

SciTaRC: A Plan-Annotated Scientific Tabular QA Benchmark for Language Reasoning and Complex Computation

Hexuan Wang Yaxuan Ren Srikar Bommireddypalli Shuxian Chen

Adarsh Prabhudesai Rongkun Zhou Elina Baral Philipp Koehn

Center for Language and Speech Processing, Johns Hopkins University

{hwang302, phi}@jhu.edu

Abstract

We introduce SciTaRC, an expert-authored benchmark for question answering over scientific tables that targets composite, multi-step reasoning. To enable fine-grained diagnostic analysis beyond end-task accuracy, SciTaRC pairs each question with a manually constructed reasoning plan and explicit complexity metrics. State-of-the-art models fail on at least 23% of these questions, while highly capable open-weight models like Llama-3.3-70B collapse on 65.5% of the benchmark. Error analysis shows that, in zero-shot settings, failures are driven primarily by question comprehension, where models misinterpret the scientific query and derive the wrong reasoning objective. To determine whether overcoming this gap is sufficient, we use the structured plans to decouple strategy formulation from execution. Surprisingly, providing oracle step-by-step plans yields only limited gains and fails to eliminate the performance gap. This reveals a substantial execution bottleneck: both natural language and code-based methods struggle to reliably carry out long-horizon computational chains over structured data. Ultimately, SciTaRC serves as a rigorous diagnostic testbed for studying both planning and execution in scientific table reasoning.¹

1 Introduction

Artificial intelligence is increasingly tasked with accelerating scientific discovery, powering applications from automated literature reviews to autonomous “AI scientists” (Wang et al., 2023; Lu et al., 2024). Scientific tables are central repositories of empirical knowledge, combining heterogeneous structure, domain-specific notation, and implicit assumptions. Answering questions over scientific tables therefore requires models to simultaneously understand language, retrieve precise evidence, and execute multi-step procedures (Lu et al., 2023; Sui et al., 2024). While large language models (LLMs) have advanced significantly in knowledge recall and multi-step reasoning (Naveed et al., 2024; Ferrag et al., 2025), traditional algorithms often remain superior for tasks requiring long-horizon execution, precise intermediate computation, or strict control flow (Plaat et al., 2025; Wu et al., 2025b). This observation has motivated hybrid approaches that combine neural and symbolic reasoning for agentic AI systems (Wan et al., 2024; Colelough & Regli, 2025). Scientific tables provide a natural stress test for these paradigms by demanding a seamless integration of semantic understanding and logical execution.

However, existing table QA benchmarks capture only part of this challenge. Prior work has advanced evaluation through realistic formatting, multi-table reasoning, and table-text integration (Qiu et al., 2024; Zhu et al., 2025; Zhang et al., 2025), but it provides limited support for *composite reasoning*. In reality, scientific analysis rarely consists of isolated tasks; rather, it requires chaining retrieval, comparison, aggregation, calculation, and conditional

¹Dataset and code: <https://huggingface.co/datasets/JHU-CLSP/SciTaRC> and <https://github.com/JHU-CLSP/SciTaRC>

Question: Which language is the hardest for the Qwen2-Audio in terms of LLM-based accuracy? For which models is a different language the hardest?

F1 / LLM Acc ↑	Cmn	Eng	Fra	CCFQA (Text † / Speech ‡)			Spa	Yue	Avg.
† QA (X→X)				Jpn	Kor	Rus			
GPT-4o-mini	59.2 / 63.6	78.7 / 82.0	74.1 / 73.7	60.3 / 63.4	50.9 / 55.3	62.7 / 42.0	74.1 / 76.3	51.6 / 59.0	63.9 / 64.4
Phi-4-Multimodal	6.9 / 13.4	32.4 / 28.7	32.6 / 15.8	6.5 / 11.9	7.1 / 5.6	22.4 / 11.3	32.0 / 16.6	4.0 / 6.7	18.0 / 13.8
Qwen2-Audio	17.7 / 40.5	46.5 / 54.4	46.0 / 37.2	9.9 / 18.6	10.3 / 11.4	37.3 / 18.2	45.8 / 39.9	9.7 / 24.4	27.9 / 30.6
Qwen2.5-Qwen2-72B	12.7 / 18.1	29.7 / 26.7	33.0 / 13.9	7.0 / 2.0	4.7 / 2.5	27.1 / 4.3	37.0 / 12.8	13.2 / 13.8	20.6 / 11.8
Qwen2.5-Qwen2-72B-Audio	17.7 / 53.2	66.4 / 60.3	58.1 / 39.7	38.1 / 32.9	25.1 / 19.9	48.1 / 22.7	58.9 / 44.1	25.9 / 21.7	39.7 / 31.7
Phi-4-Multimodal-Audio	17.8 / 15.5	19.1	64.4 / 68.8	62.1 / 65.0	57.2 / 59.4	56.1 / 56.6	59.2 / 58.7	61.7 / 61.7	61.7 / 61.7
Qwen2-Audio	25.5 / 22.3	25.1	64.4 / 68.8	62.1 / 65.0	57.2 / 59.4	56.1 / 56.6	59.2 / 58.7	61.7 / 61.7	61.7 / 61.7
Qwen2.5-Omni-3B	12.5 / 11.1	18.6	64.4 / 68.8	62.1 / 65.0	57.2 / 59.4	56.1 / 56.6	59.2 / 58.7	61.7 / 61.7	61.7 / 61.7
Qwen2.5-Omni-7B	45.7 / 42.4	47.9	64.4 / 68.8	62.1 / 65.0	57.2 / 59.4	56.1 / 56.6	59.2 / 58.7	61.7 / 61.7	61.7 / 61.7
† SQA (X→X)									
GPT-4o-mini-Audio	38.6 / 36.6	75.9							
Phi-4-Multimodal	7.1 / 25.3	40.0							
Qwen2-Audio	19.4 / 31.5	48.6							
Qwen2.5-Omni-3B	41.3 / 39.6	59.1							
Qwen2.5-Omni-7B	55.2 / 53.5	68.2							
LLM-SQA (ours)	51.3 / 45.5	74.8							
‡ XSQA (X→8)									
GPT-4o-mini-Audio	44.9 / 34.8	61.7							
Phi-4-Multimodal	20.7 / 6.7	1.5							
Qwen2-Audio	25.9 / 16.6	1.2							
Qwen2.5-Omni-3B	34.6 / 24.1	6.8							
Qwen2.5-Omni-7B	47.6 / 38.8	12.9							
LLM-SQA (ours)	56.6 / 45.8	21.0							

1. Select Target
SELECT Qwen2-Audio (purple highlighted rows)

2. Loop, Extract & Average
LOOP for all languages
SELECT LLM Acc
COMPUTE Avg per lang
Avg(Cmn) = 27.73
...
Avg(Kor) = 7.55
...

3. Find Hardest Language
COMPUTE argmin1 over all languages
Lowest Score → Korean

4. Compare All Models
LOOP for each model:
COMPUTE Avg all langs
COMPUTE argmin2
IF (argmin2 ≠ Kor) → Add
Ex 1: Qwen2.5-Omni-7B
Lowest Avg → Kor (19.83)
(Kor = Kor) → Skip
Ex 2: GPT-4o-mini
Lowest Avg → Rus (50.35)
(Rus ≠ Kor) → Added!

5. Final Answer
Korean; different for GPT-4o-mini-Audio (Cantonese) and GPT-4o-mini (Russian)

Figure 1: **A representative SciTaRC task.** Instead of simple fact retrieval, the models must execute a multi-step procedure that combines targeted lookup, computation, aggregation, and conditional comparison to derive the final answer.

logic within a single, coherent solution process, iterating over variable-length sets and branching on values computed earlier.

To rigorously evaluate this setting, we introduce **SciTaRC** (Scientific Table QA requiring Reasoning and Computation), a benchmark of 371 expert-authored questions for question answering over scientific tables. Beyond end-task evaluation, SciTaRC offers a methodological framework for fine-grained diagnostic analysis: each question is paired with a manually written structured reasoning plan and explicit complexity annotations. Where existing benchmarks report only whether a model failed, these annotations support fine-grained analysis of the conditions under which it fails. Figure 1 illustrates a representative example. Despite recent advances, state-of-the-art models such as GPT-5 achieve only 76.8% accuracy, while highly capable open-weight models like Llama-3.3-70B-Instruct fail on over 65% of questions.

In this paper, we detail the construction of SciTaRC, evaluate a range of state-of-the-art models, and utilize our complexity metrics and structured plans to analyze their failure modes. We find that zero-shot performance is initially constrained by a **comprehension bottleneck**, where models misinterpret the scientific query or misunderstand relevant table entities. However, providing oracle plans to remove this source of error reveals that resolving comprehension is insufficient. Instead, it exposes a severe **execution bottleneck**: models still struggle to faithfully execute long chains of computation and logic, even with the correct reasoning path. Furthermore, imposing rigid external plans on generalist models often incurs a **compliance cost** that disrupts their intrinsic reasoning. Thus, our results suggest that while planning remains an important research frontier, reliable execution over structured data is an equally critical requirement for future autonomous scientific agents.

2 Related Work

Table QA Benchmarks. Early benchmarks such as WikiTableQuestions (Pasupat & Liang, 2015) and TabFact (Chen et al., 2020) focused on relatively clean tables with simple reasoning patterns. More recent work has pushed toward greater complexity by targeting messy real-

world formats (Zhu et al., 2025), multi-table relational reasoning (Qiu et al., 2024), and diverse analytical tasks (Wu et al., 2025a). Domain-specific datasets have also emerged for finance (Katsis et al., 2022) and hierarchical structures (Cheng et al., 2022). Open-domain resources such as DataBench (Osés Grijalba et al., 2024) target tabular QA over consumer, business, and survey data with typed atomic answers. Most similar to our dataset is SCITAT (Zhang et al., 2025), which also addresses scientific tables, but whose questions are LLM-generated and isolate a single reasoning type (e.g., arithmetic or lookup). Across these benchmarks, tables are cleaned into regular grids, a question is answered by a lookup or a single aggregation, and short atomic answers are scored by exact or fuzzy match. SciTaRC instead poses expert-authored questions whose answers require composite reasoning and computation, over tables kept in the raw $\text{\LaTeX}$ of the papers themselves, the form an autonomous system must handle when it reads the literature directly. Because such answers are free-form and multi-part, we score them with an LLM judge validated against human annotation, and we annotate each question with a structured reasoning plan and complexity metrics that make failures diagnosable rather than merely countable.

Reasoning Methods and Diagnostics. Recent surveys (Fang et al., 2024; Wu et al., 2025b) highlight persistent challenges in table understanding. Diagnostic studies reveal specific weaknesses: Sui et al. (2024) shows sensitivity to serialization formats, and Akhtar et al. (2023) demonstrate inconsistency in numerical reasoning. Methodological advances include multi-agent frameworks (Bai et al., 2025; Ma, 2025), specialized reasoning architectures (Xiong et al., 2025), and inference-time scaling (Yang et al., 2025b). SciTaRC complements this work by providing complexity metrics, a diagnostic testbed, and a systematic analysis of failure modes on scientific tables.

Neuro-Symbolic Approaches. Program-based reasoning methods like Program-of-Thought (Chen et al., 2023) and PAL (Gao et al., 2023) generate executable code to disentangle computation from reasoning. DACO (Wu et al., 2024) advances code-based data analysis, while recent work explores LLM-symbolic integration (Kulkarni et al., 2025). Despite these advances, questions remain about whether program-based methods can handle the heterogeneous structures of scientific tables in zero-shot settings. SciTaRC addresses this through manually written structured plans that enable systematic decomposition of planning failures versus execution failures—a distinction not possible with existing benchmarks.

3 Dataset

Our dataset is based on scientific papers sourced from recent arXiv preprints, mainly in the domain of artificial intelligence. We consolidated the downloaded $\text{\LaTeX}$ sources for each paper into a single file, deliberately retaining the raw $\text{\LaTeX}$ format to reflect the authentic, heterogeneous structure of real-world scientific papers, so that the input matches what a system reading the literature directly would receive.

3.1 Question Creation

For each paper, a question author selected 1 to 3 tables, wrote a question, and provided the answer. Each dataset instance includes the arXiv paper ID, question text, gold answer, the $\text{\LaTeX}$ sources for the relevant tables, all tables in the paper, and the full text. The question creators (all authors of this paper) range from undergraduate students to a professor in natural language processing. Questions were designed to be unambiguously answerable using only the tables, captions, and general domain knowledge. Each question was subsequently verified by two reviewers other than its creator. Some questions were too complex for manual calculation. In those cases, short ad-hoc scripts were written to compute the correct gold answer. These scripts were written in a mix of programming languages and are not included in the released dataset.

3.2 Evaluation Protocol

Because answers are free-form and often consist of multiple text and numeric components, matching model generations with the gold answer is non-trivial. For instance, the gold

answer in Figure 1 is: “Korean; different for GPT-4o-mini-Audio (Cantonese) and GPT-4o-mini (Russian)”. However, a perfectly correct model might respond with full, descriptive sentences rather than a concise list. To comprehensively assess performance despite formatting variations, we employ two metrics:

Exact Match (EM): We include Exact Match as a traditional TableQA baseline, requiring strict string equivalence. While standard for simple factoid lookups, EM is fundamentally inadequate for the composite, free-form answers required by SciTaRC. It heavily penalizes models for minor formatting differences, stylistic choices, or extra words despite correct underlying reasoning, thereby measuring superficial formatting brittleness rather than true semantic or mathematical reasoning failures.

LLM-as-a-Judge: To effectively capture semantic and logical correctness, we adopt an LLM-as-a-Judge protocol (Zheng et al., 2023) as our primary metric. We prompt Llama-3.3-70B-Instruct (Grattafiori et al., 2024) with the question, the model’s prediction, and the gold answer (exact prompt in Appendix B). To ensure this automated judge is reliable, we manually annotated a subset of 245 questions. The LLM judge demonstrated high alignment with human evaluations, achieving a 97.6% agreement rate for GPT-4o outputs and 95.5% for Llama-3-8B-Instruct outputs.

3.3 Structured Reasoning Plans

To systematically diagnose whether models fail at strategy formulation (planning) or step-wise implementation (execution), we pair each question with a manually created reasoning plan (see the visual overview in Figure 1, and Appendix C for the complete plan and further examples). Expressing these plans in **structured natural language** rather than a complex programming language removes strict syntax barriers. This deliberate design reduces the risk that observed failures reflect compilation or parsing errors rather than flawed logical execution.

To ensure consistency and eliminate stylistic confounders, the syntax is standardized around five core operations: **SELECT** (retrieves table values), **COMPUTE** (executes mathematical or comparative operations), **IF .. ELSE** (evaluates conditional logic), **LOOP** (iterates over lists), and **RETURN** (outputs the final result). This framework mimics natural problem-solving. Operations can be applied directly over lists without explicit loops, with nested logic indicated simply by indentation. To maintain conversational flow, variables passed sequentially between statements remain implicit. Explicit naming (e.g., `overall_mean`) is strictly reserved for tracking concurrent intermediate results across long execution horizons.

3.4 Quantifying Complexity

To quantify the challenges of SciTaRC, we define metrics characterizing difficulty along two dimensions. Table 1 summarizes the dataset statistics across these dimensions.

Metric	Input Complexity			Reasoning Complexity			
	S_{tab}	S_{cell}	S_{tok}	L_{plan}	I_{retr}	I_{calc}	C_{flow}
Mean	1.3	287.0	2,527	9.8	3.4	4.0	3.0
Max	3	3,338	22,671	27	18	18	12

Table 1: **SciTaRC Dataset Statistics.** These metrics illustrate the composite reasoning focus over large scientific tables. High average values for plan horizon (L_{plan}) and table size (S_{cell}) highlight the requirement for high-fidelity execution across long-horizon scientific contexts.

Input Complexity. We measure the amount of information to be processed:

- **Context Load (S_{tok}):** Total token count of serialized tables, measuring context length.
- **Structural Size (S_{cell}):** Total grid cells parsed from LATEX , a proxy for the number of data items.

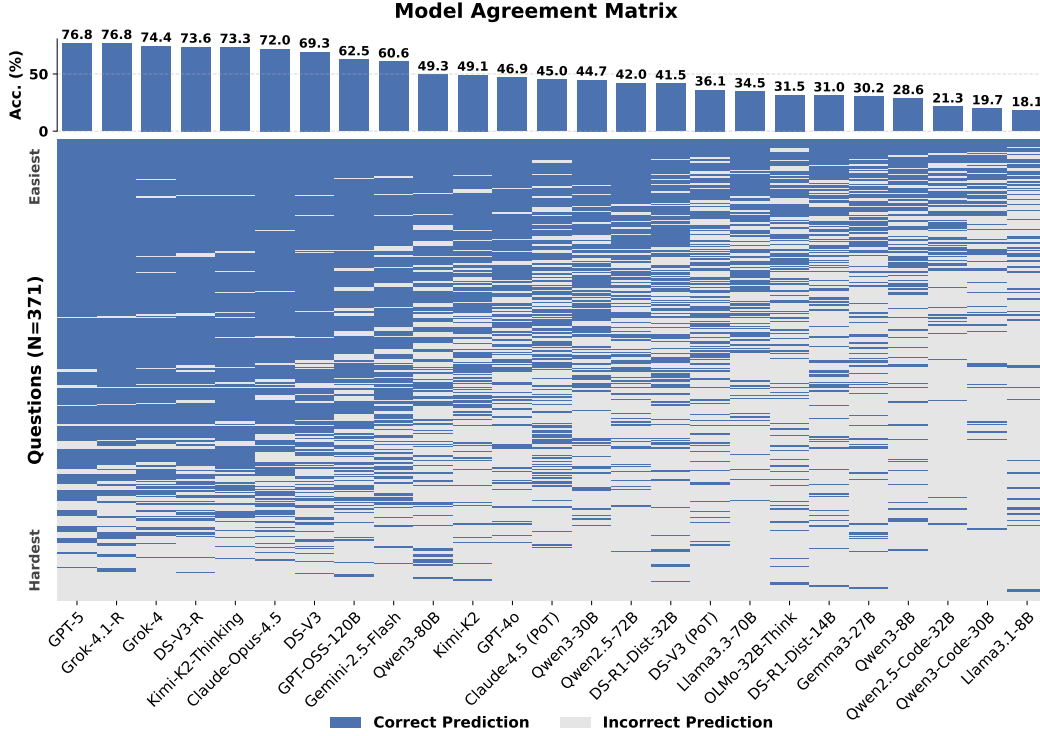


Figure 2: **Model Agreement Matrix**. Rows represent questions sorted by difficulty, defined as the total number of models that answered correctly (easiest at top). Columns represent models sorted by accuracy. **Blue** indicates a correct prediction.

- **Input Scope (S_{tab})**: Number of tables required, measuring cross-table linking needs.

Reasoning Complexity. We quantify algorithmic hardness via ground-truth plans:

- **Calculation Intensity (I_{calc})**: Count of COMPUTE operations, measuring arithmetic burden.
- **Retrieval Demand (I_{retr})**: Count of SELECT operations, measuring retrieval frequency.
- **Plan Horizon (L_{plan})**: Total instruction count, serving as a proxy for reasoning depth.
- **Control Flow (C_{flow})**: Composite score $C_{flow} = N_{loop} + N_{if} + D_{max}$, where N_{loop} and N_{if} count iterative and conditional blocks, and D_{max} represents maximum nesting depth.

4 Benchmarking State-of-the-Art

4.1 Benchmark Setup

We evaluate 24 models spanning two primary categories: **Proprietary Models** (e.g., GPT-5, Grok-4.1) and **Open-Weight Models**. The latter includes both leading generalist models (e.g., DeepSeek-V3.2 (DeepSeek-AI et al., 2025b), Llama 3.3 (Grattafiori et al., 2024), Qwen 3 (Yang et al., 2025a)) and code-optimized specialists (e.g., Qwen-Coder (Hui et al., 2024)).

We employ two zero-shot prompting paradigms tailored to the strengths of each model: **Chain-of-Thought (CoT)** (Wei et al., 2023; Kojima et al., 2023) for generalist models to encourage step-by-step natural language reasoning, and **Program-of-Thought (PoT)** (Chen et al., 2023) for code-optimized models and select generalists (e.g., DeepSeek-V3.2, Claude 4.5 Sonnet) to generate executable Python code. All prompts are provided in Appendix D.

Model Family	Model Name	Method	Size	LLM-Judge (%)	EM (%)
<i>Proprietary Models</i>					
OpenAI	GPT-5	CoT	-	76.8	22.1
xAI	Grok-4.1-fast-reasoning	CoT	-	76.8	2.7
xAI	Grok-4	CoT	-	74.4	34.0
Moonshot	Kimi-K2-Thinking	CoT	-	73.3	24.8
Anthropic	Claude-Opus-4.5	CoT	-	72.0	20.2
Google	Gemini-2.5-Flash	CoT	-	60.6	31.0
Moonshot	Kimi-K2	CoT	-	49.1	26.2
OpenAI	GPT-4o	CoT	-	46.9	10.8
Anthropic	Claude-Sonnet-4.5	PoT	-	45.0	0.5
<i>Open-Weight Models</i>					
DeepSeek	DeepSeek-V3.2 (thinking)	CoT	685B	73.6	32.1
DeepSeek	DeepSeek-V3.2 (non-thinking)	CoT	685B	69.3	11.6
OpenAI	GPT-OSS-120B	CoT	120B	62.5	0.3
Qwen	Qwen3-Next-80B-A3B	CoT	81B	49.3	12.7
Qwen	Qwen3-30B-A3B-2507	CoT	30B	44.7	8.1
Qwen	Qwen2.5-72B-Instruct	CoT	73B	42.0	4.9
DeepSeek	DS-R1-Distill-Qwen-32B	CoT	32B	41.5	10.5
DeepSeek	DeepSeek-V3.2 (non-thinking)	PoT	685B	36.1	0.8
Meta	Llama-3.3-70B-Instruct	CoT	71B	34.5	5.1
AllenAI	OLMo-3.1-32B-Think	CoT	32B	31.5	1.6
DeepSeek	DS-R1-Distill-Qwen-14B	CoT	14B	31.0	7.0
Google	Gemma-3-27B-IT	CoT	27B	30.2	0.8
Qwen	Qwen-3-8B	CoT	8B	28.6	0.0
Qwen	Qwen-2.5-Coder-32B	PoT	32B	21.3	6.7
Qwen	Qwen-3-Coder-30B	PoT	30B	19.7	6.7
Meta	Llama-3.1-8B-Instruct	CoT	8B	18.1	0.0

Table 2: **SciTaRC Main Leaderboard.** Models are grouped by access level and sorted by primary accuracy. All evaluations are zero-shot, using CoT by default and PoT for code-optimized models. Our primary metric, **LLM-Judge (%)**, evaluates semantic and logical correctness; **EM (%)** measures strict string equivalence as a traditional TableQA baseline. Best results per category are **bolded**.

4.2 Zero-Shot Benchmark Performance

Table 2 summarizes the zero-shot performance across all models, while Figure 2 visualizes agreement at the question level. We observe three primary trends:

Reasoning Strategies Amplify Capabilities. Models trained with reinforcement learning (DeepSeek-AI et al., 2025a) for reasoning significantly boost performance, enabling models to outperform larger standard baselines. For instance, Kimi-K2-Thinking (73.3%) exceeds its base model Kimi-K2 (49.1%) by over 24 percentage points. This capability allows efficient scaling: the 32B DeepSeek-R1-Distill reasoning model (41.5%) scores higher than the 71B Llama-3.3 model (34.5%).

Model Scale Matters, but Capabilities Vary. While larger models generally achieve higher overall accuracy, their capabilities are not a strict superset of weaker models. As shown in Figure 2, smaller models occasionally solve “hard” questions that frontier models miss. This stochasticity suggests that reasoning is not a single monotonic skill; different architectures may find different types of logic easier, even if they lack overall capacity.

Zero-Shot Code Execution Struggles. Contrary to the expectation that tables are best handled by code, Program-of-Thought (PoT) performs substantially worse than natural language reasoning. Strong generalist models suffer massive degradation when forced to write code—DeepSeek-V3.2 drops from 69.3% (CoT) to 36.1% (PoT)—and dedicated code

specialists like Qwen-2.5-Coder achieve only 21.3%. Code appears brittle when applied to the ad-hoc, heterogeneous structures of scientific tables, whereas natural language is more robust to interpret the noise.

4.3 Analysis of Difficulty Factors

To understand the mechanisms behind model failure, we first analyze performance against the complexity metrics defined in Section 3.4. We focus on two dimensions: the scale of the input data and the complexity of the required reasoning.

Input Scale Exposes Fragility. Performance generally degrades as the structural size of the table (S_{cell}) increases (see Appendix E for detailed visualizations). This downward trend is consistent across both natural language (CoT) and code generation (PoT) paradigms. However, the data reveals a sharp divide in model robustness. Frontier models exhibit remarkable stability: Grok-4 (+0.1%) and DS-V3 (-2.0%) maintain their accuracy even as tables exceed 400 cells. In contrast, smaller models suffer catastrophic collapse due to context pressure, with Gemma-3-27B dropping by 41.4 percentage points. We observe similar patterns for Context Load (S_{tok}) and Input Scope (S_{tab}).

Impact of Reasoning Complexity. Beyond data scale, we examine how the algorithmic requirements of the question affect performance (Figure 3). We observe a **consistent negative trend** across all metrics, but the drivers of failure differ.

- **Plan Horizon is a Universal Bottleneck (L_{plan}):** Multi-step reasoning proves uniquely difficult for *all* systems. Even robust frontier models like DeepSeek-V3.2 (thinking) drop by 13.4% when moving to long-horizon questions, and the average system degrades by 25.2% to 35.5% depending on the method (CoT or PoT). This suggests that maintaining execution fidelity across long sequences remains a persistent challenge across the board.
- **Code Generation Degrades with Complexity (I_{calc}):** Contrary to the expectation that code handles math better, Program-of-Thought (PoT) degrades faster than natural language on calculation-heavy tasks. PoT accuracy drops by 29.8% (vs 23.2% for CoT), likely because generating the correct syntax for long operation chains zero-shot is more prone to error than natural language reasoning.
- **Complex Logic Separates Reasoning Models (C_{flow}):** Questions with nested loops or conditionals reveal a stark divergence. While standard models like GPT-4o suffer significant degradation (-29.5%), reasoning-optimized models like Kimi-Thinking and DeepSeek-V3.2 remain remarkably robust (+0.9% and -2.6% respectively), indicating their specific training enables them to handle structural logic that confuses generalist models.

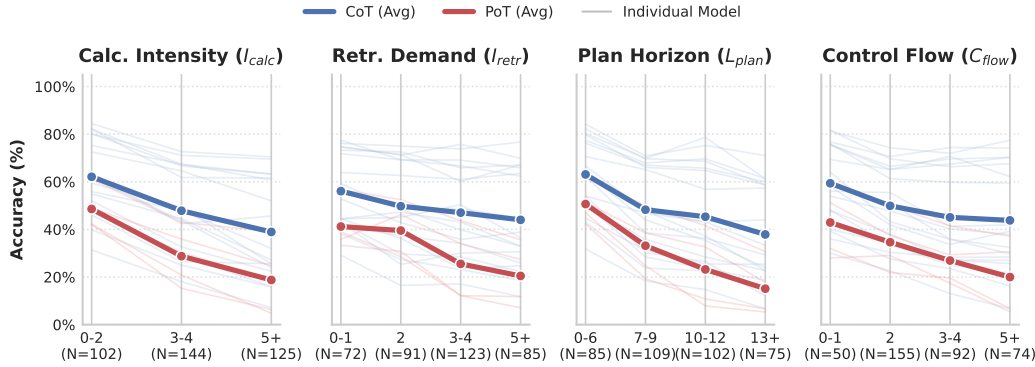


Figure 3: **Impact of Reasoning Complexity.** We measure accuracy against four metrics: Calculation Intensity (I_{calc}), Retrieval Demand (I_{retr}), Plan Horizon (L_{plan}), and Control Flow (C_{flow}). Across all dimensions, increased complexity correlates with lower performance.

5 Error Analysis: The Comprehension Bottleneck

To understand the widespread zero-shot failures observed in Section 4.2, we conducted a qualitative analysis of 98 incorrect outputs from DeepSeek-V3.2 (thinking). We categorize these failures using a four-class taxonomy:

- **Comprehension:** Misunderstanding the question intent, constraints, or table attributes, leading to the wrong objective.
- **Localization:** Selecting incorrect rows, columns, or values despite a correct objective.
- **Calculation:** Incorrect numeric result given the correct evidence and operation.
- **Memory:** Failing to retain and apply intermediate results across multi-step plans.

Each failure is assigned based on its earliest unrecoverable point. Table 3 shows that **Comprehension** errors dominate (73%), revealing that models struggle most with semantic grounding—translating complex scientific queries into correct operational objectives. Conversely, the low **Localization** rate (7%) demonstrates that parsing raw $\text{\LaTeX}$ structure is not the primary hurdle. We note that comprehension is the first step, and the model may have made subsequent errors which we did not count.

Category	Ratio
Comprehension	73%
Calculation	17%
Localization	7%
Memory	3%

Table 3: Error distribution.

6 Disentangling Reasoning: The Planning Ablation

Section 5 revealed that zero-shot models fail predominantly at comprehension—translating queries into actionable objectives. This raises a critical hypothesis: if we completely bypass this barrier by providing an explicit, structured reasoning strategy, the performance gap should theoretically close. To test this, we systematically decouple strategy formulation from mechanical execution via a controlled planning ablation.

6.1 Experimental Design

We evaluate four representative models (Qwen2.5-Coder-32B, DS-R1-Distill-32B, Qwen3-30B, Qwen3-80B) across three settings (prompts in Appendix D): (1) **Direct Inference** (S_{direct}), the baseline zero-shot condition solving the problem in a single pass; (2) **Autonomous Planning** (S_{auto}), a two-stage pipeline where the model generates a plan (Section 3.3) and then executes it in a fresh context; and (3) **Oracle Planning** (S_{oracle}), which injects the ground-truth plan into the execution context. Crucially, identical generation prompts for S_{auto} and S_{oracle} isolate the impact of plan quality.

To analyze planning efficacy, we define the **Gain Curve**. We rank questions $q_1, \dots, q_N$ by baseline difficulty under S_{direct} . Let $y_i \in \{0, 1\}$ denote correctness for a given model on q_i . We define the Success Probability as a centered sliding-window average $P_{\text{success}}(i) = \frac{1}{|W_i|} \sum_{j \in W_i} y_j$, where $W_i = \{j : \max(1, i - \lfloor w/2 \rfloor) \leq j \leq \min(N, i + \lfloor w/2 \rfloor)\}$ and $w = 50$. This filters stochastic noise to visualize robust trends across the difficulty spectrum.

6.2 Results: The Execution Bottleneck

Execution is the Hard Ceiling. Table 4 and Figure 4 reveal that eliminating the comprehension barrier does not close the performance gap. Even when provided with a perfect Oracle plan to eliminate comprehension errors, models fail to exceed a hard ceiling of $\sim 49\%$ accuracy. As Figure 4 visually decomposes, absolute **Execution Failure (Red)**—failing under both autonomous and Oracle conditions—dominates across all models, consistently exceeding Plan Failure (Yellow). For example, Qwen-Coder fails over 65% of questions due to execution errors, compared to just $\sim 12\%$ that are recoverable from flawed planning strategies. This exposes a severe **Execution Bottleneck**: models fundamentally struggle

Model	Accuracy (%)			Diagnostics		
	Direct	Auto	Oracle	Δ_{Auto}	Δ_{Max}	Δ_{Plan}
Qwen2.5-Coder-32B	21.3	23.2	27.5	+1.9	+6.2	4.3
DS-R1-Dist-32B	41.5	36.1	45.8	-5.4	+4.3	9.7
Qwen3-30B	44.7	34.8	45.3	-10.0	+0.6	10.5
Qwen3-80B	49.3	41.8	49.1	-7.5	-0.2	7.3

Table 4: **Impact of Planning and Upper Bounds. Diagnostics:** Δ_{Auto} denotes the gain from autonomous planning (Auto – Direct). Δ_{Max} denotes the absolute potential gain when the comprehension bottleneck is eliminated (Oracle – Direct). Δ_{Plan} represents the performance gap caused by flawed strategy formulation (Oracle – Auto). Even with perfect Oracle guidance, models hit a hard ceiling at $\sim 49\%$ accuracy.

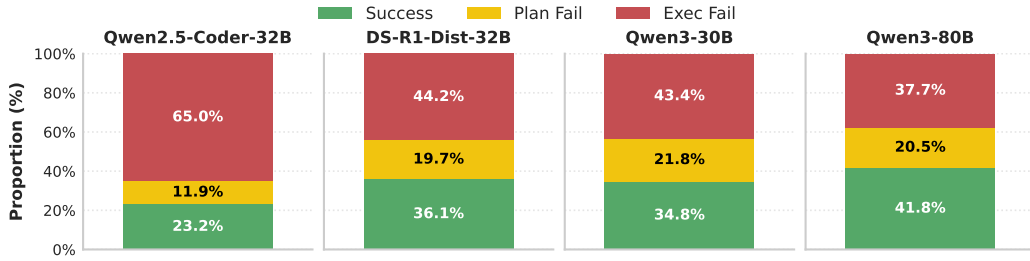


Figure 4: **Decomposition of Failure Modes.** We classify failure modes by testing if the Oracle plan rescues autonomous (S_{auto}) failures. **Plan Failure (Yellow):** Failed autonomously but succeeded with an Oracle plan (flawed strategy). **Execution Failure (Red):** Failed under both autonomous and Oracle conditions (unrecoverable by planning). The dominant red bars confirm that execution capacity, not strategy, is the primary bottleneck.

significantly more to *faithfully execute* multi-step numerical and logical operations than to *formulate* the required strategy.

The Cost of Compliance. Figure 5 and Table 4 illustrate a distinct capability trade-off. Explicit planning aids the code-specialized Qwen-Coder (+1.9% Auto, +6.2% Oracle), confirming that structured planning reduces errors in programmatic generation. Conversely, generalist language models consistently regress (-5.4% to -10.0%). As visualized by the red shading in the Gain Curves, this highlights a **Compliance Cost**: forcing a generalist model to strictly follow a rigid algorithmic path actively disrupts its natural reasoning flow on simpler tasks where unguided intuition already suffices.

Moreover, for the largest generalist model (Qwen3-80B), even perfect Oracle guidance (49.1%) fails to surpass its zero-shot baseline (49.3%). This suggests that an external strategy must *naturally align* with a model’s intrinsic reasoning mode to be effective; otherwise, the cognitive overhead of strict compliance outweighs the benefit of strategic clarity. However, for smaller architectures, the Oracle Plan remains an upper bound, revealing significant latent execution capability that autonomous planning fails to unlock.

7 Conclusion

We presented SciTaRC, a novel expert-authored benchmark for scientific table question answering that targets composite, multi-step reasoning over heterogeneous structured data. Beyond questions and answers, SciTaRC supplies a structured reasoning plan and complexity metrics for every instance, which is what allows failures to be diagnosed rather than merely counted. Our experiments show that this setting challenges even frontier models, which fail at least 23% of the benchmark. Our analyses reveal a two-stage failure pattern: zero-shot models frequently misinterpret the reasoning objective, and supplying

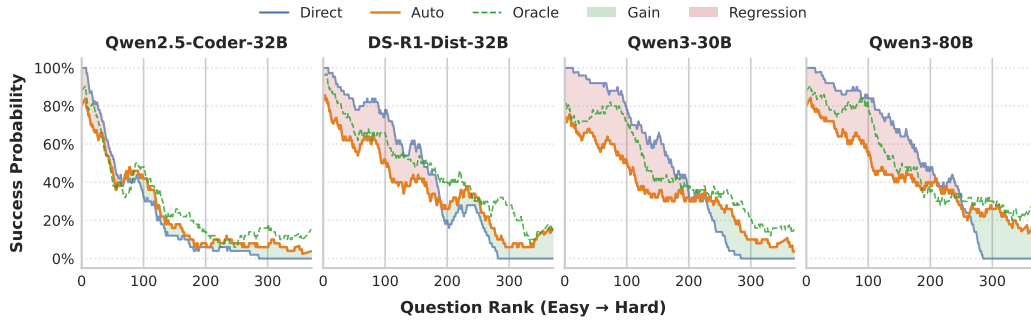


Figure 5: **Performance Gain Curves.** We plot success probability across the difficulty spectrum (see Appendix F for question-level changes). **Code Models (Left)** show consistent gains from planning on hard tasks. **Generalist Models (Right)** often regress on easy tasks (Red Shading) and gain on original unsolved tasks (Green Shading)

oracle plans to remove that barrier still leaves a large gap, indicating that faithful execution, not strategy formation, is the binding constraint. Isolating these limitations highlights two priorities for future architecture design. First, to resolve the compliance cost in generalist models, future systems need methods that provide strategic guidance without disrupting intrinsic reasoning. Second, the brittleness of zero-shot code generation points to the need for deeper neuro-symbolic integration, moving beyond isolated code execution. As the field advances toward autonomous AI scientists, reliable execution of complex mathematical and logical operations over structured empirical data remains a prerequisite that current systems do not meet. SciTaRC provides a rigorous testbed to measure and drive this progress.

Acknowledgments

This work is supported in part by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001125C0304. This work is also supported in part by an Amazon Initiative for Artificial Intelligence (AI2AI) Faculty Research Award.

References

- Mubashara Akhtar, Abhilash Shankarampeta, Vivek Gupta, Arpit Patil, Oana Cocarascu, and Elena Simperl. Exploring the numerical reasoning capabilities of language models: A comprehensive analysis on tabular data. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 15391–15405, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.1028. URL <https://aclanthology.org/2023.findings-emnlp.1028/>.
- Ye Bai, Minghan Wang, and Thuy-Trang Vu. MAPLE: Multi-agent adaptive planning with long-term memory for table reasoning. In Jonathan K. Kummerfeld, Aditya Joshi, and Mark Dras (eds.), *Proceedings of the 23rd Annual Workshop of the Australasian Language Technology Association*, pp. 132–158, Sydney, Australia, November 2025. Association for Computational Linguistics. ISBN 1834-7037. URL <https://aclanthology.org/2025.alta-main.10/>.
- Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyu Zhou, and William Yang Wang. Tabfact: A large-scale dataset for table-based fact verification, 2020. URL <https://arxiv.org/abs/1909.02164>.
- Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks, 2023. URL <https://arxiv.org/abs/2211.12588>.

Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. HiTab: A hierarchical table dataset for question answering and natural language generation. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1094–1110, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.78. URL <https://aclanthology.org/2022.acl-long.78/>.

Brandon C. Colelough and William Regli. Neuro-Symbolic AI in 2024: A Systematic Review, April 2025. URL <http://arxiv.org/abs/2501.05435>. arXiv:2501.05435 [cs].

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanbiao Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yudian Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025a. URL <https://arxiv.org/abs/2501.12948>.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Erhang Li, Fangqi Zhou, Fangyun Lin, Fucong Dai, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoran Wei, Haowei Zhang, Haowen Luo, Haozhe Ji, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, Jialiang Huang, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jingchang Chen, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jinhua Zhu, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexin Huang, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Liang Zhao, Liangsheng Yin, Lihua Guo, Lingxiao Luo, Linwang Ma, Litong Wang, Liyue Zhang, M. S. Di, M. Y. Xu, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Panpan Huang, Peixin Cong, Peiyi Wang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, S. H. Liu, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoifei Cai, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Songyang Zhou, Tao Ni,

- Tao Yun, Tian Pei, Tian Ye, Tianyuan Yue, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjun Gao, Wentao Zhang, Xi Gao, Xiangwen Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyuan Li, Xu Chen, Xuecheng Su, Xuehai Pan, Xuheng Lin, Xuwei Fu, Y. Q. Wang, Yang Zhang, Yanhong Xu, Yanru Ma, Yao Li, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yiliang Xiong, Ying He, Ying Zhou, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuduan Wang, Yue Gong, Yuhan Wu, Yuheng Zou, Yukun Li, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehua Zhao, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhiyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Zizheng Pan, Zongqing Yao, Bei Feng, Hui Li, J. L. Cai, Jiaqi Ni, Lei Xu, Meng Li, Ning Tian, R. J. Chen, R. L. Jin, S. S. Li, Shuang Zhou, Tianyu Sun, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xinnan Song, Xinyi Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, Dongjie Ji, Jian Liang, Jianzhong Guo, Jin Chen, Leyi Xia, Miaojun Wang, Mingming Li, Peng Zhang, Ruyi Chen, Shangmian Sun, Shaoqing Wu, Shengfeng Ye, T. Wang, W. L. Xiao, Wei An, Xianzu Wang, Xiaowen Sun, Xiaoxiang Wang, Ying Tang, Yukun Zha, Zekai Zhang, Zhe Ju, Zhen Zhang, and Zihua Qu. Deepseek-v3.2: Pushing the frontier of open large language models, 2025b. URL <https://arxiv.org/abs/2512.02556>.
- Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, and Christos Faloutsos. Large language models (llms) on tabular data: Prediction, generation, and understanding – a survey, 2024. URL <https://arxiv.org/abs/2402.17944>.
- Mohamed Amine Ferrag, Norbert Tihanyi, and Merouane Debbah. Reasoning Beyond Limits: Advances and Open Problems for LLMs, March 2025. URL <http://arxiv.org/abs/2503.22732>. arXiv:2503.22732 [cs].
- Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models, 2023. URL <https://arxiv.org/abs/2211.10435>.
- 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, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Keshal Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin

Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath R-parthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkan Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippas Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Goya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshv, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu

- Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL <https://arxiv.org/abs/2407.21783>.
- Binyuan Hui, Jian Yang, Zeyu Cui, Jiayi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, Yunlong Feng, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. Qwen2.5-coder technical report, 2024. URL <https://arxiv.org/abs/2409.12186>.
- Yannis Katsis, Saneem Chemmengath, Vishwajeet Kumar, Samarth Bharadwaj, Mustafa Canim, Michael Glass, Alfio Gliozzo, Feifei Pan, Jaydeep Sen, Karthik Sankaranarayanan, and Soumen Chakrabarti. AIT-QA: Question answering dataset over complex tables in the airline industry. In Anastassia Loukina, Rashmi Gangadharaiah, and Bonan Min (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pp. 305–314, Hybrid: Seattle, Washington + Online, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-industry.34. URL <https://aclanthology.org/2022.naacl-industry.34/>.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023. URL <https://arxiv.org/abs/2205.11916>.
- Atharv Kulkarni, Kushagra Dixit, Vivek Srikumar, Dan Roth, and Vivek Gupta. LLM-symbolic integration for robust temporal tabular reasoning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19914–19940, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1022. URL <https://aclanthology.org/2025.findings-acl.1022/>.
- Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.
- Xinyuan Lu, Liangming Pan, Qian Liu, Preslav Nakov, and Min-Yen Kan. SCITAB: A Challenging Benchmark for Compositional Reasoning and Claim Verification on Scientific Tables, October 2023. URL <http://arxiv.org/abs/2305.13186>. arXiv:2305.13186 [cs].
- Yiran Rex Ma. Paneltr: Zero-shot table reasoning framework through multi-agent scientific discussion. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, June 2025. doi: 10.1109/ijcnn64981.2025.11228106. URL <http://dx.doi.org/10.1109/IJCNN64981.2025.11228106>.
- Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A Comprehensive Overview

- of Large Language Models, October 2024. URL <http://arxiv.org/abs/2307.06435>. arXiv:2307.06435 [cs].
- Jorge Osés Grijalba, L. Alfonso Ureña-López, Eugenio Martínez Cámara, and Jose Camacho-Collados. Question answering over tabular data with DataBench: A large-scale empirical evaluation of LLMs. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 13471–13488, Torino, Italia, May 2024. ELRA and ICCL. URL <https://aclanthology.org/2024.lrec-main.1179/>.
- Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In Chengqing Zong and Michael Strube (eds.), *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1470–1480, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1142. URL <https://aclanthology.org/P15-1142/>.
- Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Back. Multi-Step Reasoning with Large Language Models, a Survey, November 2025. URL <http://arxiv.org/abs/2407.11511>. arXiv:2407.11511 [cs].
- Zipeng Qiu, You Peng, Guangxin He, Binhang Yuan, and Chen Wang. Tqa-bench: Evaluating llms for multi-table question answering with scalable context and symbolic extension, 2024. URL <https://arxiv.org/abs/2411.19504>.
- Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study, 2024. URL <https://arxiv.org/abs/2305.13062>.
- Zishen Wan, Che-Kai Liu, Hanchen Yang, Chaojian Li, Haoran You, Yonggan Fu, Cheng Wan, Tushar Krishna, Yingyan Lin, and Arijit Raychowdhury. Towards Cognitive AI Systems: a Survey and Prospective on Neuro-Symbolic AI, January 2024. URL <http://arxiv.org/abs/2401.01040>. arXiv:2401.01040 [cs].
- Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL <https://arxiv.org/abs/2201.11903>.
- Xianjie Wu, Jian Yang, Linzheng Chai, Ge Zhang, Jiaheng Liu, Xeron Du, Di Liang, Daixin Shu, Xianfu Cheng, Tianzhen Sun, Tongliang Li, Zhoujun Li, and Guanglin Niu. Tablebench: a comprehensive and complex benchmark for table question answering. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI’25/IAAI’25/EAAI’25*. AAAI Press, 2025a. ISBN 978-1-57735-897-8. doi: 10.1609/aaai.v39i24.34739. URL <https://doi.org/10.1609/aaai.v39i24.34739>.
- Xiaofeng Wu, Alan Ritter, and Wei Xu. Tabular data understanding with llms: A survey of recent advances and challenges, 2025b. URL <https://arxiv.org/abs/2508.00217>.
- Xueqing Wu, Rui Zheng, Jingzhen Sha, Te-Lin Wu, Hanyu Zhou, Mohan Tang, Kai-Wei Chang, Nanyun Peng, and Haoran Huang. Daco: Towards application-driven and comprehensive data analysis via code generation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 90661–90682. Curran Associates, Inc., 2024. doi: 10.52202/079017-2878. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/a4cb1444fb05839d0113d2773da88a49-Paper-Datasets_and_Benchmarks_Track.pdf.

- Sishi Xiong, Dakai Wang, Yu Zhao, Jie Zhang, Changzai Pan, Haowei He, Xiangyu Li, Wenhan Chang, Zhongjiang He, Shuangyong Song, and Yongxiang Li. Tablereasoner: Advancing table reasoning framework with large language models, 2025. URL <https://arxiv.org/abs/2507.08046>.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a. URL <https://arxiv.org/abs/2505.09388>.
- Zheyuan Yang, Lyuhao Chen, Arman Cohan, and Yilun Zhao. Table-r1: Inference-time scaling for table reasoning tasks. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20616–20635, Suzhou, China, November 2025b. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1040. URL <https://aclanthology.org/2025.emnlp-main.1040/>.
- Xuanliang Zhang, Dingzirui Wang, Baoxin Wang, Longxu Dou, Xinyuan Lu, Keyan Xu, Dayong Wu, and Qingfu Zhu. SCITAT: A question answering benchmark for scientific tables and text covering diverse reasoning types. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3859–3881, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.199. URL <https://aclanthology.org/2025.findings-acl.199/>.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL <https://arxiv.org/abs/2306.05685>.
- Junnan Zhu, Jingyi Wang, Bohan Yu, Xiaoyu Wu, Junbo Li, Lei Wang, and Nan Xu. TableEval: A real-world benchmark for complex, multilingual, and multi-structured table question answering. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7137–7157, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.363. URL <https://aclanthology.org/2025.emnlp-main.363/>.

A LLM Usage Disclosure

LLMs were used for generating plots, formatting LaTeX tables, and limited code drafting during development. All experimental design decisions, data construction, analysis, and final code were reviewed, verified, and finalized by the authors. LLM tools did not contribute to the scientific claims or conclusions presented in this paper. In addition, Llama-3.3-70B-Instruct serves as the automatic judge in our primary evaluation metric (Appendix B).

B Evaluation Prompt

We designed the evaluation prompt to address three specific scoring challenges in scientific table reasoning:

- **Stylistic Invariance:** Surface-level variations (e.g., typos, phrasing) are ignored provided the core factual content is accurate.
- **Numerical Tolerance:** To account for rounding differences, the protocol permits a deviation of ± 1 in the final significant digit.
- **Strictness on Extra Information:** To avoid rewarding recall bias, the inclusion of unrequested components not present in the ground truth is penalized.

Note on Binarization: While the prompt utilizes a ternary scale (0.0, 0.5, 1.0) to allow the judge to explicitly categorize partial matches, we strictly binarize these scores for the final leaderboard. The ternary structure serves to isolate partially correct answers from the fully correct class; for our reported metrics, only a score of 1.0 is treated as a success (1), while 0.5 and 0.0 are treated as failures (0).

C Example Questions and Reasoning Plans

Tables 6 and 7 present five complete instances spanning the complexity range of SciTaRC, each with its question, gold answer, expert-authored structured reasoning plan, and plan-derived metrics. The fourth instance is the task shown in Figure 1. Together they illustrate that question length is a poor proxy for difficulty: the second and fifth questions are stated in under thirty words, yet their plans require nested iteration over variable-length sets and branching on values computed earlier.

D Inference Prompts

In the following templates, [Table Data] represents the serialized text of the relevant tables, and [Question] represents the target query.

D.1 Direct Inference (No Plan)

These prompts establish the baseline performance for generalist (CoT) and specialist (PoT) models.

D.2 Plan Generation

This prompt is used in the **Autonomous Planning** setting to generate the structured reasoning path.

D.3 Plan-Guided Inference

These prompts inject the generated (Autonomous) or ground-truth (Oracle) plan into the context to guide the model’s execution.

System Prompt
You are evaluating a table QA system. Compare the predicted answer against the ground truth answer. SCORING RULES (follow exactly): - 1.0: CORRECT - Prediction contains the same core factual information as ground truth - 0.5: PARTIALLY CORRECT - Prediction has some correct information but missing key parts - 0.0: INCORRECT - Prediction is factually wrong or completely unrelated CRITICAL REQUIREMENTS: - ALL components in ground truth MUST be mentioned in prediction - If ground truth has multiple parts, prediction must include ALL parts - Missing ANY component = maximum score 0.5 - Adding extra components not in ground truth = maximum score 0.5 - Numbers must be nearly identical - only final digit can differ by ± 1 EXAMPLES: - Ground: "Japanese" Prediction: "None. HTML2Text performs best in Japanese" -> 1.0 - Ground: "Adam" Prediction: "Adam optimizer with 0.001 learning rate" -> 1.0 - Ground: "ResNet-50" Prediction: "ResNet-50 and DenseNet-121" -> 0.5 - Ground: "4.2" Prediction: "approximately 3.8" -> 0.0 CRITICAL: Return ONLY this exact JSON format with start/end tags: [Evaluation Start] "reasoning": "Brief explanation", "score": 1.0 [Evaluation End] Now evaluate: Question: {question} Ground Truth: {ground_truth} Prediction: {prediction} [Evaluation Start]

Table 5: The LLM-as-a-Judge prompt used for automatic evaluation.

E Extended Complexity Analysis

Figure 6 provides a comprehensive view of model performance against three dimensions of input scale: **Structural Size** (S_{cell}), **Context Load** (S_{tok}), and **Input Scope** (S_{tab}).

Context Load (S_{tok}). Consistent with cell count, increasing the token length of the input exerts significant pressure on weaker models. Qwen3-8B drops by 30.0% and Qwen3-Code by 36.0% when moving from short ($< 1k$) to long ($> 3.3k$) contexts. However, reasoning-optimized frontier models remain immune to this noise, with DeepSeek-V3.2 (+1.5%) and Grok-4.1-R (+1.1%) showing slight performance gains in longer contexts.

Input Scope (S_{tab}). Increasing the number of distinct tables (from 1 to 3+) reveals a divergence in capabilities. While typical models degrade, top reasoners like DeepSeek-V3.2 (+13.0%) perform better on multi-table questions. This suggests that strong models possess the reasoning depth to handle the structured comparisons in these questions, successfully linking information across disjoint tables where weaker models lose context.

F Extended Planning Analysis

To complement the probabilistic Gain Curves, we present differential heatmaps ($N = 371$) in Figures 7–9. These plots visualize the exact state changes for individual questions, distinguishing between "Gain" (Wrong $\rightarrow$ Right) and "Regression" (Right $\rightarrow$ Wrong).

Q: How much lower is the F1-score of the second-best model compared to GPT-4? **A:** 0.1066

SELECT F1-scores for all models
 COMPUTE sort models by F1-score in descending order
 COMPUTE difference = F1-score of GPT-4 - F1-score of second-best model
 RETURN difference

$L_{plan} = 4, I_{retr} = 1, I_{calc} = 2, C_{flow} = 0, S_{tab} = 1, S_{cell} = 130$

Q: Which zero-shot models outperform any of the fine-tuned models on any metric and language direction? **A:** InternVL2.5-38B, InternVL2.5-78B, and Qwen2.5-VL-72B (on chrF++ EN-ZH)

SELECT Zero-shot Systems, all metrics, both language directions
 LOOP for each zero-shot model
 SELECT scores for current zero-shot Model
 LOOP for each metric and language direction
 SELECT Fine-tuned models scores
 COMPUTE max_ft_score = max fine-tuned model scores
 COMPUTE diff = zero-shot model score - max_ft_score
 IF diff > 0
 COMPUTE add zero-shot model to result_list
 RETURN result_list

$L_{plan} = 10, I_{retr} = 4, I_{calc} = 3, C_{flow} = 6, S_{tab} = 1, S_{cell} = 160$

Q: Considering only the models that appear in all three tables, between the Single-Domain→Cross-Domain and the Cross-Domain→Multi-Domain transitions, and considering all OOD ratios, how many of those model-transition-OOD-ratios yield an increase in overall F1? **A:** 12

SELECT Models present in all 3 Tables
 LOOP for each Model
 SELECT 60% OOD Ratio from Table 1, 80% from Table 2, and 90% from Table 3
 LOOP for each OOD Ratio
 SELECT Overall F1 scores for SD, CD, and MD
 LOOP for each Transition
 IF Transition is "Single-Domain→Cross-Domain"
 COMPUTE increase = f1_cd - f1_sd
 IF Transition is "Cross-Domain→Multi-Domain"
 COMPUTE increase = f1_md - f1_cd
 IF increase > 0
 COMPUTE increment positive_count by 1
 RETURN positive_count

$L_{plan} = 13, I_{retr} = 6, I_{calc} = 3, C_{flow} = 10, S_{tab} = 3, S_{cell} = 336$

Table 6: **Complete SciTaRC instances across the complexity range (1 of 2).** Each block shows the question, gold answer, expert-authored structured reasoning plan, and the plan-derived complexity metrics defined in Section 3.4. Instances are ordered by plan horizon (L_{plan}).

Q: On which test sets are bigger models (either model size or longer token windows) always at least as good as smaller models of the same kind? **A:** HellaSwag, PiQA

```

SELECT all test sets
SELECT all model families
SELECT all token-window groups
LOOP for each test set
  COMPUTE violation_exists = false
  LOOP for each model family
    SELECT models in this family ordered from smallest to largest
    LOOP for each adjacent model pair
      IF larger_model_score < smaller_model_score on this test set
        COMPUTE violation_exists = true
  LOOP for each token-window group
    SELECT models in this group ordered from shortest to longest
    LOOP for each adjacent model pair
      IF longer_window_score < shorter_window_score on this test set
        COMPUTE violation_exists = true
  IF violation_exists = false
    SELECT this test set as valid
RETURN valid test sets

```

$L_{plan} = 18, I_{retr} = 9, I_{calc} = 3, C_{flow} = 12, S_{tab} = 1, S_{cell} = 165$ (highest C_{flow} in SciTaRC)

Q: Which language is the hardest for the Qwen2-Audio model in terms of LLM-based accuracy? For which models is a different language the hardest? **A:** Korean; different for GPT-4o-mini-Audio (Cantonese) and GPT-4o-mini (Russian)

```

SELECT Qwen2-Audio, all languages
LOOP for each language
  SELECT LLM accuracy in all settings
  COMPUTE average1
COMPUTE argmin1
SELECT all models
LOOP for each model
  SELECT all languages
  LOOP for each language
    SELECT LLM accuracy in all settings
    COMPUTE average2
  COMPUTE argmin2
  IF argmin2 is not the same as argmin1
    COMPUTE add this model to list
RETURN argmin1, list

```

$L_{plan} = 15, I_{retr} = 6, I_{calc} = 5, C_{flow} = 6, S_{tab} = 1, S_{cell} = 280$ (the task in Figure 1)

Table 7: **Complete SciTaRC instances across the complexity range (2 of 2).** Continued from Table 6.

Standard Chain-of-Thought (CoT)

You are a helpful science assistant who answers questions about information in tables.

Here is the relevant tabular data:

[Table Data]

You may think through the question step by step. Your final response should be "Answer:" followed by the answer.

Question: [Question]

Standard Program-of-Thought (PoT)

Write Python code to analyze the table and answer the question.

Table:

[Table Data]

Question: [Question]

Write Python code that prints the final answer.

Table 8: Baseline prompts for Language and Code execution without explicit planning.

Plan Generation Prompt

You are a helpful assistant that creates step-by-step plans for answering questions about tables.

Operations:

- SELECT: Extract specific values or subsets from the table
- LOOP: Iterate over items (e.g., LOOP for each model)
- COMPUTE: Numerical operations (average, argmax, difference, etc.)
- IF: Conditional logic
- RETURN: State what to return for the final answer

Rules:

- Use exact syntax: "LOOP for each", "SELECT", "COMPUTE", "IF", "RETURN"
- Indent after LOOP and IF
- Use actual entity names, not generic "row" or "column"

Example Plan:

LOOP for each model

 SELECT accuracy on Dataset A

 SELECT accuracy on Dataset B

 COMPUTE average across datasets

COMPUTE best_model = argmax(average)

RETURN model with highest average accuracy

Table:

[Table Data]

Question: [Question]

Write a concise step-by-step plan using the operations above. Do not solve it, just write the plan:

Table 9: The prompt used to elicit structured plans.

G Few-Shot Ablation

The main experiments evaluate all models zero-shot. To test whether in-context demonstrations can teach models the structured plan syntax and thereby relieve the execution bottleneck, we ran an additional 2-shot ablation on four open-weight models. We prepended two demonstrations to the prompt, each consisting of a table, a question, a step-by-step reasoning trace (CoT) or execution trace (PoT), the final answer, and, in the **FS-Oracle** condition, the corresponding structured plan. The two demonstration questions are held out, so all few-shot numbers are computed over the remaining 369 questions; this smaller

Plan-Guided Chain-of-Thought
You are a helpful science assistant who answers questions about information in tables.
Here is the relevant tabular data:
[Table Data]
Here is a step-by-step plan to follow:
[Plan]
How to read the plan:
- SELECT: Look up and extract the specified values from the table
- LOOP for each X: Repeat the indented steps for every X
- COMPUTE: Perform the calculation described
- IF: Only do the indented steps when the condition is true
- RETURN: This is the final answer to provide
Follow this plan carefully step by step to answer the question. Your final response should be "Answer:" followed by the answer.
Question: [Question]
Plan-Guided Program-of-Thought
Write Python code to analyze the table and answer the question.
Table:
[Table Data]
Here is a step-by-step plan to implement:
[Plan]
How to read the plan:
[...Same reading guide as above...]
Question: [Question]
Write Python code that implements this plan and prints the final answer.

Table 10: Prompts for executing a provided reasoning plan (Self or Gold).

denominator accounts for the minor differences between the ZS-Direct column below and Table 2. Proprietary models are omitted because of API constraints on long prompts.

Model	Accuracy (%)			FS-Oracle Errors	
	ZS-Direct	FS-Direct	FS-Oracle	Total	Regressions
Qwen3-30B	44.4	47.1	47.7	193	52
Gemma-3-27B	30.1	28.7	32.5	249	57
Llama-3.3-70B	34.4	29.0	29.3	261	71
Qwen3-Coder-30B (PoT)	19.5	25.7	35.2	239	18

Table 11: **Two-shot ablation** ($N = 369$). **ZS-Direct**: zero-shot, no guidance. **FS-Direct**: two demonstrations. **FS-Oracle**: two demonstrations plus the ground-truth structured plan. **Regressions** counts questions answered correctly under ZS-Direct but failed under FS-Oracle.

Two observations carry over from the zero-shot setting. First, the ceiling remains low. Even with demonstrations *and* a correct plan, the strongest model reaches 47.7%, failing more than half the benchmark. Teaching the plan syntax explicitly does not convert correct strategies into correct answers, which is consistent with execution rather than comprehension being the limiting factor.

Second, the compliance cost persists and remains architecture-dependent. The code model gains the most from oracle plans (+15.7 points over ZS-Direct) and regresses on only 18

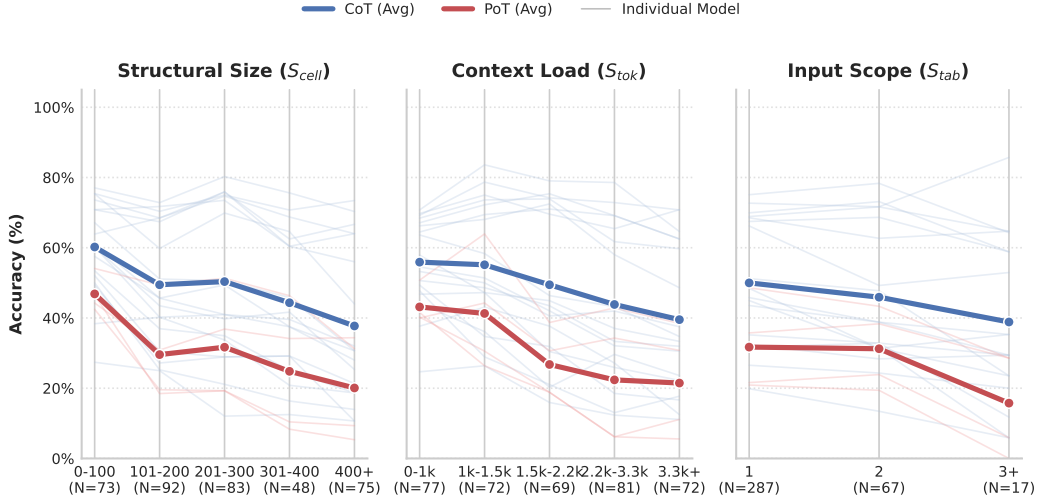


Figure 6: **Extended Complexity Analysis.** Performance breakdown across three dimensions of input scale: (Left) Structural Size (S_{cell}), (Middle) Context Load (S_{tok}), and (Right) Input Scope (S_{tab}). While increasing structural size and context load generally degrades performance for weaker models, multi-table reasoning shows divergent behaviors: top models often improve on tasks requiring cross-table linking. *Sample sizes (N) for each bin are denoted on the x-axis.*

questions. The generalist models gain little or lose ground, and 52–71 of their previously correct answers turn wrong when they are required to follow an external plan. Demonstrations do not eliminate this cost; the tension between a rigid algorithmic path and a generalist model’s intrinsic reasoning survives explicit instruction in the format.

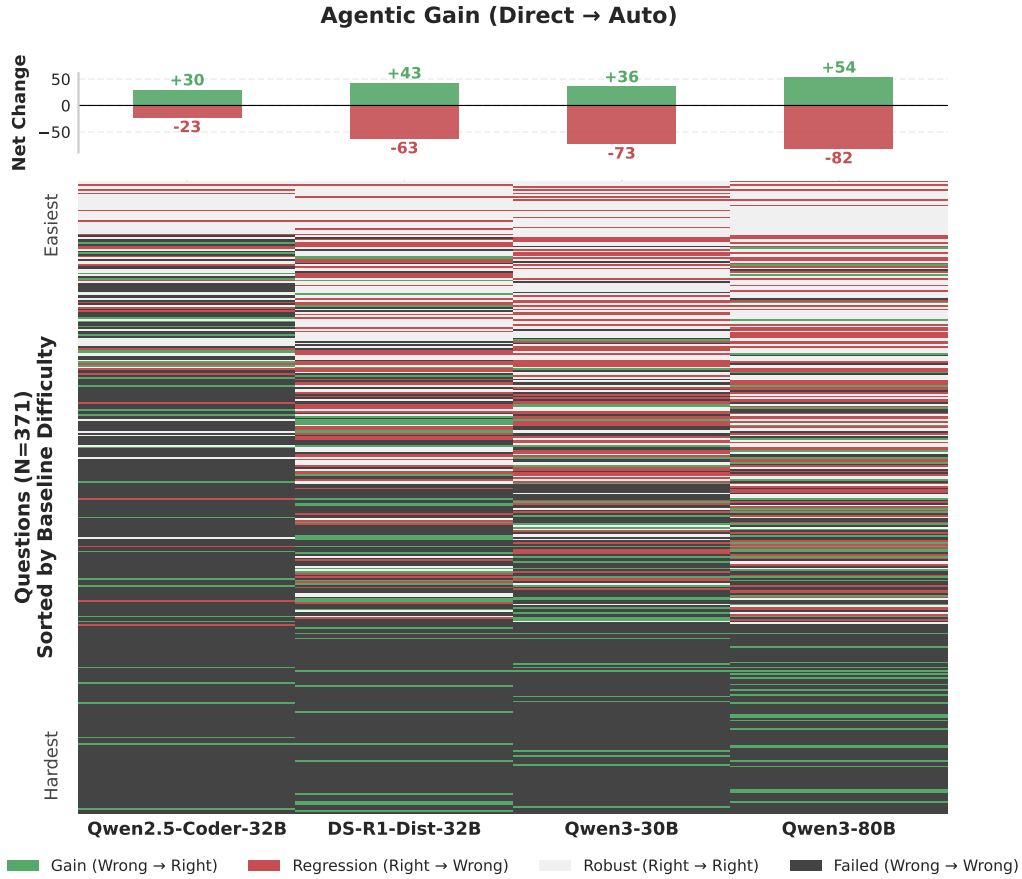


Figure 7: **Autonomous Gain (Direct → Autonomous)**. Consistent with the Gain Curve, planning exhibits a dual effect. For the Code model (left), we observe net positive gains (+7) concentrated on harder questions. For Language models (right), we observe net regression (e.g., -37 for Qwen3-30B), primarily driven by the "Cost of Compliance" on easier questions where the overhead of planning disrupts simple retrieval tasks.

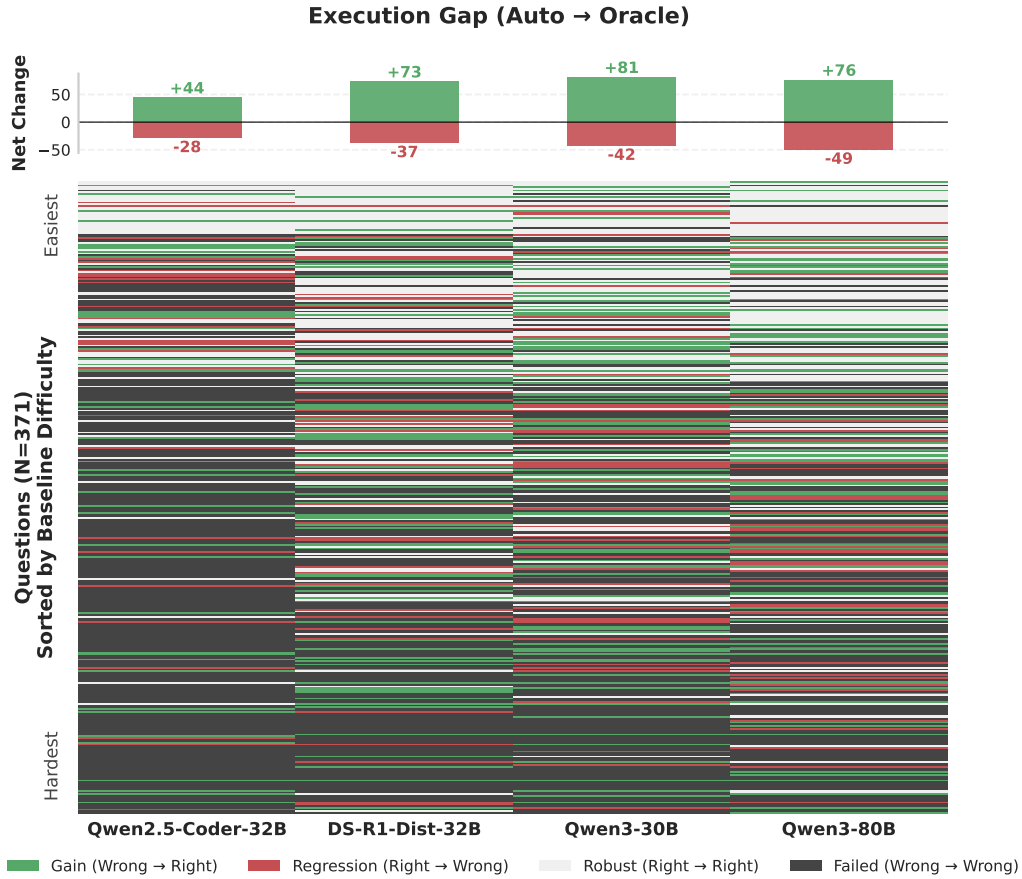


Figure 8: **Plan Correction (Autonomous → Oracle)**. This plot isolates the quality of the generated plan. While we observe net positive gains (+16 to +39) when moving to oracle plans, the non-trivial volume of **Regression (Red)** cases indicates that external strategies can sometimes disrupt the model’s intrinsic reasoning flow. The dominance of shared failures (gray pixels) between autonomous and oracle planning confirms that the primary bottleneck is execution fidelity, as even perfect strategies often fail to overcome implementation errors.

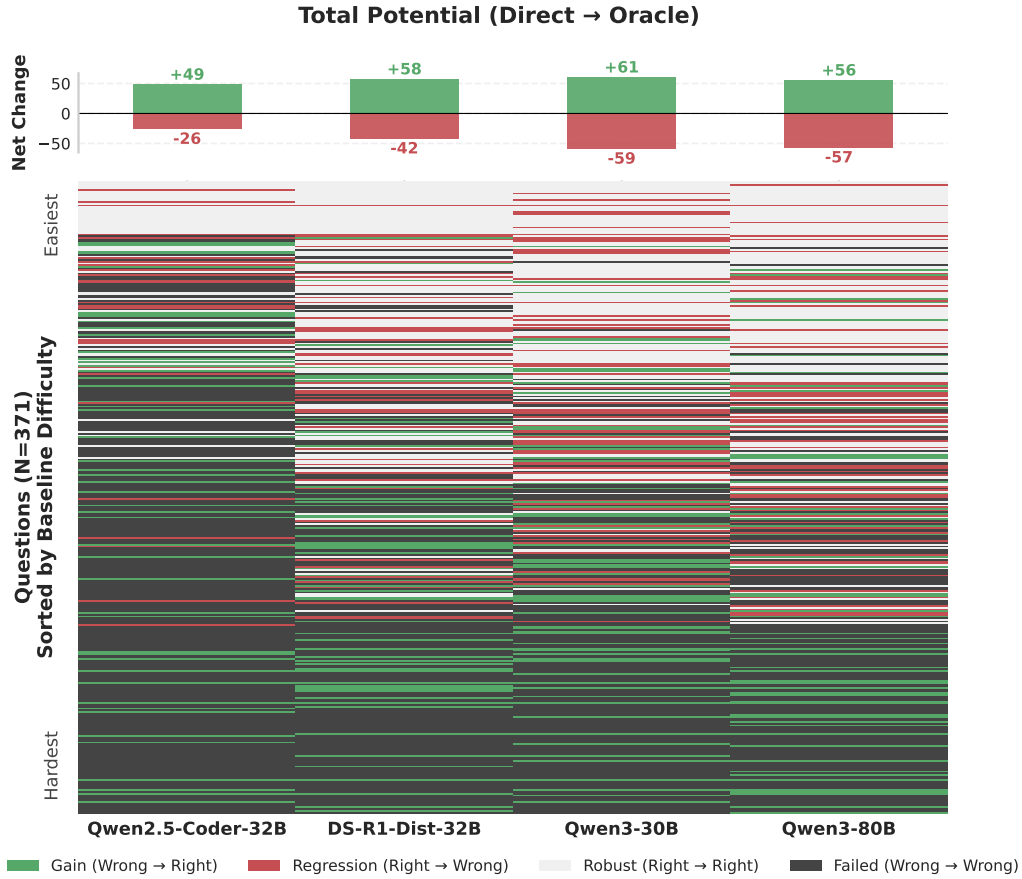


Figure 9: **Total Potential (Direct → Oracle)**. This compares the baseline directly against the upper bound. Even with perfect Oracle guidance, the net gains are moderate (−1 to +23). The vast majority of difficult questions remain **Failed (Gray)**, demonstrating that a significant portion of the dataset hits a hard capability ceiling that cannot be solved by planning assistance alone.