained as a reliability-weighted sum over fused scores:

$$r(x, y, \mathbf{v}) = \sum_{k \in \mathcal{K}} w_k(x, y, \mathbf{v}) \cdot \tilde{s}_k(x, y, \mathbf{v}). \tag{5}$$

As a result, optimization is dominated by fused scores and reliable weights.

3.4 Oracle-Guided Pretraining

To construct the VRM, we perform supervised distillation from an "Oracle-as-a-Judge" pipeline. Initially, we collect base tuples consisting of instructions, responses, and tool execution traces, denoted as $(x, y, \mathbf{v})$. To mitigate self-preference bias, we then employ an ensemble of oracles to evaluate these tuples, generating soft quality ratings $\mathbf{s}^*$ and dimension importance weights $\mathbf{w}^*$. By combining these elements, we construct the comprehensive

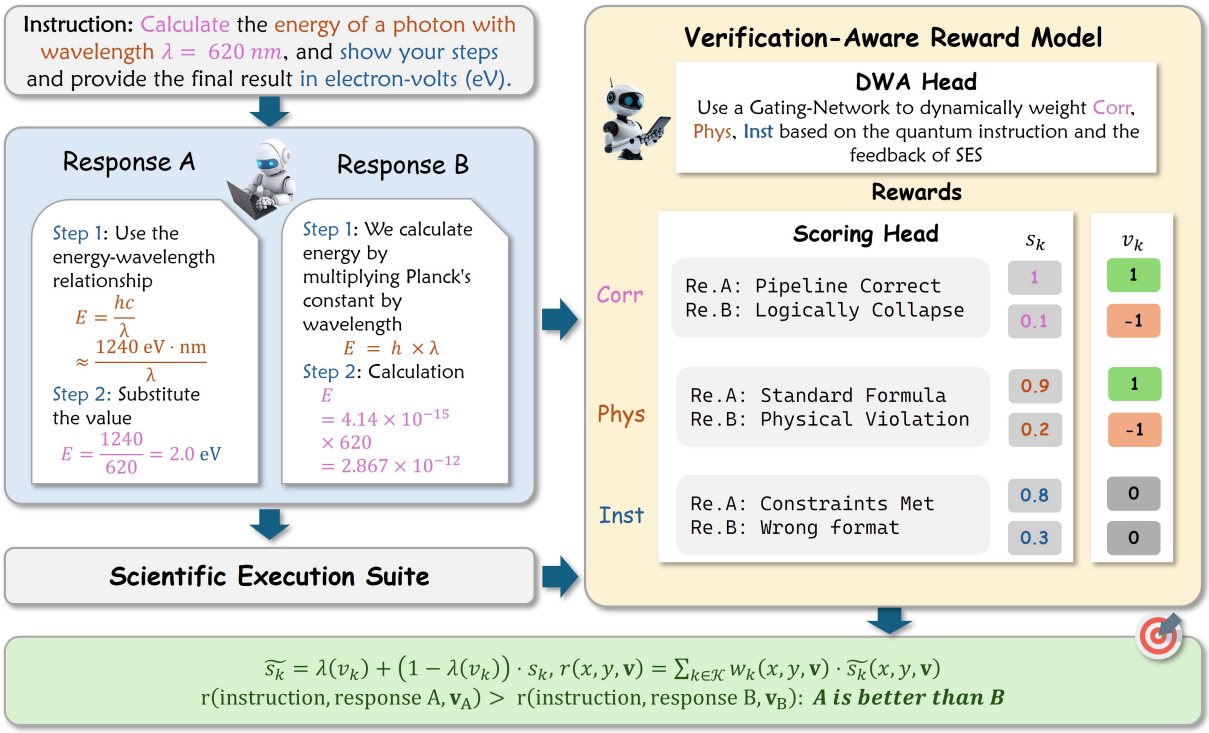

Figure 2: Overview of the verification-aware reward model.

dataset $\mathcal{D} = \{(x, y, \mathbf{v}, \mathbf{s}^*, \mathbf{w}^*)\}$. Crucially, our oracle models are strictly disjoint from downstream baselines and judges, eliminating information leakage and ensuring the VRM learns generalized verification over model-specific biases. The VRM is then trained to jointly optimize score regression and weight prediction via a multi-task objective:

$$\mathcal{L} = \mathbb{E}_{(x,y,\mathbf{v})\sim\mathcal{D}} \left[\|\mathbf{w} - \mathbf{w}^*\|_2^2 + \beta \|\mathbf{s} - \mathbf{s}^*\|_2^2 \right], \tag{6}$$

where β is a balancing coefficient. This alignment phase enables the VRM to serve as an efficient, standalone reward model for subsequent reinforcement learning, eliminating the need for expensive oracle queries.

4 Experiments

This section evaluates our work across multiple aspects. We detail the experimental setup in § 4.1 and present the main comparative results in § 4.2. To demonstrate the robustness of our approach, we then assess its generalizability to alternative reinforcement learning algorithms in § 4.3 and out-of-distribution scientific benchmarks in § 4.4. Furthermore, we provide a fine-grained analysis of model behaviors, specifically regarding verbosity (§ 4.5) and error distribution (§ 4.6). Finally, we conduct ablation studies in § 4.7 to isolate component contributions.

4.1 Experimental Setup

Backbones and Baselines. We validate our framework across a spectrum of open-source backbones, primarily focusing on the Qwen3 family (Team, 2025) and the LLaMA3 series (Grattafiori et al., 2024), as well as DeepSeek-R1-0528-Qwen3-8B. To thoroughly benchmark the effectiveness of our proposed method, we compare the following experimental settings: (1) **Baseline**: The unadapted open-source models; (2) **SFT**: Supervised fine-tuning on QUANTUMQA; (3) **Skywork RM**: Proximal Policy Optimization (PPO) (Schulman et al., 2017) training guided by Skywork-Reward (Liu et al., 2025), a state-of-the-art general-purpose reward model, serving as a baseline for scalar-reward RL; (4) **RLVR (Ours)**: PPO training driven by our VRM, serving as our primary optimization strategy; (5) **RLVR (GRPO)**: Group Relative Policy Optimization (GRPO) (Shao et al., 2024a) training guided by our VRM, evaluated to demonstrate the algorithm-agnostic generalizability of our approach. Additionally, we include frontier proprietary models (e.g., ChatGPT-5, Qwen3-Max) and human expert performance (evaluated on a representative stratified sample of 400

Models	Method	Fill-in-the-blank	True/False	Multiple Choice	Problem Solving	Short Answer
		Exact Match ↑	Accuracy ↑	Accuracy ↑	ACC_U ↑	ACC_U ↑
Qwen3-8B	Baseline	0.289	0.841	0.895	0.512	0.792
	SFT	0.500	0.899	0.967	0.621	0.817
	RLVR (ours)	**0.578**	**0.912**	0.978	**0.680**	**0.897**
	RLVR (GRPO)	0.568	0.904	**0.979**	0.655	0.875
Meta-Llama-3.1-8B-Instruct	Baseline	0.318	0.792	0.851	0.374	0.599
	SFT	0.494	0.876	0.959	0.517	0.799
	RLVR (ours)	**0.564**	**0.908**	**0.971**	**0.614**	**0.877**
DeepSeek-R1-0528-Qwen3-8B	Baseline	0.189	0.829	0.807	0.561	0.807
	SFT	0.442	0.881	0.948	0.579	0.821
	RLVR (ours)	**0.487**	**0.888**	**0.960**	**0.608**	**0.850**
Human-Expert	N/A	0.720	0.950	0.975	0.700	0.925
ChatGPT-5	Baseline	0.310	0.894	0.910	0.642	0.829
DeepSeek-R1-0528	Baseline	0.183	0.847	0.866	0.665	0.952
GLM-4.6	Baseline	0.361	0.839	0.947	0.708	0.952
Qwen3-Max	Baseline	0.491	0.903	0.957	0.804	0.977
Kimi-K2-Thinking	Baseline	0.323	0.815	0.862	0.689	0.966
Qwen3-235B-A22B-Instruct	Baseline	0.338	0.875	0.932	0.847	0.952

Table 1: Zero-shot performance comparison on QuantumQA using greedy decoding. Note: **Bold** font indicates the best performance within each model group. Underline denotes the overall best result in each column.

instances) as upper-bound references to contextualize the reasoning gap.

Benchmarks and Datasets. We conduct our primary evaluation on the held-out test split of QUANTUMQA. To further verify the robustness of our approach, we extend our evaluation to the quantum mechanics subsets of SUPERGPQA (Du et al., 2025) and PHYSICS (Feng et al., 2025). We employ these subsets as necessary alternatives to other domain-specific benchmarks, such as QUANTUMBENCH (Minami et al., 2025) and QUANTUMLLMINSTRUCT (Kashani, 2024), which are excluded from this study due to access restrictions and data incompleteness, respectively.

Evaluation Metrics. We adopt task-specific metrics tailored to the output format: *Accuracy* for Multiple-Choice/True-False tasks and *Exact Match* for Fill-in-the-blank tasks. For open-ended Problem Solving and Short Answer tasks, we utilize a unified score, ACC_U, derived via an LLM-as-a-judge protocol (Liu et al., 2023; Vertsel and Rumiantsau, 2024; Zheng et al., 2023; Li et al., 2024). Specifically, to mitigate the self-preference bias inherent in model-based evaluation (Zheng et al., 2023; Wang et al., 2024b), we employ Qwen3-Max to score responses on a normalized scale of $[0, 1]$ based on reasoning rigor and correctness. To ensure evaluation reliability, we validated this automated protocol against human expert annotations on a subset of the data, observing a high Spearman correlation. For other standard benchmarks, we follow their official evaluation protocols, normalizing results to $[0, 1]$ for consistency. Detailed validation results are provided in Appendix B.

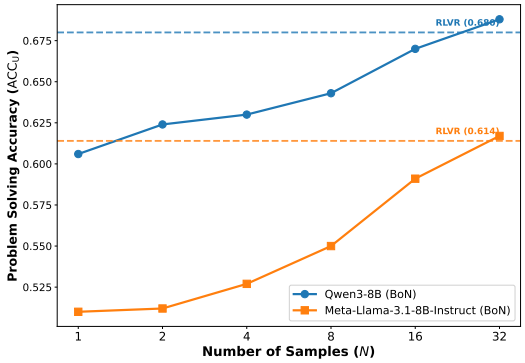

Figure 3: Best-of-N performance scaling.

4.2 Main Results

Table 1 presents the primary evaluation results. We conduct a comprehensive analysis to evaluate the overall efficacy and practical advantages of our proposed approach.

Effectiveness of Supervised Fine-Tuning. Comparing SFT against base models, we observe consistent performance gains across all metrics. This confirms that domain-specific supervision is essential for injecting quantum knowledge. However, the reliance on static ground-truth labels limits SFT's generalization in complex reasoning, motivating the integration of verification-based reinforcement learning.

Impact of Verification-Aware RL. Applying RLVR yields significant improvements over SFT baselines. For Qwen3-8B, our method improves Problem Solving accuracy from 0.621 to 0.680 and Short Answer accuracy from 0.817 to 0.897. Similar trends are observed with Meta-Llama-3.1-8B-Instruct. These gains indicate that our VRM effectively steers the model toward rigorous reasoning paths by providing fine-grained, step-by-step verification signals that SFT fails to capture. Notably, while RLVR delivers substantial gains on these base models, the improvements on heavily distilled reasoning models like DeepSeek-R1-0528-Qwen3-8B are more modest (e.g., 0.579 to 0.608 in Problem Solving). This phenomenon is intuitive: models like DeepSeek-R1-0528 are already extensively optimized through large-scale reasoning distillation and reinforcement learning, resulting in highly structured reasoning trajectories with limited margin for reward reshaping. Nevertheless, RLVR still provides consistent improvements and systematically reduces physical and logical violations even in this heavily distilled setting. This highlights that VRM-based supervision is highly effective for base models with weaker domain-aligned reasoning, while serving as a complementary and rigorous refinement step for already highly aligned models.

Parameter Efficiency & Proprietary Models. Our approach demonstrates that precise reward engineering can compensate for model scale. The VRM-optimized Qwen3-8B (0.680 in Problem Solving) outperforms significantly larger models, including ChatGPT-5 (0.642) and DeepSeek-R1-0528 (0.665). This suggests that open-weights models, when optimized with high-quality verifiable rewards, can bridge the gap with frontier proprietary systems.

Inference-Time Scaling and Distillation. We evaluate the discriminative power of our VRM using Best-of-N (BoN) sampling. As shown in Fig. 3, performance scales monotonically with sample size N, validating the VRM's effectiveness as a verifier. Notably, our RLVR policy (single sample) matches the performance of the SFT policy with BoN ($N = 32$). This confirms that RLVR effectively distills the verification signal into the policy parameters, achieving the accuracy of extensive inference-time search with significantly reduced computational cost.

Method	SuperGPQA	Physics
	Acc. ↑	Acc. ↑
Qwen3-8B		
Baseline	0.388	0.254
SFT	0.418	0.352
RLVR (Ours)	**0.466**	**0.423**

Table 2: Zero-shot performance on broad scientific benchmarks.

Comparison with Human Experts. Despite these improvements, a gap remains between the best model and human experts. Human experts maintain a clear advantage in precision-oriented tasks (Fill-in-the-blank, True/False), attributed to superior lexical exactness and robustness against conceptual nuances. Conversely, models outperform humans in complex generative tasks (Problem Solving), leveraging their vast computational capacity and knowledge synthesis. This indicates that while models excel at broad reasoning, they still lack the rigorous discrimination required for expert-level precision.

4.3 Generalization to Alternative RL Algorithms

To further demonstrate the algorithm-agnostic nature of our VRM, we evaluated its integration with GRPO. As shown in Table 1, integrating VRM with GRPO on Qwen3-8B yields consistent improvements over the standard SFT baseline across all five evaluation metrics. This demonstrates that our VRM is not restricted to PPO, but is also highly compatible with group-based policy optimization. Furthermore, while the GRPO variant slightly underperforms our fully-tuned PPO baseline, a gap we attribute to the restricted sampling budget, it exhibits a similar trajectory of improvement over SFT. Overall, these findings indicate that VRM is a robust and adaptable mechanism for enhancing verifiable reasoning across different reinforcement learning paradigms.

4.4 Generalization on Scientific Benchmarks

We assess out-of-distribution (OOD) generalization on standard scientific benchmarks (Table 2). Our RLVR method consistently surpasses the SFT baseline, boosting accuracy on the quantum subsets of both SUPERGPQA and PHYSICS. These gains indicate that our verification signals foster robust reasoning capabilities beyond the source domain, effectively mitigating overfitting risks.

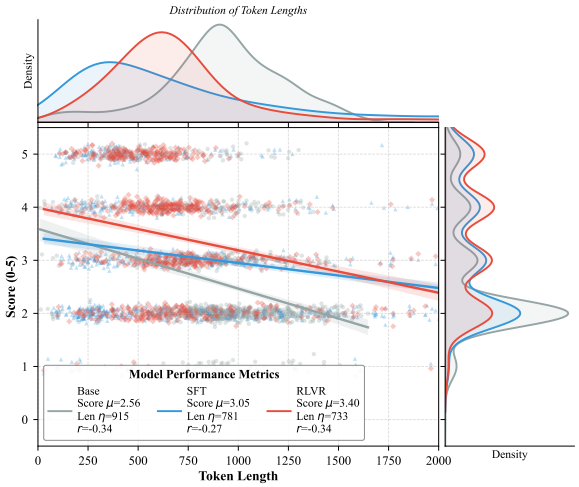

Figure 4: Joint distribution of token length and solution quality. Scatter plot with marginal density estimations for Base, SFT, and RLVR models on the Qwen3-8B backbone. Solid lines represent linear regression fits. The inset table reports the statistical metrics: μ denotes the mean value for *Score* (solution quality) and *Len* (η, token length), while r represents the Pearson correlation coefficient between length and quality.

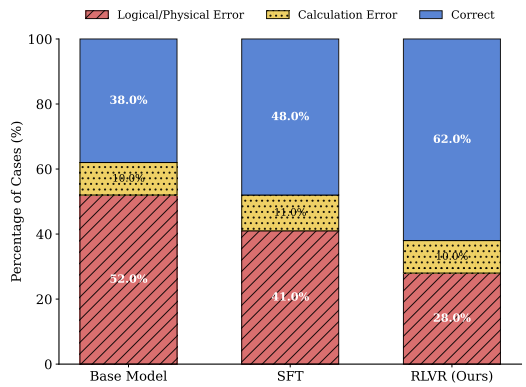

Figure 5: Distribution of error types across training stages.

4.5 Verbosity and Efficiency Trade-off

To rule out reward hacking, specifically verbosity bias (Singhal et al., 2024), we analyze the correlation between solution length and quality on Qwen3-8B (Fig. 4). Unlike Base and SFT models which exhibit redundancy, RLVR achieves superior accuracy with the lowest token usage. The observed negative correlation confirms that VRM mitigates reasoning loops: by integrating verification signals, the reward is incentivized to pursue efficient derivation paths rather than exploiting sequence length.

4.6 Fine-Grained Error Analysis

To investigate the source of performance gains, we conducted a blind human evaluation on 100 sampled Problem Solving tasks, classifying errors into logical/physical violations and calculation errors. As shown in Fig. 5, SFT exhibits a high rate of logical violations (41.0%) with 48.0% accuracy. In contrast, RLVR reduces logical errors to 28.0% and improves accuracy to 62.0%. This demonstrates that VRM acts as an effective regularizer, significantly enhancing physical consistency.

4.7 Ablation Study

To investigate the contribution of the verification signals and the adaptive weighting mechanism, we conduct a series of ablation experiments on Qwen3-8B. Results are summarized in Table 3.

Impact of Individual Verification Signals. We first isolate the contributions of distinct verification components. The consistent performance degradation observed in the ablated variants validates the architectural necessity of each component. Additionally, ablating domain-specific verifiers (Math and Physics) leads to significant performance degradation compared to removing semantic constraints (Instruction Following). specifically, removing the math verifier causes a sharper drop ($0.680 \rightarrow 0.636$) than ablating mathematical correctness (0.641), indicating that deterministic feedback from external tools provides stronger supervision than internal semantic consistency. Conversely, the marginal impact of instruction ablation suggests that performance on scientific tasks is predominantly driven by rigorous reasoning signals rather than format adherence.

Effect of reward aggregation strategies. We further evaluate the ARF mechanism by comparing it with two baselines: fixed uniform weights, and heuristic task-dependent weights manually assigned based on question type. Both static strategies underperform the ARF, particularly on Problem Solving and Short Answer questions. This suggests that static or manually designed weighting schemes fail to accommodate the varying reliability of verification signals in heterogeneous queries. In contrast, ARF effectively modulates rewards based on signal availability, attenuating noise from inapplicable verifiers while preserving informative gradients for robust and stable optimization.

Method	Fill-in-the-blank	True/False	Multiple Choice	Problem Solving	Short Answer
	Exact Match ↑	Accuracy ↑	Accuracy ↑	ACC_U ↑	ACC_U ↑
RLVR (Ours)	**0.578**	**0.912**	**0.978**	**0.680**	**0.897**
Ablation on Reward Signals					
w/o mathematical correctness	0.476	0.876	0.960	0.641	0.859
w/o physical consistency	0.560	0.908	0.971	0.646	0.872
w/o instruct following	0.566	0.909	0.971	0.654	0.892
w/o math verifier	0.549	0.908	0.968	0.636	0.877
w/o physics verifier	0.563	0.909	0.975	0.640	0.865
Ablation on Weighting Strategies					
Fixed Weight Strategy	0.562	0.907	0.971	0.645	0.877
Heuristic Weight Strategy	0.567	0.912	0.972	0.648	0.894
Alternative Reward Model					
Skywork RM	0.451	0.893	0.961	0.600	0.821

Table 3: Ablation study of reward components and weighting strategies on Qwen3-8B. The best results are highlighted in **bold**.

Comparison with Alternative Reward Models. To verify the effectiveness of our proposed reward model, we replace the VRM with a state-of-the-art general-purpose preference model (Skywork RM) (Liu et al., 2025). A substantial performance drop is observed across all task categories. This result highlights a key limitation of generic reward models in scientific domains: preference-based scoring lacks the granularity to detect subtle calculation errors or violations of physical consistency.

5 Conclusion

In this paper, we addressed the critical challenges of hallucination and the lack of verifiable feedback in applying LLMs to rigorous scientific domains. We construct and release QUANTUMQA, a large-scale dataset that serves as a reliable resource for training and evaluation. Building on this, we propose the VRM, a specialized feedback mechanism tailored for RLVR that dynamically integrates deterministic solvers with multidimensional semantic evaluation. This mechanism effectively mitigates the reward sparsity of rigid execution environments while strictly enforcing scientific rigor, thereby significantly enhancing the model's capability for reliable scientific reasoning.

6 Limitations

This work has several limitations that help clarify the scope of its contributions.

First, the current iteration of QUANTUMQA targets textual and symbolic derivations. While sufficient for verifying mathematical correctness and physical consistency, quantum mechanics often involves multimodal representations (e.g., circuit diagrams), which are not yet integrated into our synthesis pipeline.

Second, our current dataset predominantly captures formalized, post-hoc derivations, omitting the exploratory reasoning processes (e.g., trial-and-error, self-correction) central to recent Long Chain-of-Thought (Long-CoT) paradigms. However, this limitation motivates a highly promising direction for future research by leveraging our Problem Solving subset. Because these instances feature multi-step theoretical proofs (averaging over 5,000 characters) and are rigorously verified by our hybrid verification framework to prevent hallucinations, they serve as pristine prompt seeds. Building on this, our critical next step is synthetic augmentation. We will use these seeds to elicit extended reasoning traces from advanced LLMs, subsequently filtering the outputs against our ground truth to advance Long-CoT capabilities in complex scientific domains.

Finally, regarding cross-domain generalization, our current empirical scope is limited to quantum mechanics. We selected this field as a representative, high-complexity testbed due to its rigorous axiomatic foundation and the availability of deterministic verifiers. However, it is important to note that the VRM architecture and the SES interface are designed to be domain-agnostic. The core mechanism, integrating semantic reward signals with execution feedback, can be extended to other natural science domains (e.g., chemistry or classical mechanics) by replacing the symbolic solvers and rule checkers within the SES with corresponding domain-specific tools. Broadening our empirical

validation to a wider spectrum of scientific reasoning tasks to demonstrate the framework's generalizability remains an important direction for future work.

Ethical Considerations

Ethics and Data Policy. QUANTUMQA integrates model-generated content with publicly available materials, all utilized in compliance with their respective licenses, and will be released under a permissive open-source license. To ensure data privacy and quality, our pipeline employs automated filtering to remove personally identifiable information (PII) and incorporates an expert verification stage, detailed in Appendix H, to screen for scientific errors and potential biases. Furthermore, this verification process received Institutional Review Board (IRB) approval, and all participating experts provided informed consent. Participants were compensated at a fair rate of $20 USD/hour, which exceeds the local living wage. Finally, only anonymized data will be released for research purposes.

Intended use and risks. This work aims to improve the reliability of large language models in quantum mechanics using verifiable, rule-based reward signals. While the proposed multi-dimensional reward model enhances mathematical correctness and physical consistency, it may still produce imperfect evaluations due to limitations of verification tools or learned components. Users should avoid misuse such as reward hacking and over-reliance on automated judgments in high-stakes scientific settings, and are encouraged to perform independent verification when necessary.

AI assistance. We used large language models, including ChatGPT, to assist with language polishing and presentation refinement, while all technical design, experiments, and conclusions were determined by the authors.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862.

Jaymari Chua, Chen Wang, and Lina Yao. 2025. Learning natural language constraints for safe reinforcement learning of language agents. *arXiv preprint arXiv:2504.03185*.

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.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2024. Safe RLHF: Safe reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations*.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024. How abilities in large language models are affected by supervised fine-tuning data composition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 177–198, Bangkok, Thailand. Association for Computational Linguistics.

Xeron Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, Kaixin Deng, Shuyue Guo, Shian Jia, Sichao Jiang, Yiyan Liao, Rui Li, Qinrui Li, Sirun Li, and 76 others. 2025. SuperGPQA: Scaling LLM evaluation across 285 graduate disciplines. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Kaiyue Feng, Yilun Zhao, Yixin Liu, Tianyu Yang, Chen Zhao, John Sous, and Arman Cohan. 2025. Physics: Benchmarking foundation models on university-level physics problem solving. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11717–11743, Vienna, Austria. Association for Computational Linguistics.

Simone Foderà, Gloria Turati, Riccardo Nembrini, Maurizio Ferrari Dacrema, and Paolo Cremonesi. 2024. Reinforcement learning for variational quantum circuits design. *Preprint*, arXiv:2409.05475.

Zhenxiao Fu, Fan Chen, and Lei Jiang. 2025. Qagent: An llm-based multi-agent system for autonomous openqasm programming. *Preprint*, arXiv:2508.20134.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, Olli Järviniemi, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla,

Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, and 5 others. 2025. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *Preprint*, arXiv:2411.04872.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2024a. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. 2024b. ToRA: A tool-integrated reasoning agent for mathematical problem solving. In *The Twelfth International Conference on Learning Representations*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

David J Griffiths and Darrell F Schroeter. 2018. *Introduction to quantum mechanics*. Cambridge university press.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025a. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Xiaoyu Guo, Minggu Wang, and Jianjun Zhao. 2025b. Quanbench: Benchmarking quantum code generation with large language models. *Preprint*, arXiv:2510.16779.

Alexander Havrilla, Maksym Zhuravinskyi, Duy Phung, Aman Tiwari, Jonathan Tow, Stella Biderman, Quentin Anthony, and Louis Castricato. 2023. trlX: A framework for large scale reinforcement learning from human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8578–8595, Singapore. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Shima Imani, Seungwhan Moon, Lambert Mathias, Lu Zhang, and Babak Damavandi. 2026. TRACE: A framework for analyzing and enhancing stepwise reasoning in vision-language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3611–3625, Rabat, Morocco. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

J Robert Johansson, Paul D Nation, and Franco Nori. 2012. Qutip: An open-source python framework for the dynamics of open quantum systems. *Computer physics communications*, 183(8):1760–1772.

Shlomo Kashani. 2024. Quantumllminstruct: A 500k llm instruction-tuning dataset with problem-solution pairs for quantum computing. *Preprint*, arXiv:2412.20956.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. LLMs-as-judges: A comprehensive survey on LLM-based evaluation methods. *Preprint*, arXiv:2412.05579.

Qingyao Li, Xinyi Dai, Xiangyang Li, Weinan Zhang, Yasheng Wang, Ruiming Tang, and Yong Yu. 2025. CodePRM: Execution feedback-enhanced process reward model for code generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8169–8182, Vienna, Austria. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. 2025. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*.

Hongwei Liu, Zilong Zheng, Yuxuan Qiao, Haodong Duan, Zhiwei Fei, Fengzhe Zhou, Wenwei Zhang, Songyang Zhang, Dahua Lin, and Kai Chen. 2024. MathBench: Evaluating the theory and application proficiency of LLMs with a hierarchical mathematics benchmark. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6884–6915, Bangkok, Thailand. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. 2025. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In *The Thirteenth International Conference on Learning Representations*.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1773–1781, Toronto, Canada. Association for Computational Linguistics.

Aaron Meurer, Christopher P Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K Moore, Sartaj Singh, and 1 others. 2017. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103.

Yuchun Miao, Liang Ding, Sen Zhang, Rong Bao, Lefei Zhang, and Dacheng Tao. 2025. Information-theoretic reward modeling for stable rlhf: Detecting and mitigating reward hacking. *Preprint*, arXiv:2510.13694.

Shunya Minami, Tatsuya Ishigaki, Ikko Hamamura, Taku Mikuriya, Youmi Ma, Naoaki Okazaki, Hiroya Takamura, Yohichi Suzuki, and Tadashi Kadowaki. 2025. Quantumbench: A benchmark for quantum problem solving. *Preprint*, arXiv:2511.00092.

Michael A Nielsen and Isaac L Chuang. 2010. *Quantum computation and quantum information*. Cambridge university press.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. 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 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Matteo Paltenghi and Michael Pradel. 2024. A survey on testing and analysis of quantum software. *Preprint*, arXiv:2410.00650.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Preprint*, arXiv:1912.01703.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

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*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. 2025. Rewarding progress: Scaling automated process verifiers for LLM reasoning. In *The Thirteenth International Conference on Learning Representations*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024a. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

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

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2024. A long way to go: Investigating length correlations in RLHF. In *First Conference on Language Modeling*.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *Preprint*, arXiv:2211.09085.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.

Aliaksei Vertsel and Mikhail Rumiantsau. 2024. Hybrid LLM/rule-based approaches to business insights generation from structured data. *Preprint*, arXiv:2404.15604.

Sanjay Vishwakarma, Francis Harkins, Siddharth Golecha, Vishal Sharathchandra Bajpe, Nicolas Dupuis, Luca Buratti, David Kremer, Ismael Faro, Ruchir Puri, and Juan Cruz-Benito. 2024. Qiskit humaneval: An evaluation benchmark for quantum code generative models. In *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 01, pages 1169–1176.

John Von Neumann. 2018. *Mathematical foundations of quantum mechanics: New edition*. Princeton university press.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024a. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10582–10592, Miami, Florida, USA. Association for Computational Linguistics.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024b. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2024c. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. In *International Conference on Machine Learning*, pages 50622–50649. PMLR.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. 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)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024d. Helpsteer 2: Open-source dataset for training top-performing reward models. *Advances in Neural Information Processing Systems*, 37:1474–1501.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36:59008–59033.

Rui Yang, Ziruo Wang, Yuntian Gu, Yitao Liang, and Tongyang Li. 2025a. QCircuitbench: A large-scale dataset for benchmarking quantum algorithm design. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zhaohui Yang, Chenghua He, Xiaowen Shi, Shihong Deng, Linjing Li, Qiyue Yin, and Daxin Jiang. 2025b. Beyond the first error: Process reward models for reflective mathematical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 4711–4728, Suzhou, China. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*.

Weihao Zeng, Can Xu, Yingxiu Zhao, Jian-Guang Lou, and Weizhu Chen. 2024. Automatic instruction

evolving for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6998–7018, Miami, Florida, USA. Association for Computational Linguistics.

Ruiyi Zhang, Peijia Qin, Qi Cao, and Pengtao Xie. 2025. Dreamprm-code: Function-as-step process reward model with label correction for llm coding. *Preprint*, arXiv:2512.15000.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Guoyin Wang, and 1 others. 2023. Instruction tuning for large language models: A survey. *ACM Computing Surveys*.

Yuxuan Zhang, Roeland Wiersema, Juan Carrasquilla, Lukasz Cincio, and Yong Baek Kim. 2024. Scalable quantum dynamics compilation via quantum machine learning. *Preprint*, arXiv:2409.16346.

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. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: less is more for alignment. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

A Related Works

Scientific Question Answering Datasets Advancing scientific LLMs demands high-quality corpora for both effective instruction tuning and rigorous evaluation. While general benchmarks like MMLU (Hendrycks et al., 2021) and GPQA (Rein et al., 2024) establish fundamental evaluation baselines, they are predominantly limited to multiple-choice formats and lack the granular reasoning steps necessary for training models in specialized physics. Consequently, focus has shifted to quantum-specific resources, yet existing works exhibit a critical trade-off between scale and scientific rigor. On one hand, evaluation-centric datasets like QuantumBench (Minami et al., 2025) ensure validity through expert annotation but are insufficient for training due to limited scale. On the other, large-scale training corpora like QuantumLLMInstruct (Kashani, 2024) offer extensive coverage but lack rigorous physical verification mechanisms to prevent hallucination. Additionally, several recent datasets target quantum algorithm implementation and code generation (Yang et al., 2025a; Vishwakarma et al., 2024; Paltenghi and Pradel, 2024), as well as low-level circuit design and compilation tasks (Foderà et al., 2024; Fu et al., 2025; Zhang et al., 2024). However, these resources primarily emphasize programming and circuit-level objectives, and they still do not provide scalable, process-supervised data needed to systematically elicit mathematical reasoning in quantum theory (Liu et al., 2024; Vishwakarma et al., 2024). Our QUANTUMQA bridges this gap by providing a verifiable, large-scale dataset that supports both reliable training and rigorous evaluation under physics-consistent constraints.

Verifiable Reasoning and Alignment Recent advancements in mathematical reasoning have evolved from simple pre-training (Lewkowycz et al., 2022) to sophisticated reinforcement learning pipelines that leverage CoT and process supervision (Wei et al., 2022; Shao et al., 2024b; Luo et al., 2025). However, standard alignment techniques, primarily RLHF, often fail in rigorous scientific domains. They tend to optimize for subjective plausibility rather than objective correctness, suffering from reward over-optimization (Gao et al., 2023; Skalse et al., 2022; Miao et al., 2025) and frequently yielding plausible-sounding hallucinations that violate fundamental physical constraints (Ouyang et al., 2022; Taylor et al., 2022; Chua et al., 2025). To address these limitations, the field has shifted toward RLVR, which grounds reasoning in execution signals (Gou et al., 2024b,a). Despite this progress, existing RLVR frameworks typically rely on sparse, outcome-based supervision (Lightman et al., 2023; Uesato et al., 2022) manifesting as monolithic binary rewards (Wu et al., 2023). Such coarse feedback is inadequate for complex fields like quantum mechanics, which demands the satisfaction of orthogonal constraints. While PRMs succeed in math (Setlur et al., 2025; Yang et al., 2025b) and coding (Li et al., 2025; Zhang et al., 2025) via dense executable signals, they face challenges in science. Unlike formal domains, scientific problems often lack discrete intermediate ground truths, making it difficult to define the unambiguous, step-wise checks required

for reliable process supervision. While concurrent works like TRACE (Imani et al., 2026) share our goal of emphasizing stepwise reasoning analysis over outcome-only supervision, they focus on Vision-Language Models, leaving an unaddressed need for physics-grounded LLMs that utilize deterministic and semantic evaluation to validate scientific trajectories. Furthermore, existing multi-objective RL frameworks mainly prioritize balancing general-purpose human preferences, such as helpfulness and safety (Wang et al., 2024a; Bai et al., 2022; Dai et al., 2024), while often neglecting domain-specific constraints like physical consistency. Moreover, these methods typically rely on static scalarization (Wang et al., 2024d), lacking adaptive reward mechanisms to accommodate the varying availability of verification signals in scientific reasoning. Our VRM bridges this gap by dynamically fusing deterministic symbolic verification with probabilistic semantic evaluation, adapting supervision strength to the specific verifiability of each query.

B Validation of LLM-as-a-Judge Evaluation

To ensure the reliability of using an LLM-as-a-judge for our tasks, we conduct a comprehensive validation study. This study focuses on two key aspects: alignment with human expert consensus and inter-model robustness across different state-of-the-art LLMs.

Alignment with Human Experts We evaluate the consistency between our primary LLM judge and human experts on a stratified sample of 100 responses. Each response is graded by four experts on a 1–5 Likert scale, with the average score serving as the ground truth.

As shown in Table 4, the human inter-annotator agreement is substantial (ICC = 0.80), providing a reliable baseline. The LLM judge demonstrates robust alignment with human consensus, achieving an overall Spearman's ρ of 0.82. Notably, Problem Solving tasks exhibit stronger alignment ($\rho = 0.84$) compared to Short Answer tasks($\rho = 0.79$). We attribute this to the fact that mathematical and scientific derivations follow rigid logical rules, whereas Short Answer questions often involve semantic nuances that introduce greater subjectivity. Although a slight calibration offset exists (MAE = 0.55), the high rank correlation confirms the model's reliability for relative comparative evaluation, which is the primary focus of our experiments.

Task Type	**QWK**	**Spearman ρ**	**MAE**
Overall	0.78	0.82	0.55
Problem Solving	0.81	0.84	0.48
Short Answer	0.75	0.79	0.62

Table 4: Agreement between LLM judge and human experts. Human inter-annotator agreement (ICC) is 0.80. The model shows strong ranking capability (High ρ) despite absolute scoring deviation (MAE).

Robustness Across Different LLM Judges To ensure our evaluation metric is not biased toward specific model artifacts from Qwen3-Max, we further compare its judgments against other highly capable models, specifically Kimi-K2-Thinking and ChatGPT-5. All models were evaluated using identical prompts and parameters to ensure comparability.

Judge Model	**Spearman ρ** (vs. Human)	**Pearson r** (vs. Qwen3-Max)
Qwen3-Max (Ours)	0.82	-
Kimi-K2-Thinking	0.76	0.85
ChatGPT-5	0.83	0.87

Table 5: Consistency across different LLM judges. All evaluated models demonstrate high validity against human experts and strong inter-model agreement, indicating a shared capability in capturing output quality.

As presented in Table 5, all evaluated judges demonstrate high validity, exhibiting strong rank correlations with human experts ($\rho \geq 0.76$) and high inter-model agreement. These results suggest that despite architectural differences, these LLMs consistently capture similar underlying dimensions of output quality. Given these quantitative findings, we conclude that the specific judge used in our main experiments (Qwen3-Max) provides a fair, robust, and unbiased evaluation for this scientific reasoning task.

C Experimental Settings

We provide the comprehensive experimental settings for the our training pipeline: SFT, VRM training, and RLVR. All models are trained using bfloat16 precision to optimize memory efficiency without compromising numerical stability.

C.1 Hyperparameter Configuration and Search

Given the computational constraints of large language model training, we adopt established conventions from recent alignment literature (Ouyang et al., 2022; Rafailov et al., 2023) rather than performing an exhaustive grid search. We conduct a targeted search for the most sensitive hyperparameters, specifically the learning rate and batch size.

Selection Process. For the SFT and VRM stages, we sweep over learning rates $\alpha \in \{5 \times 10^{-6}, 1 \times 10^{-5}, 2 \times 10^{-5}\}$. We select $\alpha = 1 \times 10^{-5}$ for SFT and $\alpha = 5 \times 10^{-6}$ for VRM training, as these values achieve the lowest validation loss without overfitting.

For the RLVR stage, optimization stability is paramount. We find that standard learning rates often lead to policy collapse. Consequently, we search for a conservative learning rate in the range $\{1 \times 10^{-7}, 3 \times 10^{-7}, 5 \times 10^{-7}\}$. We identify that a lower learning rate of 3.0×10^{-7} provides the most stable improvement in reward signals while maintaining KL divergence within a reasonable range. The best-found hyperparameters are reported in Table 6.

C.2 Supervised Fine-Tuning

Before reinforcement learning, we perform SFT to initialize the policy model. This stage ensures the model acquires the foundational capabilities required for instruction following and format compliance. We train the model for 3 epochs with a global batch size of 128. We utilize a cosine learning rate scheduler with a peak learning rate of 1.0×10^{-5} and a warmup ratio of 0.1. This conservative schedule prevents catastrophic forgetting of pre-trained knowledge while adapting to scientific reasoning tasks.

C.3 VRM Training

To establish a robust reward signal, we train the VRM prior to the RL stage. Utilizing the Qwen3-4B backbone, we initialize the model weights and attach the randomly initialized Scoring and DWA heads. The model is trained on the Oracle-annotated dataset $\mathcal{D}$ (described in §3.4) to minimize the multi-task regression loss.

We employ a global batch size of 64 and train for 4 epochs. To ensure precise regression of the soft scores and reliability weights, we utilize a reduced learning rate of 5.0×10^{-6} with a linear decay schedule. The resulting model is then frozen and used as the critic during the subsequent RLVR phase.

C.4 RLVR with Verification-Aware Reward Model

In the final phase, we employ reinforcement learning to optimize the SFT-initialized policy model against the scalar reward provided by the VRM. While our primary experiments utilize PPO, we also evaluate our framework using GRPO to demonstrate its algorithm-agnostic nature.

Generation and Rollout. In this phase, we generate responses with a maximum length of 4096 tokens to accommodate complex reasoning chains. To encourage diverse exploration of the solution space, we employ nucleus sampling with $p = 0.85$ and top-k sampling with $k = 50$, setting the temperature to 0.6.

PPO Optimization. For our primary optimization phase using PPO, we set the learning rate to 3.0×10^{-7}, which is significantly lower than that of the SFT phase to ensure stable policy updates. The PPO algorithm runs for 4 epochs per batch with a buffer size of 1.

Generalization to GRPO. In this configuration, we maintain the learning rate at 3.0×10^{-7} and set the group size to 16. The effective batch size is configured based on a per-device count of 5 with 3 gradient accumulation steps.

C.5 Computational Budget

All training stages were conducted on a single compute node equipped with 8 NVIDIA H200 GPUs using full 8-way parallelism. We report the approximate wall-clock time and the corresponding total GPU hours for each phase as follows:

- SFT Phase: $\approx$ 5 hours (40 GPU hours) per model.
- VRM Phase: $\approx$ 10 hours (80 GPU hours) per model.
- RLVR Phase: $\approx$ 8 hours (64 GPU hours) per model.

C.6 Software and Implementation Details

Our implementation is built upon the PyTorch framework (Paszke et al., 2019) (v2.1.2) and the Hugging Face Transformers library (Wolf et al., 2020) (v4.37.0). For the reinforcement learning (PPO) phase, we utilize the TRL library (Havrilla et al., 2023) with standard hyperparameter configurations unless otherwise specified. Regarding evaluation, to ensure reproducibility and consistency with prior baselines, we use the official evaluation scripts provided by the respective benchmark creators without modification. For any text generation metrics (if applicable), we employ the Hugging Face evaluate library (v0.4.0) with default tokenizer settings.

Hyperparameter	**SFT**	**VRM**	**RLVR (PPO)**
Backbone	Qwen3-8B	Qwen3-4B	Qwen3-8B
Per-device Batch Size	8	4	5
Gradient Accumulation	2	2	2
Global Batch Size	128	64	80
Learning Rate	1.0×10^{-5}	5.0×10^{-6}	3.0×10^{-7}
LR Scheduler	Cosine	Linear	Cosine
Warmup Ratio	0.1	0.03	0.1
Num. Epochs	3	4	1
Specific Configs			
Loss Function	Cross-Ent.	MSE	PPO-Clip
Max Tokens	4096	2048	4096
Temperature	-	-	0.6
Top-k / Top-p	-	-	50 / 0.85
PPO Epochs	-	-	4

Table 6: Hyperparameters used for SFT, VRM training, and RLVR (PPO). The VRM serves as a frozen reward signal during the RLVR stage.

D Detailed Dataset Construction and Verification Pipeline

To ensure procedural transparency and facilitate reproducibility, we detail the specific implementation of our data synthesis pipeline. As illustrated in Fig. 1, the construction pipeline operates as an iterative, two-phase protocol designed to transform authoritative knowledge into high-quality instruction data while ensuring rigorous physical fidelity.

D.1 Phase 1: Task-Adaptive Data Construction

This phase employs a strictly ordered generation pipeline to derive instruction data from foundational texts.

Step 1: Seed Initialization (Source → Seed) We first digitize authoritative quantum mechanics and quantum information textbooks (Griffiths and Schroeter, 2018; Nielsen and Chuang, 2010) via Optical Character Recognition. Subsequently, we prompt DeepSeek-V3 to extract core theorems and principles from the processed text. To eliminate redundancy, we utilize a pre-trained sentence encoder (all-MiniLM-L6-v2) to evaluate semantic similarity, filtering out entries that exceed a cosine similarity threshold of $\Upsilon = 0.85$. This ensures a unique and high-quality seed repository.

Step 2: Hierarchical Concept Decomposition (Seed → Topic) To ensure granular conceptual coverage, we leverage Qwen3-Max, guided by specific system prompts, to systematically decompose each validated seed into a diverse array of fine-grained sub-topics. We again employ the semantic deduplication protocol on the generated outputs, rigorously filtering out redundancies to maintain a highly diverse topic pool.

Step 3: Data Generation (Topic → Question/Answer Pair) To maximize dataset diversity, we adopt a heterogeneous model ensemble generation strategy comprising DeepSeek-V3, Qwen3-Max, and ChatGPT-5. Tasks are bifurcated into deterministic and nondeterministic streams. Global dataset diversity is maintained via the aforementioned semantic similarity threshold ($\Upsilon = 0.85$).

- Deterministic Tasks: For closed-ended formats, we employ an end-to-end generation process. The ensemble generates the question stem (Q), the ground-truth answer (A), distractors (if applicable), and a difficulty label ($L \in \{\text{Undergraduate}, \text{Graduate}, \text{Research}\}$).
  - Multiple Choice: The model generates Q, the correct option (A_{correct}), and three plausible distractors ($A_{\text{distractors}}$) based on common quantum mechanics misconceptions to ensure high discriminative power. Outputs are constrained to option letters.
  - True/False: The model generate declarative statements requiring deep conceptual understanding to verify. Outputs are strictly boolean.
  - Fill-in-the-Blank: The model selects a key theorem or definition and masks critical variables, ensuring the provided context is sufficient for a unique, deterministic completion. The output contains only the missing term.

- Nondeterministic & Complex Tasks: For open-ended or calculation-heavy tasks, we adopt a two-phase stepwise generation pipeline.
    1. Question Generation (Topic $\rightarrow$ Q): The ensemble generates high-complexity queries, ranging from definitional synthesis to complex, multi-step derivations seeded with high-difficulty parameters.
    2. Answer Derivation & Profiling ($Q \rightarrow A+L$): The models produce a structured standard solution simulating a standard textbook answer key, alongside a difficulty label (L) that serves as a gating signal for subsequent reasoning injection.

Step 4: Adaptive CoT Injection (L + Q + $A \rightarrow$ Chain-of-Thought) We implement an adaptive mechanism to determine the necessity of explicit reasoning traces. We feed the triplet (Q, A, L) into our high-capability ensemble, instructing it to autonomously evaluate the cognitive load required to bridge Q and A.

If the model determines that the transition requires logical derivation or multi-step calculation (typically for Problem Solving tasks), it generates a rigorous, reverse-engineered derivation encapsulated within `<think>` tags. Conversely, for direct fact retrieval, tag generation is bypassed to prevent reasoning hallucination. Valid reasoning traces are parsed and explicitly formatted as the thought process in the final training dataset.

D.2 Phase 2: Hybrid Verification Protocol

To ensure the synthesized data meets the rigorous mathematical and physical standards required by the quantum science domain, we implement a dual-layer filtering mechanism.

Layer 1: Automated Verification Before human review, all raw synthetic samples undergo automated verification to filter out hallucinations and calculation errors.

- Symbolic Executors: We utilize the SES to computationally verify mathematical correctness and physical consistency.
- Auxiliary Critic: A strong, independent LLM critic evaluates logic and formatting. This critic is strictly decoupled from the Phase 1 synthesis ensemble to ensure independent scrutiny and prevent self-reinforcing hallucinations.

Layer 2: HITL Audit Samples that pass Layer 1 undergo a rigorous HITL audit. We implement a stratified sampling strategy to monitor batch-level performance. If the batch error rate exceeds a strict threshold ($\tau > 5\%$), the entire batch is rejected. Crucially, error patterns are systematically analyzed to refine the Phase 1 prompt templates and trigger targeted regeneration.

E Construction Pipeline of the VRM Training Dataset

The dataset $\mathcal{D} = \{(x, y, \mathbf{v}, \mathbf{s}^*, \mathbf{w}^*)\}$ is constructed through a rigorous 5-stage pipeline to ensure diversity, mitigate information leakage, and provide high-quality supervision signals.

E.1 Dual-Source Prompt Engineering

- **In-Domain Prompts:** We utilize the same topic distribution as QuantumQA but use a separate generator (GPT-4o) to synthesize entirely new questions. To guarantee zero leakage, we perform strict semantic de-duplication using Sentence-Transformers (all-MiniLM-L6-v2). Any generated prompt with a semantic similarity > 0.85 against the QuantumQA dataset and other existing prompts is automatically discarded.
- **General-Domain Prompts:** We interleave prompts from high-quality general reasoning datasets (MATH, COMMONSENSEQA, PIQA). These prompts also undergo the same BERT-based de-duplication process to ensure no accidental overlap.
- **Boundary & Constraint Perturbation:** For both in-domain and general-domain prompts, we apply a boundary mutation strategy. Specifically, we prompt a language model (GPT-4o) to generate new problem statements based on existing prompts while explicitly altering boundary conditions, physical constraints, or initial assumptions. This creates a set of adjacent problems that test the model's ability to handle subtle logical shifts. All mutated prompts are subjected to the same strict BERT-based semantic filtering.

E.2 Adversarial Response Generation

We employ a matrix of models spanning three orders of magnitude in parameter size. This includes lightweight models (Qwen2.5-1.5B/7B,

Meta-Llama-3.1-8B-Instruct) for simulating fundamental logic gaps, and frontier models (Qwen2.5-72B, Grok-3, ChatGPT-o1) for generating high-complexity reasoning traces.

For each question x, we adopt a dual-track prompting strategy. A standard prompt encourages correct, step-by-step reasoning to produce candidate positive responses y^+, while an adversarial prompt explicitly induces plausible but incorrect reasoning by imposing faulty constraints or assumptions, yielding hard negative samples y^-. Each model generates 3 responses per question from both tracks, resulting in a balanced mixture of correct solutions and challenging errors.

E.3 Deterministic Execution

All generated code traces are executed via our SES to obtain a hard verification label $\mathbf{v}$. This step provides a ground truth anchor for the subsequent phase.

E.4 Heterogeneous Oracle Annotation

We explicitly design oracle prompts to condition on each tuple $(x, y, \mathbf{v})$, requiring the Oracle Ensemble (Anthropic Claude 3.5 Sonnet and Google Gemini 2.5 Pro) to jointly produce semantic quality assessments and adaptive weights. Concretely, the oracles are instructed to:

- **Generate Soft Quality Scores** ($\mathbf{s}^*$) over multiple semantically grounded dimensions, including *Corr*, *Phys*, and *Inst*. These scores provide fine-grained supervision beyond outcome-level correctness, enabling the VRM to internalize structured evaluation signals.
- **Assign Dimension Weights** ($\mathbf{w}^*$) conditioned on verification outcomes $\mathbf{v}$. Through prompt constraints, the oracles are encouraged to place higher weights on dimensions that are explicitly verifiable, while down-weighting other dimensions.

This design allows the VRM to learn not only how to score responses along multiple semantic axes, but also how to adaptively reweight these axes based on task and verification outcomes.

E.5 Human-in-the-Loop Audit

Domain experts conduct batch sampling (5% of the data) to verify the balance of positive and negative samples and the consistency between SES execution results and Oracle scores.

F Implementation Details of SES

This section provides an extended technical description of SES, detailing its underlying verification logic and parser-executor pipeline.

F.1 Verification Logic and Script Composition

The SES is implemented as a specialized library comprising 12 modular, atomic verification scripts. Each script targets a specific class of constraints to detect outputs that may appear syntactically plausible but violate underlying rigorous laws. These verifiers are categorized into two primary groups:

- **Mathematical Consistency Checks:** These scripts validate symbolic equivalence and numerical correctness, ensuring the rigor of algebraic derivations and arithmetic precision.
- **Physical Consistency Checks:** These scripts enforce core quantum-mechanical axioms. Examples include verifying the unitarity of evolution operators and validating density matrices through positivity and trace constraints.

F.2 The Parser–Executor Pipeline

To reliably bridge the gap between natural language generations and deterministic code execution, the SES employs a robust, three-stage pipeline rather than relying on brittle heuristic string matching (e.g., regular expressions):

1. **Semantic Parsing:** An instruction-following Large Language Model (e.g., GPT-4o) acts as a semantic parser to extract targeted variables, such as scalars, matrices, or symbolic expressions, from the model's response.
2. **Type Casting and Parameter Passing:** The extracted elements are systematically converted into structured programmatic objects and passed as arguments to the corresponding verification script.
3. **Execution and Exception Handling:** The targeted script executes the verification logic and returns a deterministic boolean outcome. Crucially, if the semantic parser fails to extract valid arguments due to ambiguous formatting or incomplete reasoning, the pipeline explicitly flags the sample as unparsable. This strict exception handling ensures that positive reinforcement is exclusively reserved for well-formed, verifiable derivations.

G Granular Error Analysis of Verification Signals

To better understand the specific contributions and complementarity of the verification methods, we conducted a granular error analysis on 500 randomly sampled reasoning trajectories.

G.1 Complementarity of Deterministic and Semantic Verification

We first classify the sampled trajectories based on two orthogonal verification signals to analyze their overlap and divergences:

- **Semantic Verification:** Driven by the semantic scoring head, which predicts a continuous score $s_p \in [0, 1]$. We apply a strict threshold $\zeta = 0.8$ to categorize a trajectory as semantically valid.
- **Deterministic Verification:** A stringent binary check representing the aggregate outcome of the SES. A trajectory is deemed verified only if it produces an executable format and satisfies both mathematical correctness and physical consistency constraints.

As shown in Table 7, a significant portion of the samples (31.6%) failed the strict deterministic check, whether due to syntax parsing errors, minor mathematical calculation deviations, or strict violations of physical axioms, yet successfully passed the semantic verification. This demonstrates that the model frequently captures the correct semantic reasoning even when it struggles with precise programmatic implementation or exact constraint satisfaction. Our hybrid method successfully rewards these partially correct trajectories, preventing the RL agent from discarding valuable semantic insights while simultaneously mitigating the reward sparsity problem inherent in binary execution feedback. The semantic verifier evaluates the underlying reasoning process, whereas the deterministic verifier enforces rigorous final mathematical and physical validity.

	Passed Sem.	**Failed Sem.**	**Total**
Passed Det.	166 (33.2%)	25 (5.0%)	191 (38.2%)
Failed Det.	158 (31.6%)	151 (30.2%)	309 (61.8%)
Total	324 (64.8%)	176 (35.2%)	500 (100%)

Table 7: Confusion matrix of deterministic (Det.) vs. semantic (Sem.) verification signals.

G.2 Dimensional Analysis of Semantic Verification

To further investigate the specific bottlenecks within the reasoning process, we evaluated the semantic verification across the three aforementioned dimensions: *Corr*, *Phys*, and *Inst*. Table 8 presents the pass rates for each dimension within the same 500-sample subset.

Dimension	**Pass Rate**
Phys	68.0%
Corr	81.4%
Inst	92.6%

Table 8: Pass rates across individual semantic evaluation dimensions ($N = 500$).

The evaluation reveals that *Phys* is the most challenging dimension, exhibiting the lowest pass rate (68.0%). This highlights its critical role as the primary filter for plausible but fundamentally non-physical solutions. *Corr* functions as an intermediate logic checker, filtering out cases where physically valid terms are combined through erroneous computations. Conversely, *Inst* achieves a high pass rate (92.6%), indicating that surface-level formatting constraints are learned reliably. This allows the RL optimization to concentrate on the deeper physical and mathematical semantics. By aggregating these dimensions, the semantic verifier provides dense, structured feedback that effectively guides the model through the complex landscape of quantum physics problem solving.

H Expert Verification

To ensure the correctness and reliability of QUANTUMQA, we implemented a rigorous expert verification pipeline involving 15 domain experts, comprising 10 senior Ph.D. candidates and 5 postdoctoral researchers in theoretical physics and quantum information science. Distinct from generalist crowdsourcing, this cohort was recruited specifically for their proficiency in assessing quantum derivations and validating adherence to physical axioms.

Experts served as verifiers via a custom annotation interface integrated with SES, which facilitated the on-demand verification of intermediate computational steps. The annotation guidelines standardized four evaluation metrics used for both dataset curation and model output verification: (1) correctness regarding physical principles, (2) logi-

cal coherence of the reasoning process, (3) clarity of the problem formulation, and (4) readability of the solution. Additionally, experts were instructed to explicitly flag any potential bias or inappropriate content encountered during the review.

I Dataset Details

I.1 Additional Statistics

In this section, we provide detailed statistics regarding the task formats, domain coverage, and difficulty distribution of QUANTUMQA.

Task Heterogeneity. Figure 6 illustrates the distribution of the five task formats included in the dataset: Short Answer, Fill-in-the-Blank, True/False, Multiple Choice, and Problem Solving. Notably, Problem Solving tasks constitute the majority and exhibit significantly longer average answer lengths. This distribution reflects the dataset's design philosophy, prioritizing multi-step derivation and structured mathematical reasoning over rote memorization.

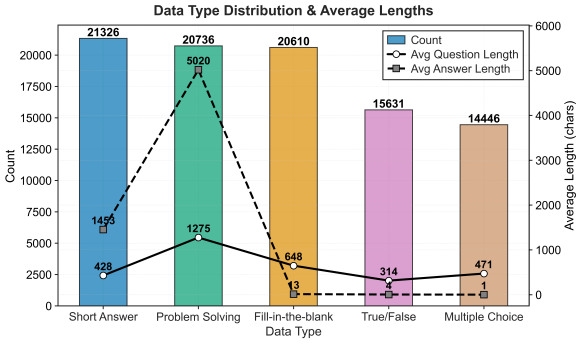

Figure 6: Distribution of question types in QUANTUMQA.

Domain, Difficulty, and Language. To ensure robust training and evaluation, we maintain balanced coverage across diverse subfields and difficulty levels. Fig. 7 and Fig. 8 depict the specific distributions for topic categories and reasoning difficulty. Regarding linguistic scope, the dataset consists exclusively of English text, covering standard scientific statement and mathematical reasoning patterns.

I.2 Dataset Examples

Table 9 provides comprehensive examples from the QUANTUMQA dataset.

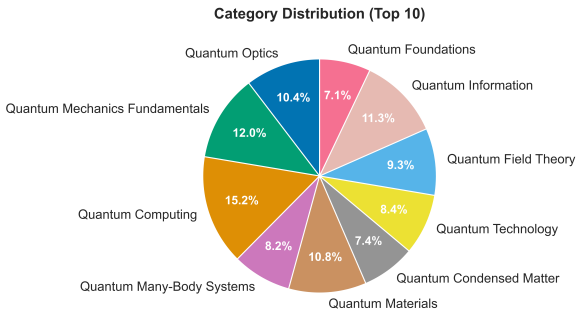

Figure 7: Top-10 category distribution of QUANTUMQA. The balanced coverage across diverse subfields ensures robust training and evaluation of quantum reasoning. Note that the cumulative percentage exceeds 100% as samples may be annotated with multiple topic labels.

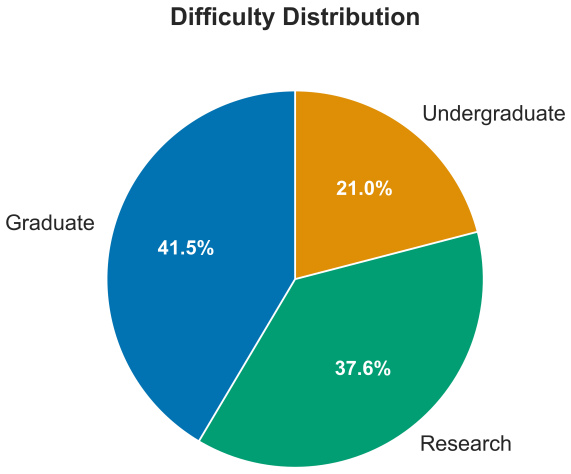

Figure 8: Distribution of difficulty levels in QUANTUMQA. The levels are categorized based on the number of reasoning steps and the complexity of physical concepts involved.

ID: 04215 **Type:** Problem Solving **Topic:** Quantum Optics **Difficulty:** Hard **SES Status:** Corr **Pass** | Phys Pass

[System Prompt]
You are an expert quantum physicist. Solve the following problem efficiently. Ensure strict adherence to mathematical logic and physical laws.
[Instruction]
Calculate the time evolution of the annihilation operator in the Heisenberg picture for a single-mode cavity field and verify the commutation relation consistency.
[Question]
Consider a Hamiltonian $H = \hbar\omega a^\dagger a$. 1. Derive the expression for $a(t)$ using the Heisenberg equation of motion $\frac{d}{dt}A = \frac{i}{\hbar}[H, A]$. 2. Prove that $[a(t), a^\dagger(t)]$ is time-independent.
[Answer]
Step 1: Equation of Motion
Calculate the commutator: $[H, a] = \hbar\omega[a^\dagger a, a] = \hbar\omega(a^\dagger[a, a] + [a^\dagger, a]a) = \hbar\omega(0 - a) = -\hbar\omega a$.
The differential equation is: $\frac{d}{dt}a(t) = -i\omega a(t)$.
Step 2: Time Evolution
Solving the ODE gives: $a(t) = a(0)e^{-i\omega t}$. Taking the Hermitian adjoint: $a^\dagger(t) = a^\dagger(0)e^{i\omega t}$.
Step 3: Verification
$[a(t), a^\dagger(t)] = [a(0)e^{-i\omega t}, a^\dagger(0)e^{i\omega t}] = e^{-i\omega t}e^{i\omega t}[a(0), a^\dagger(0)] = 1 \cdot 1 = 1$.
The commutator equals 1 for all t, confirming consistency.

SES Verification Log:
`[Corr] SymPy.check(commutator_diff == 0) -> True`
`[Phys] Unitary Evolution Consistency -> Verified`

ID: 20591 **Type:** Problem Solving **Topic:** Bound States **Difficulty:** Easy **SES Status:** Corr **Pass** | Phys **Fail**

[System Prompt]
You are an expert quantum physicist. Solve the following problem efficiently. Ensure strict adherence to mathematical logic and physical laws.
[Instruction]
Calculate the energy eigenvalue for the specified quantum state in a 1D infinite square well.
[Question]
A particle of mass m is confined in a one-dimensional infinite potential well of width L. Using the standard energy formula, calculate the energy E_n for the state with quantum number $n = 0$.
[Answer]
The energy levels for a particle in a 1D infinite box are given by the formula: $E_n = \frac{n^2\pi^2\hbar^2}{2mL^2}$.
Substituting the given quantum number $n = 0$:
$E_0 = \frac{0^2 \cdot \pi^2\hbar^2}{2mL^2} = 0$.
Thus, the energy of the particle in this state is zero.

SES Verification Log:
`[Corr] SymPy.eval(subs(E_formula, n, 0)) == 0 -> True`
`[Phys] Zero-Point Energy Check -> Failed`
Reason: Violation of Uncertainty Principle. For bound states, $n \in \mathbb{Z}^+$, minimum energy $E_1 > 0$.

Table 9: Detailed data samples from QUANTUMQA. Example 1 shows a successful generation. Example 2 demonstrates the framework's capability to detect physical hallucinations, where the model performs correct mathematical substitution (*Corr* Pass) but violates fundamental physical constraints (*Phys* Fail).

I.3 Artifact Citations and Intended Use

We strictly adhere to the citation guidelines for the artifacts used in our experiments. We confirm that our use of existing artifacts is consistent with their intended use, and the intended use of our created dataset is compatible with the original access conditions. Details are provided below:

- We collected problem sets and definitions from standard textbooks (Nielsen and Chuang, 2010; Griffiths and Schroeter, 2018) in the domain of Quantum Mechanics. These materials are copyrighted by their respective publishers. Our usage aligns with their intended purpose of educational assessment and domain knowledge evaluation. We operate under fair use principles, restricting usage strictly to non-commercial research contexts.

- We constructed the dataset using open-source Large Language Models (LLMs) in strict adherence to their respective licensing agreements. The generation of synthetic data for model alignment and evaluation is well-aligned with the intended use cases and safety guidelines specified by the model providers. This approach follows established community practices for scaling high-quality scientific reasoning data.

- The dataset created in this work is a derivative of the sources above. To maintain compatibility with the access conditions of the original educational materials, our dataset is intended solely for research purposes. It should not be used for commercial applications or outside of research contexts. The dataset will be released under a CC-BY-NC-4.0 license to enforce this restriction.

J Prompt Templates

To facilitate reproducibility and transparency, we provide the specific prompt templates used in our framework. These include the instructions for the physics-consistent data synthesis, the signal generation for our VRM, and the scalar scoring criteria.

J.1 Prompts for Data Synthesis

Table 10 presents the instruction used in our hybrid verification protocol. This prompt serves as a comprehensive filter to ensure strict adherence to physical consistency.

J.2 Prompts for Verification-Aware Reward Model

Table 11 presents the prompt template employed to construct the VRM training data. This template integrates external verification signals (from SES) with the original query and answer to generate multidimensional soft scores and their corresponding adaptive weights, which are subsequently used to train the VRM.

J.3 Prompts for Scalar Evaluation

For comparative evaluation, we employ a standard LLM-as-a-Judge prompting strategy to generate scalar scores (1-5). The template used for this assessment is shown in Table 12.

Instruction for Physical Consistency Verification

You are an expert physicist and a rigorous logic checker.
Your task is to verify the physical consistency of the following quantum mechanics problem and its solution.

Input:
Problem: {Question}
Solution: {Solution}

Verification Criteria:
1. Verify that the derivation respects fundamental principles (e.g., Uncertainty Principle, Commutation Relations).
2. Check for dimensional homogeneity in all equations.
3. Rigorously check the derivation steps, including integrals, matrix operations, and complex number arithmetic.
4. Ensure each step logically follows from the previous one without gaps.

Output Format:
1. If the content is valid, output `\boxed{PASS}`.
2. If there are any violations, output `\boxed{FAIL}` followed by a specific explanation of the error.

Table 10: The unified verification prompt used in our data synthesis framework. It enforces a rigorous standard, ensuring high-quality training data.

Instruction for VRM Signal Integration and Scoring

System Instruction:
You are an expert AI evaluator assessing the reasoning process of a model. Your objective is to synthesize external verifier signals with the original query and response to generate fine-grained, multidimensional soft scores and their corresponding adaptive confidence weights.

Context:
Original Query: {Query}
Model Response: {Response}
External Verifier Output: {Verifier_Signal} (e.g., SymPy result, Execution Status)

Task:
Evaluate the response across three specific dimensions based on the query and the verifier signal.
For each dimension, assign a soft score (0-1) and a confidence weight (0-1).

Dimensions:
1. *Mathematical Correctness (Corr)*: Is the calculation formally correct? Use the verifier signal as ground truth.
2. *Physical Consistency (Phys)*: Does the reasoning follow quantum mechanics principles?
3. *Instruction Following (Inst)*: Did the model follow constraints (e.g., format, method)?

Output Format JSON:
{
"scores": { "Corr": float, "Phys": float, "Inst": float },
"weights": { "Corr": float, "Phys": float, "Inst": float },
"rationale": "Brief explanation..."
}

Table 11: The prompt template for the Verification-Aware Reward Model. It synthesizes verifier signals to produce fine-grained soft rewards and adaptive weights.

Instruction for Scalar Quality Scoring

Please rate the quality of the following answer to the quantum mechanics question on a scale from 1 to 5.

Question: {Question}
Reference Answer: {Reference}
Model Answer: {Prediction}

Scoring Rubric:
1: Completely incorrect or irrelevant.
2: Contains severe physical errors or hallucinations.
3: Partially correct but contains minor logical flaws or calculation errors.
4: Correct reasoning and result, but lacks clarity or detail.
5: Flawless derivation, physically consistent, and clearly explained.

Output Format:
Output the integer score wrapped in a box format. For example, if the score is 5, output `\boxed{5}`.

Table 12: The prompt used for generating scalar evaluation scores (1-5).