

# Extending QuantumBench: A Selective Symbolic Enhancement and Multi-Perspective Evaluation Framework for Solving Quantum Problems

Yanyao Luo

## Abstract

Large language models have demonstrated the potential to accelerate knowledge acquisition and reasoning analysis in scientific research, gradually finding applications in high-threshold fields such as physics. However, quantum science tasks typically involve formal mathematical derivations, precise numerical computations, and highly specialized symbolic representations simultaneously. This poses greater demands on large language models, which are primarily built around natural language modeling. Existing evaluation results also indicate that the reliability and capability boundaries of these models in this domain remain unclear. Addressing the structural challenges of “insufficient formal computational verifiability” and “multiple-choice assessments failing to capture genuine reasoning capabilities” in quantum science tasks, this paper proposes a multi-perspective extended evaluation framework for quantum reasoning that rigorously reproduces the QuantumBench benchmark. Specifically, we introduce a hybrid reasoning mechanism with selective symbolic augmentation. This approach restricts external symbolic computation to high-yield subdistributions while maintaining evaluation comparability, enabling systematic analysis of the applicability boundaries of tool augmentation in quantum tasks. We simultaneously construct an open-ended quantum reasoning task suite and propose a dual evaluation protocol combining regularized automatic assessment with LLM-as-a-Judge to characterize the decoupling relationship between reasoning process quality and correctness. Furthermore, we design a graduate-level multi-stage quantum reasoning benchmark to explore the upper limits of model capabilities in higher-order theoretical deduction scenarios. Experimental results demonstrate that single multiple-choice accuracy struggles to fully reflect model reliability in quantum reasoning. In contrast, the proposed extended evaluation framework systematically reveals systemic differences across varying reasoning paths, tool usage, and task difficulty levels, providing more diagnostic applications for evaluating and deploying large language models in quantum domains.

**Keywords:** Quantum Reasoning Evaluation; Tool-Augmented Large Language Models; Symbolic Computation; Open-Ended Assessment

## 1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in general knowledge question-answering and text generation tasks, and are increasingly being in-

tegrated into scientific research workflows, including scientific writing, code generation, experiment planning, and data analysis (Brown et al., 2020; Achiam et al., 2023; Gemini Team, 2023; Grattafiori et al., 2024; Touvron et al., 2023). However, a growing body of research indicates that LLMs exhibit systematic vulnerabilities when tasked with highly formalized mathematical derivations, verifiable numerical computations, and rigorous domain-specific notation systems. These vulnerabilities manifest as computational errors, skipped derivation steps, and superficially coherent but substantively incorrect hallucinatory statements. Such issues directly limit the models’ applicability in high-stakes scientific research scenarios and obscure the true boundaries of their reasoning capabilities (Wei et al., 2022; Wang et al., 2022; Ji et al., 2023; Manakul et al., 2023). This issue is particularly pronounced in scientific reasoning tasks: even when models generate structurally complete and linguistically coherent reasoning texts, they may deviate in critical areas such as equivalent transformations, approximate conditions, or numerical consistency, leading to conclusions that are “superficially coherent but incorrect” (Kojima et al., 2022; Cobbe et al., 2021; Hendrycks et al., 2021; Stephan et al., 2024).

Quantum science presents an extreme and representative testing ground for the aforementioned challenges. On one hand, quantum mechanics and its subfields rely on highly formalized mathematical structures—including linear algebra and operator methods—demanding strict consistency between symbolic operations and physical semantics. On the other hand, quantum concepts themselves exhibit pronounced counterintuitiveness, with problem-solving often depending on implicit physical conventions (such as units, constants, approximations, and boundary conditions). In this context, model-generated errors are often not explicit syntactic mistakes but rather lurk within equivalent simplifications, the misuse of approximation conditions, or broken derivation chains. This results in “linguistically plausible” reasoning texts not necessarily corresponding to physically or mathematically correct conclusions (Meurer et al., 2017; Yao et al., 2022; Pan et al., 2024). To systematically evaluate LLMs’ performance in solving quantum problems, Minami proposed QuantumBench. This benchmark provides a crucial reference for evaluating quantum LLMs through multiple-choice questions spanning multiple quantum subfields (Minami et al., 2025).

However, multiple-choice assessments inherently possess unavoidable methodological limitations. First, correct answer rates are inevitably influenced by random guessing and question design, making it difficult to map performance differences within moderate accuracy ranges to genuine capability disparities (Hendrycks et al., 2020; Wang et al., 2024). Second, this evaluation paradigm focuses solely on final selections, failing to capture the correctness of reasoning processes, the completeness of arguments, or the verifiability of intermediate steps. This aligns with recent discussions on “process visibility  $\neq$  correctness” and has spurred widespread interest in open-ended evaluations and stronger process-oriented metrics (Zheng et al., 2023; Liu et al., 2023; Gu et al., 2024; Ye et al., 2024). Third, the predominant use of undergraduate-level questions limits the assessment of models’ upper limits in higher-order theoretical reasoning scenarios (Rein et al., 2023; Zhu et al., 2025; Phan et al., 2025).

Given the aforementioned challenges, this paper does not aim to propose a novel quantum problem-solving algorithm. Instead, it systematically extends existing quantum evaluation paradigms from an evaluative methodology perspective. We conduct rigorous replications following the standard QuantumBench workflow and introduce three complementary mechanisms to expand the signal evaluation space across three dimensions: verifiable computation, reasoning process characterization, and difficulty extrapolation.

This approach aims to more accurately delineate the capability boundaries and reliability analysis of LLMs in quantum science tasks. The core objective of this paper is to address how to construct more diagnostic evaluation frameworks in highly formalized quantum reasoning scenarios, rather than relying solely on single-metric multiple-choice accuracy. This overarching approach aligns with the recent trend of “enabling models to achieve more robust reasoning through verifiable computation and interaction with external tools” (Schick et al., 2023; Patil et al., 2023; Qin et al., 2023; Gou et al., 2024).

The main contributions of this paper are as follows:

**C1:** We propose a selective symbolic augmentation hybrid framework for quantum reasoning evaluation. Unlike existing approaches that treat tool augmentation as a contentious module, we construct a deterministic gating strategy based on question type and subdomain annotations. Symbolic execution is enabled only in subdistributions where verifiable computation yields significant gains. This allows for a systematic analysis of the conditions for positive and negative transfer effects of tool augmentation in quantum tasks, revealing its structural impact on the stability of evaluation results.

**C2:** We construct an open-ended quantum reasoning task suite and propose a dual-signal evaluation protocol. Addressing the challenge of capturing reasoning quality in multiple-choice assessments, this paper builds an open-ended task suite covering diverse quantum reasoning behaviors. It introduces a dual-scoring mechanism combining rule-based automated evaluation with LLM-as-a-Judge to quantitatively analyze the decoupling between structured reasoning capability and content correctness.

**C3:** We introduce QuantumBench-Grad, a graduate-level quantum reasoning extrapolation mechanism. While preserving the reproducibility of the original evaluation interface, we incorporate multi-stage reasoning constraints and difficulty annotations to systematically examine models’ upper limits in higher-order quantum problems. This enables analysis of whether structured reasoning behaviors reliably translate into final correctness.

## 2 Related Work

In the systematic evaluation of scientific reasoning capabilities, a mature lineage of general benchmarks has emerged. Beyond GPQA—which focuses on graduate/doctoral-level multiple-choice questions in biology, physics, and chemistry authored by domain experts, with “Google-proof” as its core design goal (Rein et al., 2023), MMLU has also become a classic benchmark for measuring models’ general knowledge and reasoning breadth through its multidisciplinary, multi-professional multiple-choice format (Hendrycks et al., 2021). Furthermore, BIG-bench complements the ecosystem of general evaluation by covering a broader range of task types and capability dimensions (Srivastava et al., 2022), while BBH further enhances the discrimination of complex reasoning abilities through its “harder subset” approach (Suzgun et al., 2022). Complementing these, SciBench focuses on university-level mathematics, chemistry, and physics problems, emphasizing multi-step reasoning and computational solution processes to reveal models’ shortcomings in solving scientific problems at a finer granularity (Wang et al., 2023). Similarly, arithmetic reasoning datasets like GSM8K are frequently employed to examine model stability and error patterns in computable tasks (Cobbe et al., 2021). However, these general scientific benchmarks typically lack specialized coverage of quantum science’s unique symbolic systems (e.g., state vectors, operators, tensor products) and subfield distributions (e.g., quantum information, quantum computing, condensed matter, and

quantum optics). Consequently, their evaluation signals cannot be directly translated into reliable characterizations of “quantum domain capabilities.” To address this gap, QuantumBench targets quantum science by compiling approximately 800 multiple-choice questions across nine quantum-related domains, each offering eight possible answers. It further analyzes models’ sensitivity to variations in question formats, providing a publicly available benchmark for evaluating LLMs in quantum domains that more closely aligns with domain characteristics and task structures (Minami et al., 2025).

Meanwhile, tool augmentation and hybrid reasoning frameworks provide crucial methodological support for enhancing the “verifiability” of scientific/quantum reasoning. Beyond demonstrating that Toolformer enables language models to learn “when to invoke external tools and how to integrate tool outputs into generation” through self-supervised learning—yielding significant gains in arithmetic and retrieval capabilities (Schick et al., 2023), Chain-of-Thought (CoT) prompts are widely employed to enhance decomposability and interpretability in complex reasoning tasks (Wei et al., 2022). Furthermore, self-consistent decoding enables marginalization between multiple reasoning paths to improve robustness (Wang et al., 2022). ReAct interweaves “think-act” sequences within reasoning trajectories, enabling models to maintain interpretable intermediate reasoning while interacting with external information sources or environments, thereby boosting success rates and credibility across multiple task categories (Yao et al., 2022). Regarding “tool-assisted executable reasoning,” PAL reduces arithmetic and logical errors by having models generate programs executed by interpreters (Gao et al., 2022; Chen et al., 2022), while ToRA further integrates tool interactions into a systematic framework for mathematical problem-solving agents to enhance verifiability and stability (Gou et al., 2024). For quantum reasoning tasks, the key value of tool augmentation often lies not in “injecting additional knowledge,” but in providing reproducible, auditable external evidence for algebraic simplification, symbolic derivation, and numerical consistency. This transforms “apparently reasonable narratives” into “verifiable chains of reasoning” (Meurer et al., 2017). Concurrently, evaluations and data construction around real-world API/tool usage capabilities are rapidly advancing: For instance, Gorilla focuses on reducing tool invocation hallucinations and improving API call correctness (Patil et al., 2023), API-Bank offers more systematic evaluations of tool enhancement and task collections (Li et al., 2023), while ToolLLM advances model training and evaluation of tool usage capabilities through large-scale real-world API instruction data (Qin et al., 2023). Collectively, these efforts provide a more direct research context for this paper’s analysis of gating and failure modes in selective symbolic computation enhancement. In quantum software and programming contexts, code assistants and instruction datasets for quantum SDK/circuit construction have also emerged, such as Qiskit Code Assistant (Dupuis et al., 2024), QuantumLLMInstruct (Kashani et al., 2024), and QCoder for quantum code generation (Mikuriya et al., 2025).

Evaluating open-ended generation tasks faces structural challenges due to their open answer space, diverse expressions, and lack of single correct answers, making LLM-as-a-Judge a scalable approximation to human evaluation. The MT-Bench system examines positional bias, verbosity bias, and self-reinforcement bias in LLM judging, proposing mitigation strategies that demonstrate strong judging models can achieve high levels of alignment with human preferences (Zheng et al., 2023). G-Eval further enhances consistency with human evaluation through structured scoring forms and chain-of-reasoning paradigms, improving the operability of open-ended text quality assessment (Liu et al., 2023). Concurrently, a systematic review on LLM-as-a-Judge systems highlights oppor-

tunities and risks in scalable evaluation, providing a more systematic research framework and practical recommendations across dimensions including consistency, bias control, and reliability verification (Gu et al., 2024). Recent empirical studies have also begun directly quantifying systematic biases in Judges across factors like preferences, presentation formats, and length (Ye et al., 2024), noting that uncertainty and variance issues may be more pronounced in Judges for mathematical derivation and verifiable reasoning tasks (Stephan et al., 2024). Therefore, in designing open-ended reasoning evaluations for quantum domains, integrating “domain-specific symbols and derivational verifiability” with “scalable discriminative evaluation protocols” can maximize coverage while minimizing the subjectivity and uncertainty inherent in open-ended responses.

## 3 QuantumBench Benchmark and Reproduction Settings

### 3.1 Dataset and Annotation Structure

QuantumBench is a multiple-choice benchmark designed to evaluate quantum scientific reasoning capabilities. The dataset comprises 769 quantum-related 8-choice-1-answer questions spanning nine subfields, with each question annotated by question type (Conceptual Understanding, Algebraic Calculation, Numerical Calculation) (Minami et al., 2025). For reproducibility in this study, we directly utilized the data files `quantumbench.csv` and `category.csv` provided in the repository to ensure consistency with the original work in terms of data sources and annotation systems. To minimize engineering details’ impact on results and ensure verifiability, we strictly adhered to the original evaluation script’s option shuffling strategy: Under a fixed random seed, we shuffled the options for each question, ensuring the order could be stably reproduced by question number. This guarantees equivalent evaluation inputs across different runs.

To facilitate subsequent comparative analysis by question type and subdomain, we further statistically analyzed the data distribution based on the local `category.csv` and summarized the results as follows (statistics derived from our actual `category.csv` computation). The overall structure shows a strong bias toward computable tasks: Algebraic computation questions dominate (74.8%), followed by numerical computation (18.7%), while conceptual understanding questions constitute a smaller proportion (6.5%). The subdomain distribution reveals a pattern where “mechanics and optics dominate, with diverse contributions from other directions,” providing a necessary foundation for subsequent discussions on model capabilities across different quantum subdomains.

Table 1: QuantumBench (769 problems) Data Distribution

Dimension	Category	Quantity N	Percentage %
Question Type	Algebraic Calculation	575	74.8
Question Type	Numerical Calculation	144	18.7
Question Type	Conceptual Understanding	50	6.5
Subfield	Quantum Mechanics	212	27.6
Subfield	Optics	157	20.4
Subfield	Quantum Field Theory	107	13.9
Subfield	Quantum Chemistry	86	11.2
Subfield	Quantum Computation	62	8.1
Subfield	Photonics	57	7.4
Subfield	Mathematics	37	4.8
Subfield	String Theory	33	4.3
Subfield	Nuclear Physics	18	2.3

### 3.2 Models, Inference Environments, and Reproducibility Protocol

For model and inference service configuration, this study selected Qwen2.5-7B as the evaluated model and invoked it locally via Ollama using the `qwen2.5:7b` format. Technical details regarding the model are documented in its official technical report (Qwen Team, 2024). The inference service utilizes Ollama’s local HTTP interface, maintaining consistent invocation chains with the repository’s local inference scheme to minimize additional variables introduced by framework differences. Experiments were conducted on a Windows 11 platform with an RTX 4060 8GB GPU, executing the full 769-question evaluation and extended experiments using Python 3.12 (with uv management).

For prompt design, we followed the zero-shot template provided by the repository, requiring the model to output a single option letter in a parsed, fixed format (e.g., “The correct answer is (X).”) to minimize the impact of output format noise on scoring and statistics. To ensure reproducibility, this study explicitly controlled all random elements: For multiple-choice questions, options were shuffled with a fixed `seed=0`. Additionally, within the hybrid reasoning framework, if a random selection branch existed in the fallback strategy, `random.seed(seed + idx)` was used to bind the random state to the question ID. This ensures stable reproducibility of experimental trajectories under identical data and code conditions (see Section 4.2 for implementation details).

### 3.3 Evaluation Metrics

For QuantumBench’s multiple-choice questions (MCQs), this paper adopts accuracy as the primary evaluation metric—the proportion of model outputs matching the correct answers. QuantumBench organizes quantum domain questions into eight-choice multiple-choice formats, covering nine quantum subfields and three question types (algebraic computation, numerical computation, and conceptual understanding). Thus, Accuracy enables comparable statistics across different problem categories under a unified interface while maintaining consistency with the original benchmark’s evaluation criteria. Furthermore, to assess “derivation, explanation, and argumentation” capabilities—areas poorly

covered by MCQ formats—this paper introduces open-ended tasks in Innovation Point 2 alongside a dual-scoring mechanism yielding a 0–100 composite score. This score is obtained by weight-based fusion of formalized automated evaluation and LLM-as-a-Judge assessment (formal definitions provided in Section 4.3). Additionally, to characterize structured reasoning capabilities in higher-difficulty scenarios, Innovation Point 3 establishes the graduate-level benchmark QuantumBench-Grad. While maintaining multiple-choice accuracy statistics, it additionally reports the “reasoning stage coverage” indicating whether answers explicitly cover Part A/B/C/D. This distinguishes “conclusion correctness” and “reasoning structure integrity” as two analyzable dimensions.

## 4 Methods

### 4.1 Baseline Reproduction Protocol

To ensure strict alignment with QuantumBench’s original evaluation workflow, this paper reproduces its standard pipeline—8-way MCQ + zero-shot instruction prompting + regularized answer extraction (Minami et al., 2025)—establishing a unified baseline for direct comparison with the original work. QuantumBench’s construction emphasizes independently solvable questions, manually crafted high-quality distractors, and categorization into nine subfields and three question types to support grouped evaluation (Minami et al., 2025). Under this framework, our baseline reproduction comprises three interconnected implementation steps: During the question-and-option rearrangement phase, we merge one correct answer with seven distractors into eight options. We then shuffle the option order using a fixed random seed `seed=0` to minimize the model’s potential exploitation of positional bias (Zheng et al., 2023; Wang et al., 2024). For prompt design, we employ a zero-shot template input of “question + 8 options + output format constraint,” requiring the model to output option letters in the fixed format `The correct answer is (X)`. This reduces parsing uncertainty from free generation and maintains consistency with the repository scripts. For answer extraction and failure handling, regularized regular expressions extract the last valid option letter (A–H) from model outputs as predictions. If extraction fails, it is marked as parsing failure and triggers fallback logic. It is crucial to emphasize that the fallback strategy for parsing failures directly impacts statistical fairness and error structure. Therefore, Section 4.2.3 further introduces an unbiased fallback mechanism to prevent systematic bias caused by fixed fallback options.

### 4.2 Innovation Point 1: Selective Symbolic-LLM Hybrid Reasoning

Quantum science problems typically involve both conceptual judgments and algebraic/numerical computations, where expression simplification, unit consistency, and numerical precision often determine the correctness of the final option (Meurer et al., 2017; Gao et al., 2022; Gou et al., 2024). Even on models with strong general reasoning capabilities, symbolic ambiguities in algebraic simplification, errors in symbol/subscript handling, and subtle deviations from numerical approximations can still lead to errors where “local reasoning appears valid but ultimately selects the wrong answer.” External symbolic computation tools (e.g., SymPy) provide executable, reproducible, and auditable computation paths (Meurer et al., 2017), thereby theoretically enhancing computational reliability; However, indiscriminate tool activation for all problems introduces additional

failure modes (e.g., unstable code generation, expression parsing failures, timeouts, and exceptions) and may degrade overall performance due to increased link complexity. Given this tension, this paper proposes a selective symbol-enhanced hybrid reasoning framework. Its objective is not to maximize multiple-choice accuracy numerically but to serve as an evaluation and diagnostic tool for characterizing the effective boundaries of external symbolic computation across different quantum task outcomes. This approach aligns with recent research trends integrating language reasoning with externally verifiable computation, exemplified by ToRA enhancing verifiability and solution stability through tool interaction in mathematical reasoning tasks (Gou et al., 2024).

#### 4.2.1 Selective Gating: When to Enable SymPy

Let the problem be denoted as  $q$ , with its problem type label  $t(q)$  and subdomain label  $d(q)$ . Here,  $t(q) \in \{\text{Conceptual, Algebraic, Numerical}\}$  corresponds to Conceptual Understanding, Algebraic Calculation, and Numerical Calculation, respectively. The deterministic gating function for v4 is defined as:

$$I_{\text{sym}}(q) = \mathbf{1}[t(q) = \text{Numerical}] \vee \mathbf{1}[t(q) = \text{Algebraic} \wedge d(q) \in \mathcal{D}_+] \quad (1)$$

where  $\mathbf{1}[\cdot]$  denotes the indicator function, and

$$\mathcal{D}_+ = \{\text{Mathematics, Quantum Computation, Quantum Chemistry, Quantum Mechanics}\} \quad (2)$$

The intuitive meaning of this gating strategy is: Symbolic path is disabled for conceptual comprehension questions, as their primary error sources stem from physical concepts and semantic judgments, where symbolic execution offers little direct benefit; Symbolic paths are always enabled for numerical computation problems, as these questions heavily rely on numerical consistency and reproducibility, where tool verification is more effective at reducing low-level computational errors; Symbolic paths are only enabled for algebraic computation problems in  $\mathcal{D}_+$ , as this subset typically features more standardized expressions, stronger resolvability, and more significant benefits from simplification and matching. To ensure reproducibility and auditability, this implementation maintains exact consistency between  $\mathcal{D}_+$  and the `SYMPY_DOMAINS` in v4 scripts (Mathematics / Quantum Computation / Quantum Chemistry / Quantum Mechanics), enabling deterministic reproduction of gate decisions based on problem type and subdomain labels.

#### 4.2.2 Two-Stage Hybrid Reasoning with Unbiased Fallback

When  $I_{\text{sym}}(q) = 1$ , this paper employs a “two-stage” strategy to balance cost and benefit. The first stage is zero-shot language solving: it leverages baseline prompts to generate answers and parse option letters. For algebraic computation problems, if parsing succeeds, it returns directly to save additional overhead. The second stage is the symbol-enhanced path: triggered fixedly for numerical calculation questions, and for algebraic calculation questions only when the first stage fails. It obtains verifiable answers through the process of “generating executable SymPy code—restricted execution—matching execution results with options.” The symbolic execution backend uses SymPy (Meurer et al., 2017), controlling reproducibility and robustness within a constrained execution environment. This includes preloading common symbols and functions (e.g., `x,y,z,t,hbar,omega`), setting execution timeouts (e.g., 30s), capturing exceptions and logging, and performing numerical/string matching between the executed `result` and options to output the final letter

prediction. By restricting tool usage to “validation/fallback” rather than “full control,” this framework aims to minimize new failure modes introduced by the toolchain while concentrating its advantages on computational steps most amenable to verifiable gains.

In engineering reproducibility, answer parsing failures constitute a primary noise source affecting MCQ statistical stability. Simple strategies like “fixed fallback to A for parsing failures” introduce strong positional bias and contaminate statistical conclusions. To address this, v3/v4 introduces hierarchical robust extraction and unbiased fallback: First, regular expressions cover multiple common expressions (e.g., `The answer is (X)`, `The correct answer is X`, and variants with brackets/bold text). If still failing, candidate letters are searched within the trailing text window. If still unsuccessful, it randomly selects from A–H using a fixed random seed to avoid fixed bias and maintain cross-run reproducibility. This process transforms “parsing failures” from a source of systematic bias into controllable random noise, making comparisons between different methods closer to genuine capability differences.

### 4.3 Innovation Point 2: Open-ended Task Construction and Dual Evaluation Framework

QuantumBench’s 8-way MCQ format facilitates large-scale statistical analysis. However, quantum problems in real research and advanced teaching contexts often require complete derivations, conceptual explanations, and arguments—not merely outputting option letters. QuantumBench also notes that future practice-oriented assessments should incorporate open-ended descriptive questions and structured task decomposition (Minami et al., 2025). Therefore, this paper constructs an open-ended task set and proposes a dual evaluation framework: “automated assessment + LLM-as-a-Judge.” Automated assessment provides interpretable constraints at the structural and formal levels, while LLM-Judge approximates signals for content correctness and argument quality. It is important to note that recent research has systematically highlighted potential structural biases in LLM-as-a-Judge regarding verbosity preferences, presentation sensitivity, and consistency (Ye et al., 2024). Moreover, such biases may be amplified and manifest as heightened uncertainty in mathematical/derivation-based content (Stephan et al., 2024). Given this reality, this paper employs dual signal fusion rather than a single Judge to mitigate overfitting risks associated with relying on a single evaluation source. The interpretability boundaries of this setup are further discussed in the Limitations section.

#### 4.3.1 Task Set Construction: 5 Open-Ended Question Types, 165 Questions Total

Open-ended tasks are stored in JSON format within `data/open-ended_tasks.json`, retaining the `original_question_id` field for traceability and auditability. Task design spans multiple capability dimensions from “derivation generation” to “error diagnosis,” specifically including `free_derivation` (50 questions), `concept_explanation` (50 questions), `analytical_qa` (15 questions), `error_diagnosis` (30 questions), and `multi_step_reasoning` (20 questions), totaling 165 questions. This segmentation enables subsequent analyses to distinguish model variations across dimensions such as derivation accuracy, concept explanation quality, error localization capability, and multi-step organizational ability, rather than being confined to the discrete answer space of MCQs.

### 4.3.2 Scorers and Fusion: AutoEvaluator, LLMJudge, and Final Scores

The AutoEvaluator outputs scores ranging from 0 to 100, calculated as a weighted sum across five interpretable dimensions: keyword coverage, reasoning depth (based on step marking and conjunction statistics), response length adaptability, formula/symbol usage (e.g., LaTeX and Dirac notation patterns), and structural integrity (requirements coverage checks). The weighting is defined as:

$$S_{\text{auto}} = \sum_{k=1}^5 w_k s_k, \quad \mathbf{w} = \{0.20, 0.25, 0.15, 0.20, 0.20\} \quad (3)$$

The objective of this design is to provide stable, verifiable “structure and form” constraints for open-ended responses without relying on external reference answers, thereby reducing irreproducibility caused by subjective evaluation alone.

LLMJudge outputs structured JSON across five dimensions: physical accuracy, mathematical correctness, logical consistency, clarity of expression, and completeness. The average of these five scores serves as the Judge’s total score:

$$S_{\text{judge}} = \frac{1}{5} \sum_{i=1}^5 s_i \quad (4)$$

Considering reproducibility and cost control, this paper employs the same model (Qwen2.5-7B) throughout evaluations, using different system prompts to simultaneously serve as both the model under test and the judge. While this setup ensures engineering reproducibility, it may introduce self-evaluation bias and is susceptible to known bias patterns in LLM-as-a-Judge frameworks (Ye et al., 2024; Stephan et al., 2024). Therefore, this paper treats it as a reproducible baseline evaluation scheme, clearly defining its potential risks and improvement directions in Section 7, without equating it to human-annotated unbiased ground truth. Consequently, this paper performs weighted fusion between automated evaluation and Judge scores, defining the final score as:

$$S_{\text{final}} = 0.4 \times S_{\text{auto}} + 0.6 \times S_{\text{judge}} \quad (5)$$

The rationale for assigning a higher weight to  $S_{\text{judge}}$  lies in the difficulty of fully capturing the physical correctness and argument quality of open-ended solutions through purely rule-based metrics. Concurrently,  $S_{\text{auto}}$  provides a “lower bound constraint” on structure and formality, mitigating to some extent the risk of bias arising from judges’ preferences for verbosity and stylistic expression (Ye et al., 2024).

## 4.4 Innovation Point 3: Development and Evaluation of QuantumBench-Grad (Graduate-level Benchmark)

QuantumBench questions primarily originate from open courses and teaching materials. Statistical analysis reveals that overall difficulty aligns more closely with undergraduate standards, excluding the highest difficulty and specialization levels. However, graduate-level quantum problems typically demand stronger theoretical frameworks and multi-stage reasoning organization (from modeling to derivation to physical interpretation). A single MCQ response alone struggles to capture whether the reasoning process is complete. Therefore, this paper introduces QuantumBench-Grad: while preserving the

original evaluation interface (still an 8-way MCQ), it incorporates multi-stage structured reasoning requirements. This enables the evaluation to maintain reproducible statistics compatible with the original pipeline while additionally observing models’ behavioral characteristics in higher-order reasoning organization.

#### 4.4.1 Data Scale, Sources, and Annotation Files

QuantumBench-Grad comprises 71 questions (`data/grad_benchmark/quantumbench_grad.csv`), divided into two parts: 21 human-designed graduate-level multi-stage template questions (Question IDs 0–20) covering typical answer structures like “stepwise derivation–physical explanation–conclusion induction”; The second part comprises 50 questions (Question IDs 21–70) sampled and enhanced from the original QuantumBench. These questions uniformly add A–D multi-step reasoning prefixes to increase reasoning complexity while retaining 8 options and standard answers, ensuring direct reuse of the existing “prompt-letter analysis–accuracy measurement” workflow. To support analyzability across domain and difficulty dimensions, this paper provides `category_grad.csv` containing fields such as `Subdomain_lecture` (for domain statistics), `DifficultyLevel`, and `ReasoningSteps`. This design enables QuantumBench-Grad to maintain the same “groupable statistics” capability at the data level as the original benchmark, while allowing for more granular discussions on “difficulty stratification—accuracy variation—reasoning structure coverage” in the experimental sections.

#### 4.4.2 Graduate Evaluation Protocol: Accuracy + Reasoning Stage Coverage

Under this evaluation protocol, the model must respond to segments Part A–D and ultimately output `The answer is X`. Beyond accuracy, this paper estimates reasoning structure coverage by extracting stage markers (e.g., “Part A,” “Step 1”) from responses as a supplementary metric for reasoning completeness. This metric does not directly equate to correctness but reveals whether the model is consistently prompted to engage in structured reasoning behavior. It thus provides an evidence foundation for subsequent analysis of whether longer reasoning texts translate into higher accuracy rates.

## 5 Experiments and Results

### 5.1 Experimental Setup

This section systematically reports the experimental configurations and results for baseline reproduction and three extensions (selective mixed reasoning, open-ended evaluation, and graduate benchmark). To ensure comparability, unless otherwise specified, all experiments employ the same model and local inference environment, maintaining output formats compatible with the original QuantumBench CSV structure. This enables direct statistical alignment of results across different methods. The tested model is Qwen2.5-7B, deployed and invoked locally via Ollama. Model specifications follow the official technical report(Qwen Team, 2024). Hardware consists of an RTX 4060 8GB GPU, with software running on Windows 11 and Python 3.12 (using uv for version management). The evaluation dataset comprises three parts: First, the full set of 769 questions from QuantumBench-MCQ (primary evaluation target for Innovation Point 1); Second, 165 open-ended questions (this section reports 10 pilot runs and 30 extended experiments on

the `free_derivation` subset); Third, the full set of 71 questions from QuantumBench-Grad (Innovation Point 3). For metrics, accuracy is used for MCQ and Grad, while the open-ended tasks employ the fusion score  $S_{\text{final}} = 0.4 \times S_{\text{auto}} + 0.6 \times S_{\text{judge}}$  defined in Section 4.3.

It should be noted that minor data annotation noise exists: as described in Section 3.1, non-standard values appear in the `Subdomain_question` field for Question IDs 659 and 660 in `category.csv`. To prevent this noise from affecting the interpretability of domain comparisons, this paper still uses the full dataset of 769 questions for overall accuracy statistics. However, when reporting statistics by subdomain, it defaults to reporting only the 9 canonical subdomains (totaling 767 questions). The source of this statistical scope difference is explicitly annotated in the corresponding tables and text.

## 5.2 Baseline Reproduction Results

We first reproduced the zero-shot baseline on the full set of 769 QuantumBench-MCQ questions to establish a unified reference for all subsequent improvements. The baseline achieved an overall accuracy of 38.49% (296/769). To characterize differences under varying cognitive load and capability components, we further analyzed baseline performance by question type, as shown in Table 2: Algebraic computation questions constituted the largest proportion yet exhibited the lowest accuracy (36.70%), thus becoming the primary bottleneck constraining overall performance. Numerical computation questions demonstrated relatively higher accuracy (44.44%) but still showed considerable room for improvement. Concept comprehension questions did not significantly outperform computational questions (42.00%), indicating that the model also exhibits instability in mastering the quantum conceptual framework.

Table 2: Baseline Accuracy by Question Type (QuantumBench-MCQ)

Question Type	Number of questions	Baseline Acc.
Algebraic Calculation	575	36.70%
Numerical Calculation	144	44.44%
Conceptual Understanding	50	42.00%

## 5.3 Innovation Point 1: Symbolic-LLM Hybrid’s MCQ Results

This section compares full-scale evaluation results between the Baseline and Hybrid v1–v4 models, further analyzing why selective gating (v4) achieves more stable net gains while suppressing tool side effects. Overall results are shown in Table 3: v1 and v2 exhibit varying degrees of degradation, indicating that indiscriminate or weakly constrained symbolic augmentation does not necessarily yield benefits; v3 largely returns to baseline levels; v4 shows limited improvement in overall accuracy (+0.78 percentage points). This outcome not only demonstrates a new performance optimum but also serves as an evaluative diagnostic signal: after rigorously controlling tool side effects, symbolic enhancement can generate stable positive transfer within specific quantum distributions. Further domain-set analysis reveals that this controversy exhibits significant structural clustering rather than uniform distribution. This phenomenon precisely highlights the capability gaps that multi-choice accuracy struggles to reflect, validating the necessity of selective gating in evaluation analysis.

Table 3: Overall Accuracy of Baseline and Hybrid v1–v4 (769 Questions)

Method	Correct Count / Total Count	Acc.	Relative to Baseline %
Baseline (Zero-shot)	296/769	38.49%	+0.00
Hybrid v1	282/769	36.67%	-1.82
Hybrid v2	259/769	33.68%	-4.81
Hybrid v3	293/769	38.10%	-0.39
<b>Hybrid v4 (Selective)</b>	<b>302/769</b>	<b>39.27%</b>	<b>+0.78</b>

To test whether the paired differences between Baseline and v4 are statistically significant, we employed a two-tailed exact McNemar test(McNemar, 1947). Among the differences between Baseline and v4, the transition from correct to incorrect responses was  $b = 103$ , while the transition from incorrect to correct responses was  $c = 109$ , yielding  $p = 0.731$ . This result indicates that the overall improvement of +0.78 percentage points is insufficient to establish statistical significance. Instead, it serves as directional evidence supporting the hypothesis that the gating strategy mitigates adverse effects.

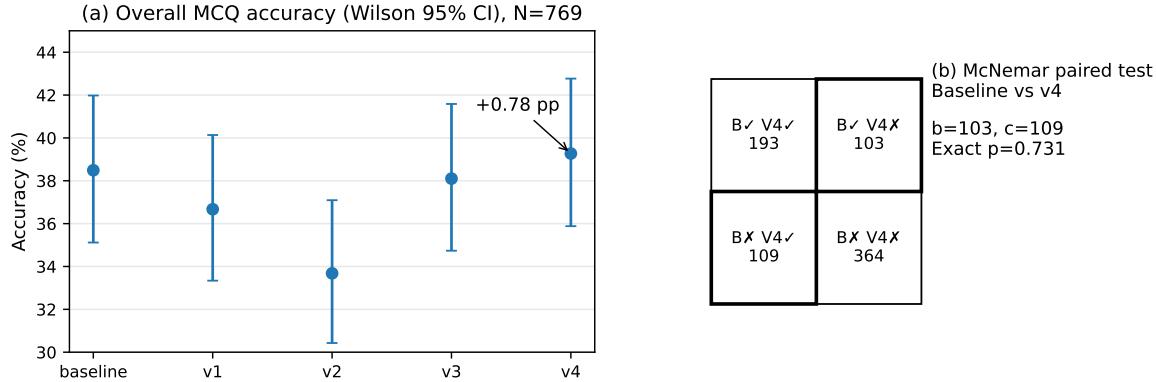


Figure 1: Overall Performance Comparison and Statistical Significance Analysis of Baseline vs. Hybrid v1–v4

To understand the sources of improvement and potential side effects, we further decomposed the results by question type and subdomain. Statistics by question type (Table 4) show that while v3 achieved significant gains on numerical computation questions (47.92%), it exhibited a pronounced negative transfer effect on conceptual questions (28.00%). This indicates that tool chaining may amplify parsing and matching failures on question types that do not require tools. By explicitly disabling the SymPy path for conceptual questions, v4 restored accuracy on these questions to baseline levels while maintaining a slight improvement on algebraic and numerical questions. This demonstrates the net effect of selective tool activation.

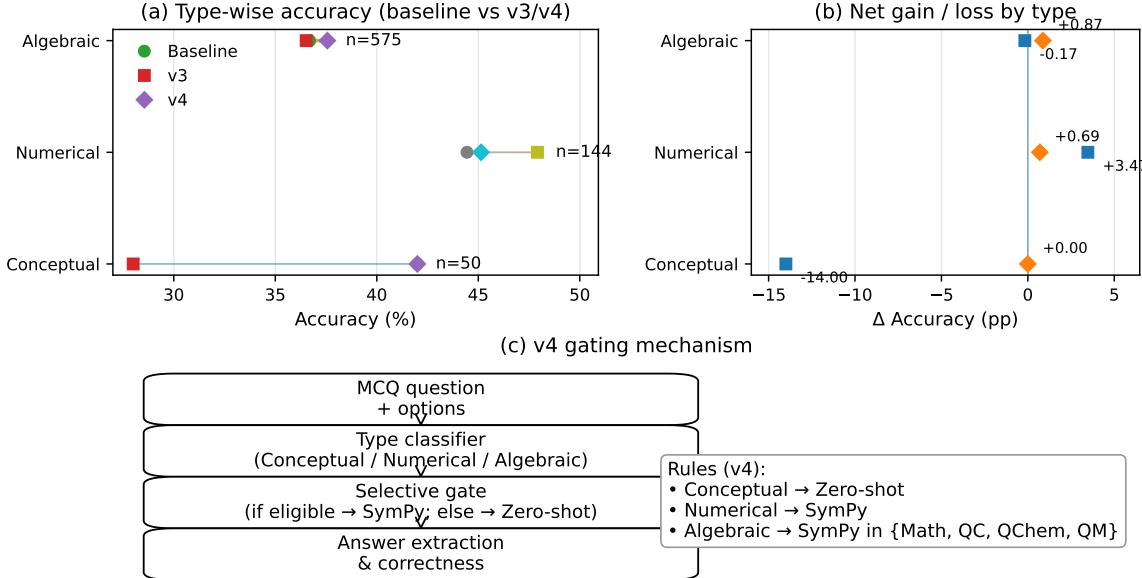


Figure 2: Performance Comparison by Question Type and Analysis of Gating Strategy Effectiveness

Table 4: Accuracy by Question Type (Baseline vs v1–v4)

Question Type	Baseline	v1	v2	v3	v4
Algebraic Calculation	36.70%	36.87%	32.35%	36.52%	<b>37.57%</b>
Conceptual Understanding	42.00%	36.00%	38.00%	28.00%	<b>42.00%</b>
Numerical Calculation	44.44%	36.11%	37.50%	<b>47.92%</b>	<b>45.14%</b>

Furthermore, under the statistical framework of 767 questions covering only 9 normative subfields (Table 5), v4’s gains exhibit pronounced domain-specific concentration, with the Quantum Computation subfield showing the largest improvement (+18.33 percentage points). A two-tailed exact McNemar test within this subfield yielded  $p = 0.0127$ . However, without multiple comparison correction, this result should be interpreted as an indicative statistical signal rather than a definitive conclusion. This phenomenon indicates that v4’s gated subset  $\mathcal{D}_+$  does capture a portion of high-yield problems characterized by “more formal expressions and verifiable computations.” It also suggests that in other subfields (e.g., Optics, Photonics), the symbolic path may still be constrained by the difficulty of expression parsing and option matching, leading to gains insufficient to offset the failure rate introduced by the tool.

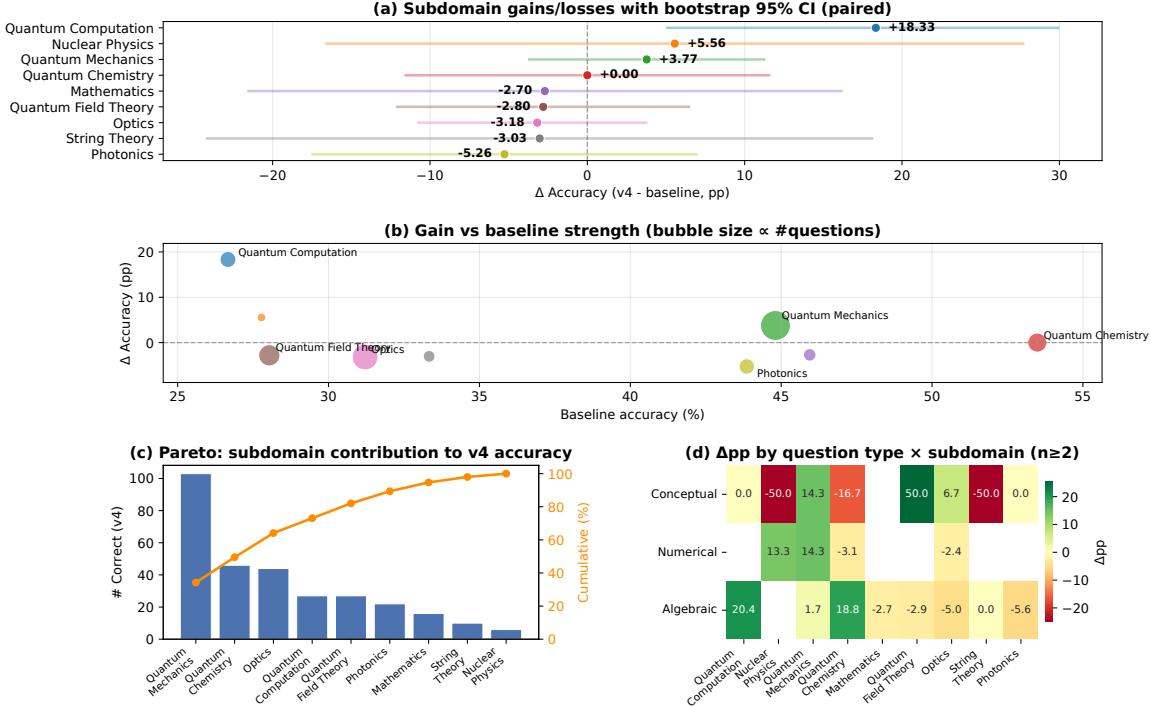


Figure 3: Analysis of Subdomain Gain Concentration and Symbol Enhancement Effect

Table 5: Subdomain Accuracy (Baseline vs v4; 9 Specification Subdomains, 767 Questions)

Subfield	Number of questions	Baseline	v4	Δ(pp)
Quantum Mechanics	212	44.81%	48.58%	+3.77
Optics	157	31.21%	28.03%	-3.18
Quantum Field Theory	107	28.04%	25.23%	-2.80
Quantum Chemistry	86	53.49%	53.49%	+0.00
<b>Quantum Computation</b>	<b>60</b>	<b>26.67%</b>	<b>45.00%</b>	<b>+18.33</b>
Photonics	57	43.86%	38.60%	-5.26
Mathematics	37	45.95%	43.24%	-2.70
String Theory	33	33.33%	30.30%	-3.03
Nuclear Physics	18	27.78%	33.33%	+5.56

To directly observe whether the gate uses tools on samples more suited to those tools, we recorded the routing results for each question in v4 ( $\text{Strategy} \in \{\text{sympy}, \text{zeroshot}\}$ ). As shown in Table 6, SymPy Hybrid covered 55.7% of the samples and achieved an accuracy of 46.26% on this covered subset, surpassing the 30.50% accuracy on the zero-shot subset. It is important to note that this comparison reflects the empirical performance of different strategy coverage subsets and does not constitute a strictly fair comparison under identical distribution conditions. However, it demonstrates that gating does empirically route samples more likely to benefit the tool path, resulting in a net gain overall. Concurrently, v4 incurs significant cost increases: Baseline averages approximately 936.6 tokens (Prompt + Completion), while v4 averages around 1492.5. On RTX 4060 8GB hardware, v4 completed 769 questions in approximately 4 hours, 34 minutes,

and 32 seconds, averaging 21.42 seconds per question. These results indicate that v4 functions more as an engineering strategy trading additional inference and execution overhead for computational consistency. Therefore, in resource-constrained or high-throughput scenarios, exploring finer-grained gating or lighter-weight verification mechanisms remains necessary to improve the cost-benefit ratio.

Table 6: v4 Routing Policy Distribution and Accuracy (769 Questions)

Strategy	Number of questions	Proportion	Acc.
SymPy Hybrid	428	55.7%	<b>46.26%</b>
Zero-shot	341	44.3%	30.50%

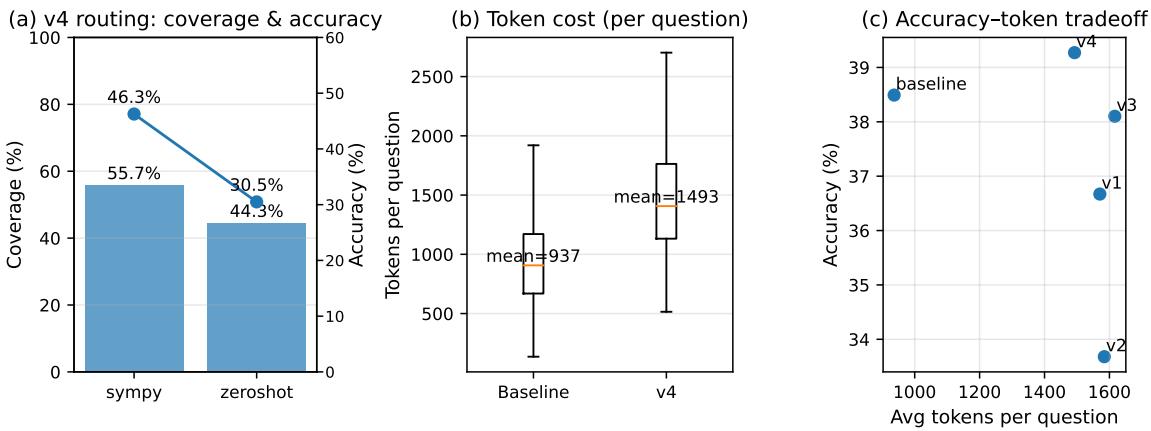


Figure 4: Distribution of v4 Routing Policies and Accuracy Rate

## 5.4 Innovation Point 2: Experimental Results on Open-Ended Reasoning Tasks

This section reports the performance of the open-ended evaluation framework on the `free_derivation` subset and analyzes the complementarity between AutoEvaluator and LLMJudge. The core challenge of open-ended evaluation lies in the open answer space and the absence of a single correct solution, necessitating a trade-off between scalability and reliability in evaluation signals. Recent research indicates that LLM-as-a-Judge approaches may exhibit systemic biases, particularly favoring verbose responses or being influenced by style and presentation (Ye et al., 2024). In mathematical derivation tasks, they also demonstrate higher uncertainty and greater variance (Stephan et al., 2024). Given this reality, this paper employs AutoEvaluator to provide stable structural/formal constraints and LLMJudge to deliver a primary signal focused on correctness. Weighted fusion is used to mitigate the risk of bias from a single evaluation source.

Table 7 summarizes results from the 10-question pilot and 30-question expanded experiments. The mean composite score  $S_{\text{final}}$  shows minimal variation from 10 to 30 questions ( $73.7 \rightarrow 71.7$ ), indicating framework stability within this subset. Simultaneously, LLMJudge exhibits significantly higher variance than AutoEvaluator, reflecting greater fluctuations in open-ended response quality and Judge’s heightened sensitivity to subtle correctness differences. This aligns with recent findings on Judge’s uncertainty behavior (Stephan et al., 2024).

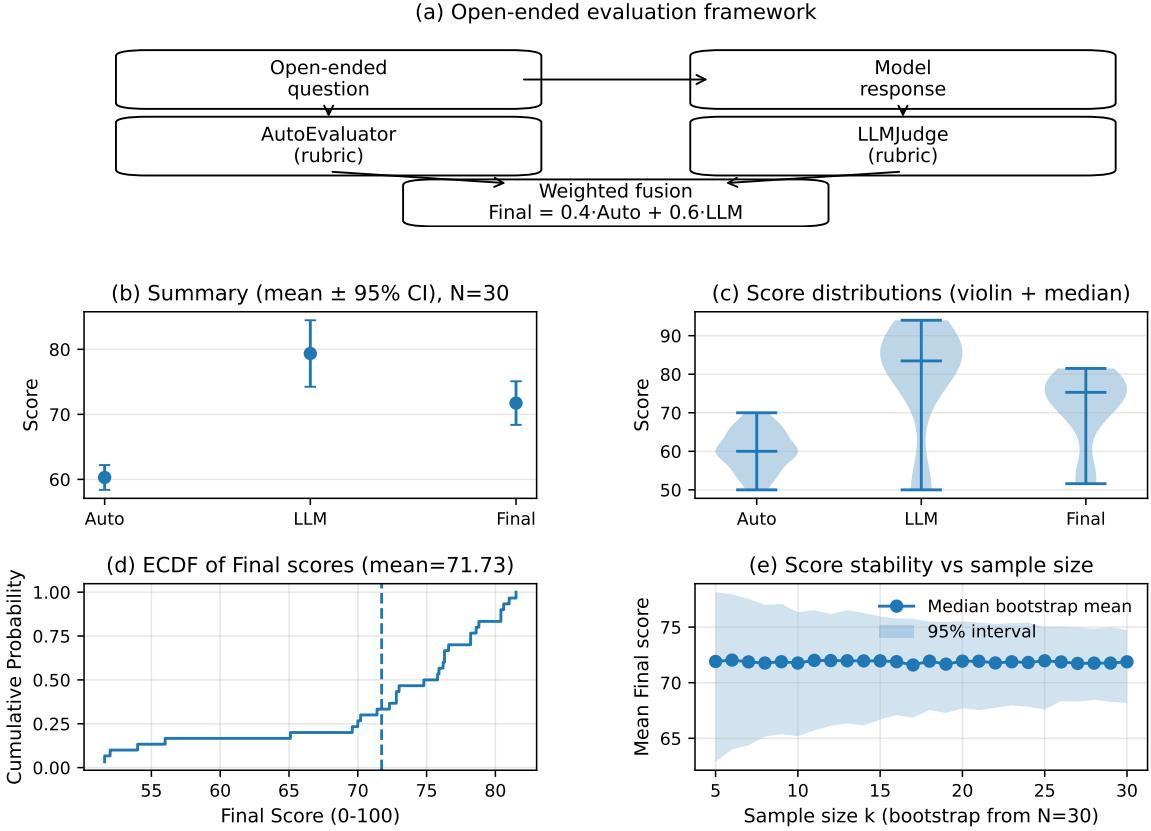


Figure 5: Open Evaluation Framework and Summary of Scoring Results

Table 7: Summary of Open-ended (Free\_Derivation) Evaluation

Scale	N	Auto (mean±std)	LLM (mean±std)	Final (mean±std)	Reasoning Steps (mean)	End-to-End Time
Trial Run	10	$58.6 \pm 6.4$	$83.7 \pm 14.3$	$73.7 \pm 8.6$	12.4	~7 min
<b>Scaled Run</b>	<b>30</b>	<b><math>60.3 \pm 5.4</math></b>	<b><math>79.4 \pm 14.5</math></b>	<b><math>71.7 \pm 9.5</math></b>	<b>12.5</b>	~21 min

In the 30-question extended experiment, this paper further calculates the mean scores of Final across domains (Table 8). Overall, the differences in mean scores between domains may stem from varying degrees of reliance on rigorous derivation versus contextual assumptions. Additionally, due to small sample sizes in certain domains (e.g., Mathematics with only 2 questions), this table is better suited for qualitative observation rather than drawing strong conclusions. To examine the complementarity of the dual evaluations, this paper calculates the correlation coefficient between Auto and LLMJudge, yielding  $r \approx 0.26$ . This low correlation indicates that the two indeed focus on different aspects: Auto emphasizes formal and structural consistency, while Judge prioritizes content accuracy and argument quality. Consequently, the fusion mechanism can mitigate the unilateral bias of either “focusing solely on structure” or “focusing solely on subjective correctness” to a certain extent.

Table 8: 30-Question Extended Experiment: FINAL Score by Domain

Field	Number of Questions	Final(Average)
String Theory	3	76.5
Quantum Computation	3	74.9
Quantum Mechanics	8	72.4
Quantum Field Theory	4	70.9
Optics	6	70.8
Photonics	4	70.5
Mathematics	2	64.0

## 5.5 Innovation Point 3: Experimental Results of QuantumBench-Grad (Graduate Benchmark)

This section reports evaluation results for the full set of 71 questions in QuantumBench-Grad, focusing on whether increased difficulty statistically manifests as a decline in overall accuracy, and whether multi-stage prompts can reliably trigger structured reasoning behavior and correlate with correctness. The overall accuracy rate is 35.21% (25/71), representing a 3.28 percentage point decrease compared to the original QuantumBench baseline (38.49%). This indicates that, while maintaining the 8-way MCQ interface unchanged, the multi-stage reasoning requirement indeed increases task difficulty and compresses the model’s accuracy space. Results stratified by difficulty are shown in Table 9. The Graduate-3 difficulty level contains only two samples, rendering it statistically insignificant; it is documented here solely as an observed phenomenon.

Table 9: QUANTUMBENCH-GRAD Accuracy by Difficulty Tier

Difficulty Level	Correct/Total	Acc.
Graduate-1	18/56	32.1%
Graduate-2	5/13	38.5%
Graduate-3	2/2	100.0%*

Note: Graduate-3 sample size is too small to be statistically significant.

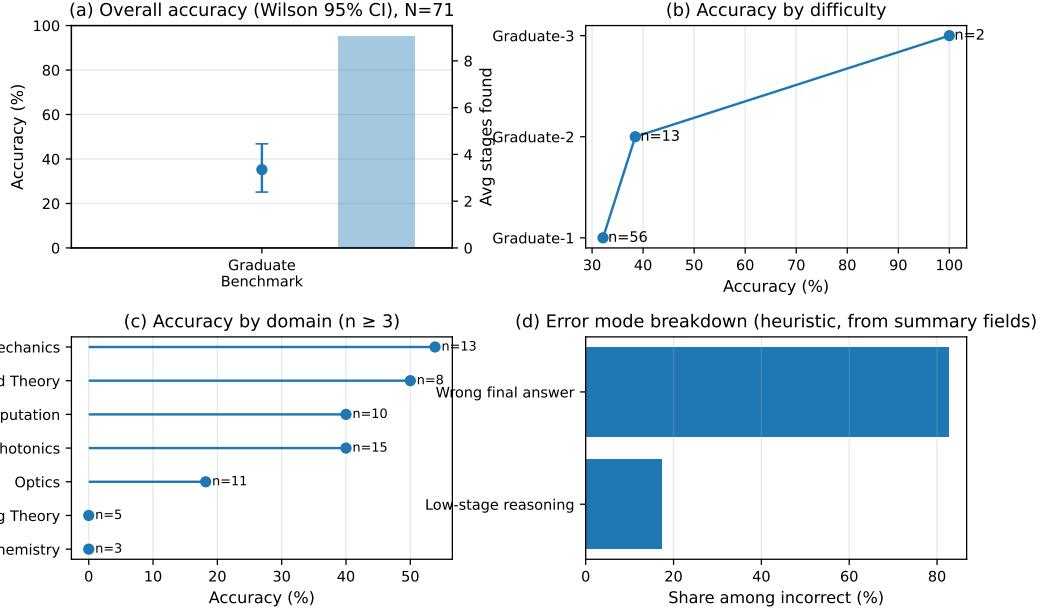


Figure 6: QuantumBench-Grad Graduate Benchmark Main Results: Overall Performance, Difficulty Stratification, and Domain Analysis

From the domain perspective (Table 10), significant variations exist among different lecture-level subdomains, with several domains exhibiting extremely small sample sizes (single-question domains). Therefore, these results are more suitable as diagnostic indicators for identifying weak domains rather than asserting definitive advantages or disadvantages of the model within specific subdomains.

Table 10: QUANTUMBENCH-GRAD Domain-Specific Accuracy (71 Questions)

Field	Accuracy/Total	Acc.
Quantum Optics	1/1	100.0%*
Quantum Mechanics	7/13	53.8%
Condensed Matter	1/2	50.0%
Quantum Field Theory	4/8	50.0%
Photonics	6/15	40.0%
Quantum Computation	4/10	40.0%
Optics	2/11	18.2%
String Theory	0/5	0.0%
Quantum Chemistry	0/3	0.0%
Quantum Information	0/1	0.0%
Atomic Physics	0/1	0.0%
Particle Physics	0/1	0.0%

Note: Single-question domains serve solely as phenomenon records.

Regarding whether structured reasoning is triggered, experiments show that under the prompt constraints of Parts A–D, the model almost always generates segmented structures (Complete = 71/71, 100%), with an average number of stage markers of 9.06 (expected value: 4), ranging from a minimum of 4 to a maximum of 35. This result indicates that multi-stage prompts can stably induce longer, more structured reasoning texts at the

behavioral level; However, the correlation between stage count and accuracy was close to zero, meaning longer reasoning chains do not necessarily yield higher final accuracy. This phenomenon aligns with recent observations that “visible reasoning processes may decouple from actual correctness,” further supporting our motivation for employing a dual-signal evaluation (“structure/form + accuracy”) in open-ended assessments: process visibility alone cannot guarantee content reliability, while final correctness alone struggles to pinpoint failure mechanisms.

## 6 Discussion

Although selective symbol enhancement yields only a +0.78 percentage point overall improvement on the full QuantumBench dataset, its gains exhibit significant domain specificity: benefits are primarily concentrated in Quantum Computation, while varying degrees of negative transfer may occur in subfields such as Optics, Photonics, and Quantum Field Theory. This phenomenon indicates that the effective gains of symbolic tools for large language models are not universally applicable. Instead, they strongly depend on the expressive rigor, executability, and the degree of alignment between the “execution result–option space” of the problem statement. When the problem statement exhibits high structural organization, clear conventions for variables and constants, and the target computation can be directly mapped to a form comparable with options, symbolic execution significantly enhances the reliability of algebraic simplification and numerical consistency. Conversely, in problems relying more on approximations, implicit unit/constant conventions, or diverse expression forms, the symbolic chain accumulates errors during expression parsing, numerization strategies, and option matching. This offsets potential gains from the tool, ultimately manifesting as negative transfer. Therefore, our findings support the view that “tool augmentation must be constrained by task structure and selectively triggered.” Tools should not be treated as universal gain modules but integrated into interpretable gating strategies, activating only on subdistributions where gains are substantial.

Open-ended evaluations further reveal a critical phenomenon elusive to MCQ metrics: models exhibit strong “structural expression capabilities” at the generation level, yet this capability does not equate to reasoning accuracy. Specifically, models consistently generate longer reasoning texts (averaging approximately 12.5 steps) in open-ended experiments, yet LLMJudge exhibited significantly higher variance than AutoEvaluator, with a low correlation coefficient of  $r \approx 0.26$ . This indicates a clear decoupling between “formal structure and content correctness” in this task—structural metrics reliably characterize answer organization and formal completeness but cannot substitute for assessments of physical and mathematical accuracy. Conversely, Judge’s correctness scores are more sensitive to subtle errors in critical derivation steps and logical leaps in reasoning. This observation also provides a unified explanation for the phenomenon observed in the graduate benchmark where “stage coverage approaches perfection yet accuracy still declines”: multi-stage prompts effectively induce models to produce more complete segmented narratives, but models may still commit operator algebra errors, omit boundary conditions, or violate unit consistency at critical derivation points, leading to incorrect final conclusions. In other words, process visibility enhances diagnosability but does not automatically translate into higher result accuracy. Therefore, jointly modeling and evaluating “structural quality signals” and “correctness signals” is essential in quantum reasoning evaluations.

From a difficulty extrapolation perspective, the overall accuracy decline of QuantumBench-Grad compared to the original benchmark indicates that this construction achieves the goal of “difficulty enhancement” at a macro level, though its statistical robustness remains constrained by sample size and annotation granularity. Future work can enhance the benchmark’s diagnostic and interpretability capabilities in three directions: First, expand the scale of higher-difficulty samples and introduce finer-grained annotations for knowledge points and reasoning steps to support more reliable hierarchical comparisons; Second, advance evaluation from “final answer assessment” to “segment-level verifiability,” such as establishing verifiable checkpoints for intermediate conclusions, thereby more directly distinguishing “reasoning chain organization capability” from “critical conclusion correctness”; Third, construct an error typology for quantum reasoning, systematically categorizing common mistakes (e.g., operator algebra errors, approximate conditions, unit/constant conventions, failed expression equivalence judgments) to generate actionable improvement guidelines and more granular capability profiles. Collectively, these findings converge on a key conclusion: evaluating LLMs in quantum domains must expand beyond single-dimensional MCQ accuracy to a multi-perspective framework encompassing “verifiable computation, structured reasoning, and open-ended correctness.” Only then can we authentically delineate the capability boundaries and reliability risks of models tackling highly formalized scientific tasks.

## 7 Limitations and Reproducibility Statement

Although this paper presents relatively systematic experimental results regarding QuantumBench reproducibility, selective symbol enhancement, and open evaluation, several significant limitations remain. First, to balance reproducibility and cost control, the same model (Qwen2.5-7B) was used as both the tested model and the judge in the open evaluation. While this setup facilitates verification and rapid iteration, it inevitably introduces potential “self-evaluation bias.” Previous studies indicate that LLM-as-a-Judge may exhibit structural biases in stylistic expression, verbosity preferences, and consistency. Therefore, the open-source scores in this paper should primarily be interpreted as scalable, comparable relative signals rather than absolute quality metrics equivalent to human annotations. Future work may quantify and mitigate such systematic errors by introducing heterogeneous or stronger models as external Judges and calibrating them against human annotations on small-scale samples.

Second, the “symbolic execution result-option matching” strategy employed in this paper relies on strong engineering assumptions. It remains incomplete for determining equivalence under general expressions, unit and dimensional consistency, and approximate conditions, potentially leading to misjudgments or negative transfer in certain subfields. Consequently, the related results support the methodological conclusion that “tool enhancement requires selective gating and error control,” rather than claiming a complete resolution of quantum symbolic equivalence. Additionally, significance tests at the sub-domain level have not undergone rigorous multiple comparison correction, and related findings should be regarded as exploratory results. Furthermore, the experiments primarily rely on a single model and hardware configuration, limiting their generalizability. Although this paper maximizes reproducibility through fixed random processes and unified statistical metrics, further validation across models, hardware, and inference frameworks remains necessary to clarify the stability and applicability boundaries of the proposed framework.

## 8 Conclusion

The core finding of this paper lies not in proposing a more efficient quantum problem-solving method, but in systematically demonstrating that, within highly formalized quantum science tasks, single-choice accuracy alone is insufficient to characterize the true reasoning capabilities and reliability boundaries of large language models. Through a combined analysis of tool augmentation, open-ended reasoning evaluation, and difficulty extrapolation, this paper reveals structured reasoning behaviors that uncover the complex relationship between verifiable computation and ultimate correctness. This establishes a research paradigm for evaluating LLMs in quantum domains that shifts from “outcome-oriented” to “process and reliability-focused.” We establish a comparable baseline under the 8-way multiple-choice question pipeline consistent with the original benchmark and introduce three complementary extensions: First, we propose a hybrid reasoning mechanism with selective symbol enhancement, boosting accuracy from 38.49% to 39.27% across all 769 questions. We observe stronger gains in the Quantum Computation subdomain, indicating that the tool enhancement’s efficacy in quantum tasks exhibits significant sub-distribution dependence. Second, we constructed an open-ended quantum reasoning task set comprising 165 questions and proposed a dual evaluation framework combining automated metrics with LLM-as-a-Judge. This achieved a composite score of 71.73/100 on a 30-question free-derivation subset, thereby incorporating reasoning process quality into a scalable quantitative assessment. Third, we constructed the 71-question graduate-level QuantumBench-Grad. By imposing multi-stage structured reasoning requirements, we tested models’ difficulty extrapolation capabilities, revealing persistent deficiencies in higher-order knowledge integration and critical derivation reliability. Overall, our findings demonstrate that relying solely on multiple-choice accuracy fails to fully characterize the true capability boundaries of LLMs in quantum domains. More representative evaluations should concurrently assess final answer correctness, computational verifiability, reasoning process quality, and extrapolation performance on higher-difficulty tasks to avoid confusion between structured narrative ability and genuine correctness.

Looking ahead, a more robust quantum domain evaluation framework requires advancement in three key directions: First, incorporating stronger, heterogeneous judges or limited human calibration into open-ended assessments to enhance the credibility and transferability of accuracy scoring; Second, advance evaluation from “result alignment” to “segmented verifiability” by establishing verifiable checkpoints at critical intermediate conclusions, enabling more direct quantification of reasoning chain reliability; Third, expand to larger-scale, higher-difficulty tasks that more closely resemble real scientific workflows (e.g., goal setting, procedural decomposition, verification and explanation, and systematic experiment planning). This ensures benchmarks not only measure answering capability but also reflect the practical applicability and reliability boundaries of models in quantum science research and engineering development scenarios. Overall, the proposed extension framework provides a multi-perspective tool for systematically evaluating LLMs in quantum domains that aligns more closely with task essence. It also lays an experimental foundation for the subsequent development of quantum agent evaluation standards that better approximate authentic scientific workflows.

## References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, et al. (2023). **GPT-4 technical report**. *arXiv preprint arXiv:2303.08774*.
- [2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. (2020). **Language models are few-shot learners**. *arXiv arXiv:2005.14165*.
- [3] Wenhui Chen, Xueguang Ma, Xinyi Wang, William W. Cohen. (2022). **Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks**. *arXiv preprint arXiv:2211.12588*.
- [4] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, et al. (2021). **Training verifiers to solve math word problems**. *arXiv preprint arXiv:2110.14168*. (GSM8K)
- [5] Aaron Grattafiori, Abhimanyu Dubey, Akhil Jauhri, Ankit Pandey, Abhishek Kadian, et al. (2024). **The Llama 3 herd of models**. *arXiv preprint arXiv:2407.21783*.
- [6] Nicolas Dupuis, Luca Buratti, Sanjay Vishwakarma, Aitana Viudes Forrat, et al. (2024). **Qiskit code assistant: Training LLMs for generating quantum computing code**. *arXiv preprint arXiv:2405.19495*.
- [7] Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, et al. (2022). **PAL: Program-aided language models**. *arXiv arXiv:2211.10435*.
- [8] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, et al. (2023). **Gemini: A family of highly capable multimodal models**. *arXiv preprint arXiv:2312.11805*.
- [9] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, et al. (2024). **ToRA: A tool-integrated reasoning agent for mathematical problem solving**. *arXiv arXiv:2309.17452*.
- [10] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Chengjin Xu, et al. (2024). **A survey on LLM-as-a-judge**. *arXiv preprint arXiv:2411.15594*.
- [11] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, et al. (2021). **Measuring massive multitask language understanding**. *arXiv arXiv:2009.03300*. (MMLU)
- [12] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, et al. (2021). **Measuring mathematical problem solving with the MATH dataset**. *arXiv arXiv:2103.03874*.
- [13] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, et al. (2023). **Survey of hallucination in natural language generation**. *arXiv arXiv:2202.03629*.
- [14] Shlomo Kashani (2024). **QuantumLLMInstruct: A 500k LLM Instruction-Tuning Dataset with Problem-Solution Pairs for Quantum Computing**. *arXiv preprint arXiv:2412.20956*.
- [15] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, Yusuke Iwasawa. (2022). **Large language models are zero-shot reasoners**. *arXiv arXiv:2205.11916*.

- [16] Minghao Li, Feifan Song, Yingxiu Zhao, Bowen Yu, et al. (2023). **API-Bank: A comprehensive benchmark for tool-augmented LLMs.** *arXiv arXiv:2304.08244*.
- [17] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, et al. (2023). **G-Eval: NLG evaluation using GPT-4 with better human alignment.** *arXiv arXiv:2303.16634*.
- [18] Potsawee Manakul, Adian Liusie, Mark J. F. Gales, et al. (2023). **SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models.** *arXiv arXiv:2303.08896*.
- [19] McNemar, Q. (1947). **Note on the sampling error of the difference between correlated proportions or percentages.** *Psychometrika*, 12, 153–157. <https://doi.org/10.1007/BF02295996>
- [20] Meurer A, Smith CP, Paprocki M, Čertík O, Kirpichev SB, Rocklin M, Kumar A, Ivanov S, Moore JK, Singh S, Rathnayake T, Vig S, Granger BE, Muller RP, Bonazzi F, Gupta H, Vats S, Johansson F, Pedregosa F, Curry MJ, Terrel AR, Roučka Š, Saboo A, Fernando I, Kulal S, Cimrman R, Scopatz A. (2017). **Sympy: symbolic computing in Python.** *PeerJ Computer Science*, 3, e103. <https://doi.org/10.7717/peerj-cs.103>
- [21] Taku Mikuriya, Tatsuya Ishigaki, Masayuki Kawarada, et al. (2025). **QCoder Benchmark: Bridging Language Generation and Quantum Hardware through Simulator-Based Feedback.** *arXiv preprint arXiv:2510.26101*.
- [22] Shunya Minami, Tatsuya Ishigaki, Ikko Hamamura, Taku Mikuriya, et al. (2025). **QuantumBench: A benchmark for quantum problem solving.** *arXiv preprint arXiv:2511.00092*.
- [23] Haining Pan, Nayantara Mudur, Will Taranto, Maria Tikhanovskaya, et al. (2024). **Quantum Many-Body Physics Calculations with Large Language Models.** *arXiv preprint arXiv:2403.03154*.
- [24] Shishir G. Patil, Tianjun Zhang, Xin Wang, Joseph E. Gonzalez, et al. (2023). **Gorilla: Large language model connected with massive APIs.** *arXiv preprint arXiv:2305.15334*.
- [25] Long Phan, Anish Agrawal, Jack Wei Lun Shi, Alice Gatti, et al. (2025). **Humanity's last exam.** *arXiv preprint arXiv:2501.14249*.
- [26] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, et al. (2023). **ToolLLM: Facilitating large language models to master 16,000+ real-world APIs.** *arXiv preprint arXiv:2307.16789*.
- [27] Qwen Team, An Yang, Baosong Yang, Binyuan Hui, et al. (2024). **Qwen2.5 technical report.** *arXiv preprint arXiv:2412.15115*.
- [28] David Rein, Betty Li Hou, Asa Cooper Stickland, et al. (2023). **GPQA: A graduate-level google-proof Q&A benchmark.** *arXiv preprint arXiv:2311.12022*.

- [29] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, et al. (2023). **Tool-former: Language models can teach themselves to use tools.** *arXiv preprint arXiv:2302.04761*.
- [30] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, et al. (2022). **Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models.** *arXiv preprint arXiv:2206.04615*.
- [31] Andreas Stephan, Dawei Zhu, Matthias Aßenmacher, et al. (2024). **From calculation to adjudication: Examining LLM judges on mathematical reasoning tasks.** *arXiv preprint arXiv:2409.04168*.
- [32] Mirac Suzgun, Nathan Schucher, Sebastian Gehrmann, et al. (2022). **Challenging BIG-Bench tasks and whether chain-of-thought can solve them.** *arXiv preprint arXiv:2210.09261.* (BBH)
- [33] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, et al. (2023). **LLaMA: Open and efficient foundation language models.** *arXiv preprint arXiv:2302.13971*.
- [34] Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, et al. (2023). **SciBench: Evaluating college-level scientific problem-solving abilities of large language models.** *arXiv preprint arXiv:2307.10635*.
- [35] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, et al. (2022). **Self-consistency improves chain of thought reasoning in language models.** *arXiv preprint arXiv:2203.11171*.
- [36] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni,, et al. (2024). **MMLU-Pro: A more robust and challenging multi-task language understanding benchmark.** *arXiv preprint arXiv:2406.01574*.
- [37] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, et al. (2022). **Chain-of-thought prompting elicits reasoning in large language models.** *arXiv preprint arXiv:2201.11903*.
- [38] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, et al. (2022). **ReAct: Synergizing reasoning and acting in language models.** *arXiv preprint arXiv:2210.03629*.
- [39] Jifan Ye, Yanbo Wang, Yue Huang, Dongping Chen, et al. (2024). **Justice or prejudice? Quantifying biases in LLM-as-a-judge.** *arXiv preprint arXiv:2410.02736*.
- [40] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, et al. (2023). **Judging LLM-as-a-judge with MT-Bench and chatbot arena.** *arXiv preprint arXiv:2306.05685*.
- [41] Minhui Zhu, Minyang Tian, Xiaocheng Yang, Tiansi Zhou, et al. (2025). **Probing the critical point (CritPt) of AI reasoning: a frontier physics research benchmark.** *arXiv preprint arXiv:2509.26574*.