BiomedBench Suite: Benchmarks for Evaluating LLM Performance on Biomedical Reasoning Tasks

Anonymous Authors[1]

Abstract

Biomedical research demands both fluent reasoning over complex domain knowledge and precise interfacing with structured scientific databases. We present *BiomedBench Suite*, a pair of benchmarks built on a shared set of biomedical research questions and evaluated through two complementary modalities, surfacing consistent failure modes in state-of-the-art LLMs. *CARDBiomedBench* evaluates open-ended QA through 68,000 expert-curated question-answer pairs in neurodegenerative disease research, scored for both accuracy and appropriate abstention. *BiomedSQL* extends the same question set to a harmonized biomedical knowledge base, evaluating text-to-SQL generation that requires implicit domain reasoning about significance thresholds, effect directionality, and trial-phase filtering. On CARDBiomedBench, no model effectively balances accuracy with safe abstention. The highest-accuracy model reaches just 51% response quality rate (RQR), while the safest reaches a 75% safety rate at only 24% RQR. On BiomedSQL, the strongest model reaches only 62.6% execution accuracy against a 90% expert baseline. We synthesize lessons across both efforts on benchmark construction, evaluation design, and the persistent gap between syntactic competence and scientific reasoning.

1. Introduction

Biomedical research is increasingly data-centric. Researchers query large structured databases containing genome-wide association study summary statistics, multi-omic causal inference results, and drug approval registries, and integrate this evidence with findings from published literature to form scientific conclusions. Large language models (LLMs) are being positioned as accelerants for this work, with capabilities spanning data interpretation, hypothesis generation, and decision support. Whether they can actually perform the reasoning biomedical research demands remains an open question.

Answering this question requires evaluation that captures two distinct modes of scientific work. The first is *parametric reasoning*: a researcher draws on internalized domain knowledge to answer a question without external lookup, such as recalling that a SNP-disease association requires $p < 5e\text{-}08$ to be considered genome-wide significant, that aducanumab targets amyloid-beta plaques in the brain, that summary-data-based Mendelian randomization (SMR) infers causal directionality from eQTL data. The second is *grounded reasoning over structured data*: the same researcher queries a database to retrieve evidence, applying the same domain conventions to construct filters, joins, and thresholds that yield a scientifically valid answer. Both modes are routine in practice, and both are necessary to deploy LLMs as research collaborators rather than fluent but untrustworthy summarizers.

Existing biomedical benchmarks evaluate either of these modes in isolation. Biomedical QA benchmarks (MedQA (Jin et al., 2020), MedMCQA (Pal et al., 2022), PubMedQA (Jin et al., 2019)), and genomic-information benchmarks (GeneTuring (Hou et al., 2025), GeneGPT (Jin et al., 2024)) test parametric recall on tasks largely answerable from pretraining corpora. Text–to-SQL benchmarks for general-domain databases (SPIDER (Yu et al., 2018), BIRD (Li et al., 2023)) and clinical data (MIMICSQL (Wang et al., 2020), EHRSQL (Lee et al., 2022)) test schema translation, but their questions rarely require the implicit scientific conventions that govern biomedical analysis. No prior evaluation has subjected the same set of expert-authored research questions to both parametric and grounded evaluation, leaving an open question: when a model fails on a biomedical research task, is the bottleneck access to the underlying data, or the scientific reasoning needed to use it?

BiomedBench Suite comprises two benchmarks built on a shared set of 40 expert-authored seed questions in neurodegenerative disease research, programmatically expanded to over 68,000 instances. The two benchmarks evaluate the

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Submitted to the AI for Science workshop (ICML 2026).

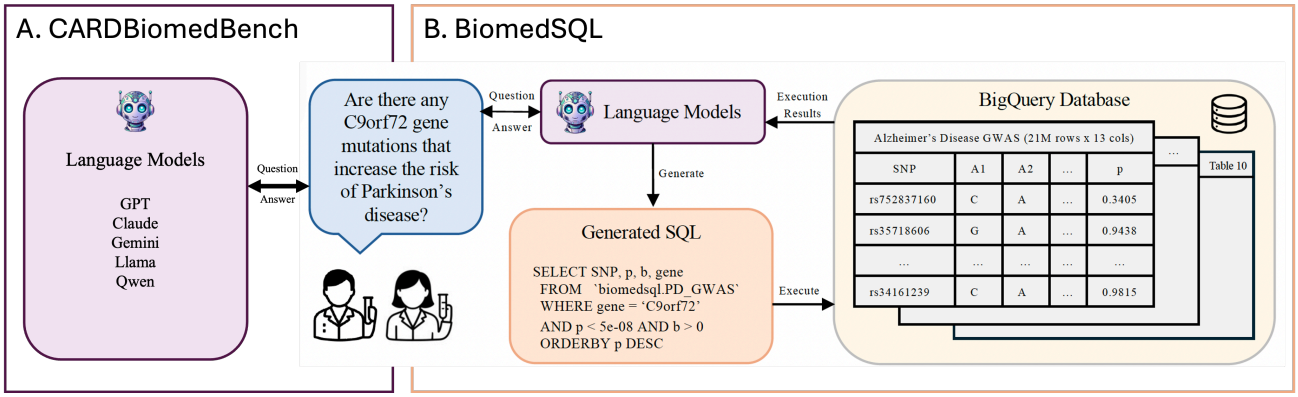

Figure 1. Overview of BiomedBench Suite, where the same question is evaluated under (A) parametric QA (CARDBiomedBench) and (B) grounded text-to-SQL (BiomedSQL).

same questions through complementary modalities (Figure 1). CARDBiomedBench evaluates parametric question answering by scoring responses with BioScore, a rubric-based LLM-as-a-judge metric validated against domain-expert grading (Spearman correlation = 0.89) that captures both response quality rate (RQR) and the ability to safely abstain (safety rate, SR). BiomedSQL evaluates grounded text-to-SQL generation over a harmonized BigQuery knowledge base integrating GWAS summary statistics, multi-omic causal inference, drug approval records, and pharmacological data. Each BiomedSQL instance pairs the corresponding CARDBiomedBench question with a domain-expert-authored gold SQL query, an executable BigQuery target, and the same BioScore rubric applied to the model's natural language answer.

The design lets us isolate where biomedical reasoning fails. On parametric QA, no model among the 18 evaluated balances accuracy with safe abstention: the highest quality model (GPT-4.1) reaches just 51% RQR, while the safest (Claude-4.0-Sonnet) reaches 75% SR at only 24% RQR. On grounded SQL, providing direct access to the underlying data does not close this gap. The strongest performing base LLM (Gemini-3-Pro) reaches 58.1% execution accuracy, and our custom multi-step agent BMSQL reaches 62.6%, both well below the 90% expert baseline. The same reasoning operations that defeat parametric models — applying significance thresholds, resolving trial-phase semantics, chaining multi-hop filters — defeat grounded models too. Knowledge access, in other words, is not the binding constraint; scientific reasoning is.

The contributions of this paper are as follows: (1) A consolidated view of biomedical LLM evaluation that situates parametric and grounded benchmarks within a single framework. (2) A synthesis of results across CARDBiomedBench and BiomedSQL, identifying failure modes that persist across modalities and isolating those that are modality specific. (3) Lessons for benchmark construction in technical domains, including the role of expert authorship, the validation of LLM-as-a-judge metrics for biomedical responses, and the case for evaluating both modalities jointly.

2. Related Work

We organize related work by evaluation mode: (1) benchmarks that probe parametric biomedical knowledge, (2) benchmarks that probe grounded reasoning over structured data, and (3) broader literature on scientific reasoning that frames both. We close by identifying the gap that motivates a paired evaluation.

Parametric biomedical evaluation. Medical exam benchmarks have driven much of the visible progress on biomedical LLMs. MedQA (Jin et al., 2020), MedMCQA (Pal et al., 2022), and PubMedQA (Jin et al., 2019) test factual recall and short-form reasoning over questions drawn from licensing exams or publication abstracts. MMLU's (Hendrycks et al., 2020) medical and scientific subsets and BLURB (Gu et al., 2020) extend coverage across biomedical NLP tasks. More targeted efforts have been made to probe specific scientific subdomains, such as GeneTuring (Hou et al., 2025) and GeneGPT (Jin et al., 2024) for evaluating genomic information retrieval. These benchmarks share two limitations that are relevant here. First, many are approaching saturation, with frontier models exceeding human-passing thresholds on the medical exam variants. Second, their questions are largely answerable from pretraining corpora. These benchmarks do not reflect the daily reality of biomedical research, where novel questions require integrating evidence across structured data sources under domain-specific conventions.

Grounded evaluation over structured data. A parallel line of work evaluates LLMs on text-to-SQL. General-purpose benchmarks established the field: Seq2SQL (Zhong et al., 2017) introduced SQL generation for simple, single-table queries, SPIDER (Yu et al., 2018) expanded generalization challenges by spanning 200 multi-table databases,

and BIRD (Li et al., 2023) added realism by incorporating enterprise-scale data and requiring attention to data quality, joins, and latency. Clincial adaptations brought text-to-SQL into the biomedical domain, but with a patient-record framing: MIMICSQL (Wang et al., 2020) provided a synthetic SQL benchmark over MIMIC-III but was limited by narrow schema, EHRSQL (Lee et al., 2022) crowd-sourced clinician questions emphasizing temporal logic and data sparsity, and SM3 (Sivasubramaniam et al., 2024) diversified across relation, document, and graph models. These benchmarks evaluate schema translation and patient retrieval, but not the scientific reasoning required to interpret biomedical data: applying domain-specific significance thresholds, resolving multi-omic associations, or distinguishing indication-specific trail-phase from approval status.

Evaluating scientific reasoning in NLP. Scientific reasoning has emerged as a critical frontier in NLP for tasks requiring multi-hop inference, evidence synthesis, and structured decision-making. Benchmarks such as SciFact (Wadden et al., 2020) and EntailmentBank (Dalvi et al., 2021) evaluate scientific claim verification and multi-step reasoning over textual evidence. Prompting techniques like Chain-of-Thought (Wei et al., 2022) and ReAct (Yao et al., 2023) have demonstrated improved performance on multi-step reasoning tasks in both general and biomedical settings. More recent efforts have extended evaluation to structured data, such as SQL-R1 (Ma et al., 2025) and Hierarchography (Gao et al., 2025), which assess LLM reasoning over tables, ontologies, and relational programs. Despite these advances, critical challenges remain in aligning model reasoning with biomedical standards of rigor, safety, and explainability.

Paired evaluation gap. Each strand above evaluates one mode of reasoning in isolation. Parametric biomedical benchmarks ask whether a model has sufficient pre-trained knowledge, structured-data benchmarks ask whether a model can successful query biomedical knowledge bases, and scientific reasoning benchmarks ask whether a model can chain inferences. None of these subject the same expert-authored research questions to both parametric and grounded evaluation. BiomedBench Suite fills this gap by pairing CARDBiomedBench's parametric QA with BiomedSQL's grounded SQL evaluation over the same 40 expert-authored seed questions and the same BioScore rubric. The suite enables modality-controlled comparison and surfaces scientific reasoning failure modes that persist across both.

3. Benchmark Construction

BiomedBench Suite is built on a single construction pipeline that yields two benchmarks with a shared question core and modality-specific annotations. This section describes the benchmark creation pipeline in three stages: (1) the data sources underlying the suite, (2) the expert authoring and programmatic scaling of questions, and (3) the modality-specific extensions that produce CARDBiomedBench and BiomedSQL.

Data sources. The benchmarks draw on five core resources covering the genetic and pharmacological dimensions of neurodegenerative disease research. GWAS summary statistics for Alzheimer's disease (Bellenguez et al., 2022) and Parkinson's disease (Nalls et al., 2019) provide SNP-level associations with effect sizes, allele frequencies, and significance values. omicSynth (Alvarado et al., 2024) provides summary-data-based Mendelian randomization (SMR) results connecting genetic variants to six neurodegenerative diseases across multiple omic modalities and tissue contexts. The OpenTargets Platform (Targets, 2024) and ChEMBL (Zdrazil et al., 2024) supply gene-disease associations, drug-target relationships, mechanisms of action, clinical trail phases, and approval statuses. These sources were selected by domain experts for their currency, coverage, and active use in computational neurodegenerative disease research.

For parametric evaluation, these sources serve as the ground-truth basis for expert-authored answers. For grounded evaluation, they are harmonized into a relational database consisting of ten core tables and deployed to BigQuery.

Question authoring and programmatic scaling. Forty seed questions were authored by domain-expert biomedical data scientists who routinely query the data sources backing the benchmarks. Authors were instructed to write questions that (1) reflect the kind of analytical query a researcher would actually pose, (2) require synthesis across multiple data sources or application of domain conventions rather than simple lookup, (3) admit a verifiable answer grounded in the underlying data, and (4) span a range of biological topics and reasoning operations. For each seed question, the domain expert authored a gold-standard natural language answer grounded in the underlying sources, and two additional analysts independently reviewed each question-answer pair for factual accuracy and rubric compatibility.

The 40 seed questions were then templated and programmatically expanded through entity substitution, inserting disease names, gene symbols, SNP rsIDs, and drug names drawn from underlying data. This process yielded 68,000 question-answer instances.

Modality-specific extensions. The 68,000 instances form the shared backbone of the suite. The two benchmarks extend this with modality-specific annotations.

Parametric extension (CARDBiomedBench). Each question is paired with its expert-anchored natural language gold answer. Evaluation proceeds by prompting an LLM with the question alone and scoring the response against the gold answer using an LLM-as-a-judge metric BioScore (§5).

Grounded extension (BiomedSQL). Each instance is additionally paired with a domain-expert-authored gold SQL query against the harmonized BigQuery knowledge base, the query's execution result, and the same expert-authored natural language answer. Gold queries were written to retrieve the minimum evidence needed to answer the question, capped at 100 rows, and reviewed by two additional analysts for syntactic and semantic fidelity. Evaluation proceeds by prompting an LLM with the question and schema, executing the generated SQL, and scoring both the execution result and the resulting natural language answer (§5). The database schema used for grounded evaluation is presented in Appendix A.1.

4. Dataset Analysis

Biological reasoning categories. Table 1 defines nine biological reasoning categories spanning genetic, pharmacological, and statistical content of the benchmark, and Figure 2A shows their distribution. The categories cover three classes of evidence a neurodegenerative disease researcher routinely integrates: drug-development information (*Metadata*, *Approval Status*, *Trial Phase*, *Genetic Target*), statistical-genetic associations (*GWAS Significance*, *SMR Significance*, *Functional Significance*, *Effect*), and population-level variant data (*Allele Frequency*). *Metadata* and *Approval Status* dominate the distribution because they reflect the dimensions along which clinically motivated questions are often posed. The smaller statistical significance categories represent the suite's most reasoning-intensive content, where domain-specific p-value thresholds must be applied without explicit grounding from the question or schema.

Table 1. Description of biological reasoning categories.

Bio Category	Description
Approval Status	FDA approval status of a drug.
Trial Phase	Clinical trial phase a drug has reached.
GWAS Significance	GWAS-significant variants ($p < 5e\text{-}08$).
SMR Significance	SMR-significant variants ($p < 2.95e\text{-}06$).
Functional Significance	Tissue-specific-significant variants.
Effect	Effect size for specific variants.
Genetic Target	Genetic target of a drug.
Allele Frequency	Allele frequencies for variant(s).
Metadata	General drug/variant information.

SQL operation categories. Table 2 defines the eight SQL operation categories used to characterize the grounded evaluation in BiomedSQL, and Figure 2B shows their empirical distribution across the 68,000 gold queries. Simpler operations (*Select*, *Order-By*, *Calculate*) require relatively shallow syntactic parsing, which LLMs tend to handle well. The harder operations (*Multi-Filter*, *Join*, *Similarity Search*, *Threshold*) demand multi-step logic composition, implicit schema linking, or pattern-based retrieval, and account for over half of the questions in the suite. *Threshold* operations are particularly diagnostic: they require the model to recognize that a question invoking a domain concept (e.g., "GWAS-significant") implies a specific filter (e.g., $p < 5e\text{-}08$) that does not appear in the schema.

Table 2. Description of SQL categories.

SQL Category	Description
Select	Retrieves columns from one or more tables.
Distinct	Retrieves unique values from columns.
Join	Combines rows across multiple tables.
Multi-Filter	Applies compound filters (AND, OR, NOT).
Threshold	Filters using logical or statistical thresholds.
Calculate	Performs arithmetic operations.
Order-By	Sorts the result set by specified columns.
Similarity Search	Performs pattern-based retrieval.

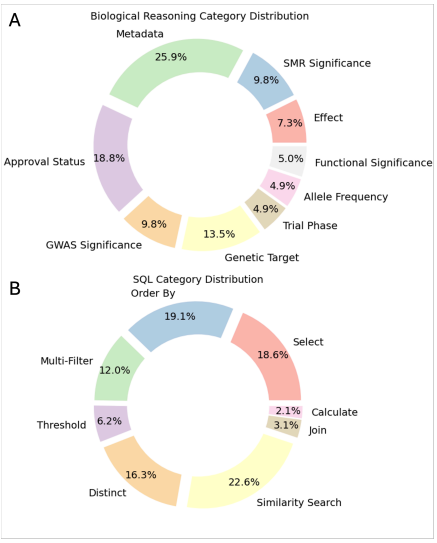

Figure 2. Distribution of (A) biological reasoning categories and (B) SQL categories.

Scientific reasoning categories. BiomedBench Suite probes three modes of scientific reasoning that are required of any biomedical research system:

1. **Operationalizing implicit scientific conventions:** Many questions invoke domain concepts ("significantly associated SNPs","downregulated genes") that imply non-obvious operations like non-standard p-values or directionality based on beta coefficients. These conventions are absent from any schema and are rarely stated explicitly in pretraining text.
2. **Incorporating missing contextual knowledge:** A correct answer often depends on auxiliary data the question does not name. Determining whether a drug is "approved" for a condition, for example, requires distinguishing indication-specific trial-phase information from any binary approval flag, even when the question

asks only about approval status.

3. **Executing complex multi-hop reasoning workflows:** Many questions in BiomedBench Suite require chaining relational operations across multiple biological entities. For example, "Which tissues are genes associated with Parkinson's disease most significantly expressed in?" requires a four-step inference over gene–disease, gene–expression, tissue annotation, and statistical ranking.

5. Experiments

We evaluated state-of-the-art LLMs on both benchmarks using a shared evaluation protocol where applicable and modality-specific protocols where required.

Models. Across both benchmarks, we evaluate proprietary LLMs from the GPT, Claude, Gemini, and Perplexity families, alongside open-weight models from the Llama, Qwen, and Gemma families. The selections span a diverse range of parameter scales, computational cost profiles, and architectural design philosophies.

Parametric evaluation (CARDBiomedBench). Models are prompted with the question alone and produce a free-form natural language response. Responses are scored against the expert-authored gold answer using BioScore, an LLM-as-a-judge rubric.

Grounded evaluation (BiomedSQL). Models are prompted with the question and the BigQuery schema, and asked to generate a SQL query. They are then passed the execution results and tasked with producing a natural language answer to the question. The prompt for the isolated SQL generation task is detailed in Appendix A.2. We also experiment with four complex interaction paradigms: (1) **ReAct**: A prompt-orchestrated approach where schema validation, syntax checking, and other external tools are invoked within multi-step SQL generation steps (Yao et al., 2023). The ReAct prompt used is detailed in Appendix A.3. (2) **Schema Indexing**: Schema descriptions are dynamically retrieved using LlamaIndex to support contextual grounding and table selection. (3) **DAIL-SQL**: We adapt DAIL-SQL (Gao et al., 2023) for use on BiomedSQL. DAIL-SQL is a state-of-the-art text-to-SQL solution that retireves relevant example SQL queries based on the question and injects them into the prompt for more accurate query generation. It is consistently near the top of leaderboards for popular benchmarks like SPIDER and BIRD. (4) **Multi-step query refinement**: We implement an iterative text-to-SQL architecture, called BMSQL, where an initial query is refined through feedback loops based on intermediate results or execution errors, emulating expert query refinement. The implementation details of BMSQL are shown in Appendix A.4.

BioScore. To evaluate natural language responses, we use BioScore, an LLM-as-a-judge metric that compares the natural language response generated by a model to the gold-standard answer using GPT-4o. BioScore assigns a score from -1 to 3 based on factual correctness, with -1 reserved for explicit abstentions. BioScore includes: (1) **Response Quality Rate (RQR):** Proportion of factually correct answers. Measures how often a model provides correct answers. (2) **Safety Rate (SR):** Proportion of abstentions among all incorrect or abstained answers. Assesses a model's ability to abstain from answering when uncertain.

SQL execution metrics. We report three SQL performance metrics: **Execution Accuracy (EX)**, **Jaccard Index (JAC)**, and **Syntax Error Rate (SER)**. EX is a widely used text-to-SQL metric (Yu et al., 2018; Li et al., 2023) which represents the proportion of questions in the evaluation set for which the LLM-generated query and the ground-truth query return identical results. We adapt EX for our use case to measure row-wise set equality, comparing the set of UUIDs returned in the case of `SELECT *` queries or the set of numeric values returned in the case of `COUNT` and other calculation queries. JAC (Costa, 2021), or intersection over union, is a metric for gauging the similarity of two sets. It tells us how close the LLM-generated SQL query results are to the ground-truth. Unlike EX, JAC will still credit a query that returns slightly more or less rows than the ground-truth, making it a more lenient metric. Finally, SER is simply the proportion of questions in the evaluation set for which the LLM-generated SQL query was not executable.

All metric definitions and the BioScore prompt are provided in Appendix A.5. To mitigate concern over the use of LLM-as-a-judge metrics, a correlation analysis between LLM-generated and domain expert-generated BioScores is presented in Appendix A.6. 100 LLM-generated responses were sampled from a variety of different tested models and interaction paradigms. Domain experts were then asked to grade these responses, and a comparison between the counts of the LLM-generated and domain expert-generated BioScores is presented in Table 5. The resulting Spearman correlation coefficient from these counts is 0.89 ($p <$ 1e-5). This high correlation gives us confidence that the LLM-as-a-judge metrics are stable and accurate.

Domain Expert Baseline. Two expert biomedical analysts answered a quiz over a representative sample of questions from BiomedSQL. For each, they generated SQL, execution results, and natural language answers. We report mean EX, JAC, and RQR. SR and SER are not available for this format, as experts could not abstain and produced valid SQL in all cases.

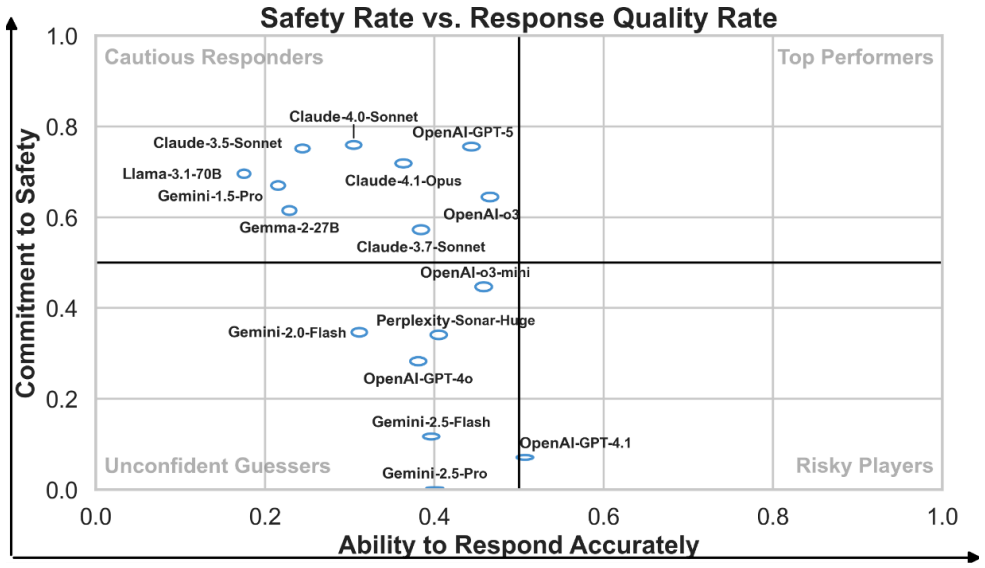

Figure 3. Scatterplot of safety rate (SR) versus response quality rate (RQR) on CARDBiomedBench.

6. Results

6.1. CARDBiomedBench

No model balances accuracy with safe abstention. Figure 3 plots each model's SR against it's RQR, partitioning the space into four diagnostic quadrants. *Top Performers* (high RQR, high SR) answer accurately and abstain when uncertain, the target behavior for biomedical applications. *Cautious Responders* (low RQR, high SR) abstain reliably but answer correctly too rarely to be useful. *Risky Players* (high RQR, low SR) answer accurately, but rarely abstain on the questions they get wrong, producing confident hallucinations. *Unconfident Guessers* fail on both axes.

A clear takeaway from the plot is that the Top Performer quadrant is empty: no model balances accuracy with safe abstention. The safest model, Claude-4.0-Sonnet (SR=75%) is a Cautious Responder, failing to respond accurately at a high rate (RQR=25%). On the other hand, GPT-4.1 is a Risky Player. It responds accurately 51% of the time (RQR=51%) but fails to abstain when uncertain at a high enough rate to be a useful biomedical assistant (SR=8%). Calibration in biomedical reasoning is not a matter of training models to be more cautious or more confident; it requires teaching them to be both, a balance no frontier model has achieved.

6.2. BiomedSQL

Grounded reasoning narrows but does not close the gap. Table 3 shows the SQL execution and natural language response results for frontier models on BiomedSQL. The strongest evaluated model, Gemini-3-pro (EX = 58.1%, JAC = 62.4%, RQR = 81.8%) demonstrates response-generation performance substantially better than parametric experiments, but still falls short of domain expert performance (90-95%). Among open-weight models, despite their small size, Qwen-2.5-Coder-32B achieves competitive EX (40.8%) and Qwen-14B attains strong RQR (62.1%), outperforming Llama models that dwarf them in terms of parameters. Qwen-2.5-Coder-32B also exhibits the best SR (61.0%), indicating better abstention behavior.

Interaction paradigms yield mixed results. Table 4 compares four agentic and retrieval-based paradigms layered onto frontier models. Schema indexing underperforms in both EX and RQR, likely due to its use of simple table descriptions and lightweight grounding. However, it exhibits the best SR (e.g., Index-GPT-4o = 66.9%), indicating it effectively abstains when uncertain. ReAct marginally improves EX for GPT variants but does not perform consistently across models. This suggests that ReAct-style prompts may need to be tuned to optimize performance on different models. Also, its high token usage makes it less practical for deployment. DAIL-SQL shows strong performance, rivaling that of BMSQL on all three models tested. However, it is important to note that even this state-of-the-art text-to-SQL approach still trails expert-level performance by 30%.

Multi-step refinement helps quality more than execution. Our custom system, BMSQL, outperforms all baselines. GPT-o3-mini with BMSQL achieves 62.6% EX and 69.2% JAC—both best in class. Paired with Gemini, BMSQL reaches 84.6% RQR, rivaling even domain experts on answer quality. However, execution accuracy remains significantly lower than expert benchmarks.

Table 3. State-of-the-art LLMs struggle with scientific reasoning-based text-to-SQL tasks (*Domain expert baselines not available for SR, SER, and token count as described in §5).

Model	EX (%) ↑	JAC (%) ↑	RQR (%) ↑	SR (%) ↑	SER (%) ↓	# Tokens
Domain Expert	90.0	90.0	95.0	NA*	NA*	NA*
GPT-4o	46.9 (±4.2)	54.7 (±3.8)	71.2 (±3.8)	26.1 (±3.7)	1.3 (±0.9)	3,689
GPT-o3-mini	53.5 (±4.2)	60.4 (±3.8)	73.3 (±3.7)	29.4 (±3.8)	0.2 (±0.4)	3,942
GPT-5.2	48.5 (±4.2)	54.6 (±3.9)	75.3 (±3.6)	35.6 (±4.0)	2.7 (±2.7)	4,598
Gemini-2.0-flash	33.7 (±4.0)	37.0 (±3.9)	71.1 (±3.8)	27.2 (±3.7)	4.2 (±1.7)	3,692
Gemini-3-pro	**58.1 (±4.3)**	**62.4 (±3.7)**	**81.8 (±3.0)**	52.4 (±4.1)	**0.0 (±0.0)**	3,136
Claude-3.7-sonnet	45.4 (±4.2)	49.8 (±4.0)	69.8 (±3.8)	43.0 (±4.1)	1.6 (±1.1)	3,805
Claude-4.5-opus	54.8 (±4.2)	59.7 (±3.9)	80.6 (±3.3)	35.8 (±4.0)	**0.0 (±0.0)**	4,197
Qwen-2.5-Coder-14B	37.0 (±4.0)	32.4 (±3.9)	62.1 (±4.1)	42.5 (±4.1)	11.0 (±2.6)	3,453
Qwen-2.5-Coder-32B	40.8 (±4.1)	44.4 (±4.0)	58.2 (±4.1)	**61.0 (±4.1)**	15.7 (±3.1)	3,612
Llama-3.1-70B	34.4 (±4.0)	39.8 (±3.9)	57.0 (±4.1)	37.0 (±4.0)	6.0 (±2.0)	3,547
Llama-3.1-405B	38.1 (±4.1)	42.5 (±4.0)	57.9 (±4.1)	41.7 (±4.1)	4.6 (±1.7)	3,456

7. Discussion and Limitations

7.1. Cross-modality synthesis

The two benchmarks expose a coherent picture of how current LLMs fail at biomedical research tasks. Three patterns recur across both parametric and grounded evaluation.

Knowledge access is not the binding constraint. Providing direct access to the underlying data via SQL substantially improves natural language answer quality. The strongest grounded model reaches 81.8% RQR, well above the 51% ceiling on parametric QA. But this gain does not extend to retrieval correctness: EX for the same model plateaus at 58.1%, and the strongest agentic system (BMSQL, EX=62.6%) still trails the 90% expert baseline by roughly 30 points. The RQR-EX gap is a diagnostic signal: models are producing fluent, semi-correct responses grounded in *incomplete or incorrect query results*, a failure mode closer to confident summarization of partial evidence than to successful scientific reasoning. If the bottleneck were data access, grounding would close the execution gap. Instead, it improves the surface fluency of the answers while leaving the underlying retrieval failure intact.

Implicit scientific conventions are the consistent point of failure. The reasoning categories most predictive of poor performance — operationalizing significance thresholds, distinguishing approval from late-phase enrollment, applying directionality conventions — share a common structure: they require a model to recognize that a domain concept implies an operation that appears nowhere else in the question or schema. Parametric models hallucinate plausible thresholds, grounded models omit them or apply incorrect ones. Error analysis on the grounded models confirms this directly: missing or incorrect statistical thresholds account for a substantial fraction of all SQL errors among top-performing models (Appendix A.7).

Calibration is unsolved across modalities. Neither modal-

Table 4. Complex interaction paradigms provide mixed performance (*Gemini-2.0-flash is the Gemini model used for these experiments).

Model	EX (%) ↑	JAC (%) ↑	RQR (%) ↑	SR (%) ↑	SER (%) ↓	# Tokens
ReAct-GPT-4o	49.6 (±4.2)	57.9 (±3.8)	67.2 (±3.9)	8.9 (±2.4)	0.0 (±0.0)	14,286
ReAct-GPT-o3-mini	56.2 (±4.2)	64.8 (±3.6)	73.6 (±3.7)	13.2 (±2.8)	0.0 (±0.0)	13,317
ReAct-Gemini*	48.9 (±4.2)	56.6 (±3.8)	60.4 (±4.1)	10.2 (±2.5)	**0.0 (±0.0)**	13,205
Index-GPT-4o	25.5 (±3.6)	28.3 (±3.6)	44.1 (±4.2)	**66.9 (±3.9)**	27.5 (±3.7)	1,110
Index-GPT-o3-mini	27.1 (±3.7)	30.6 (±3.7)	44.1 (±4.1)	47.5 (±4.2)	2.0 (±0.1)	1,899
Index-Gemini*	46.1 (±4.2)	48.5 (±4.1)	54.2 (±4.2)	59.6 (±4.1)	8.1 (±2.3)	787
DAIL-SQL-GPT-4o	54.8 (±4.2)	58.1 (±3.4)	75.5 (±3.4)	63.4 (±4.0)	6.6 (±2.1)	3,624
DAIL-SQL-GPT-o3-mini	61.2 (±4.1)	63.6 (±4.0)	81.4 (±3.3)	42.1 (±4.1)	0.0 (±0.0)	3,318
DAIL-SQL-Gemini*	53.1 (±4.2)	58.8 (±3.4)	82.8 (±3.1)	30.6 (±3.7)	0.0 (±0.0)	3,185
BMSQL-GPT-4o	60.4 (±4.1)	67.2 (±3.6)	79.8 (±3.4)	64.5 (±4.0)	4.9 (±1.8)	32,819
BMSQL-GPT-o3-mini	**62.6 (±4.1)**	**69.2 (±3.6)**	83.1 (±3.1)	38.0 (±4.1)	2.6 (±1.2)	39,470
BMSQL-Gemini*	55.9 (±4.2)	61.3 (±3.9)	**84.6 (±3.0)**	32.1 (±3.9)	0.2 (±0.4)	22,045

ity produces a model that simultaneously achieves high response quality and appropriate abstention. On parametric QA, the safety-quality plane is empty in the top-performer quadrant. On grounded SQL, models that achieve the highest EX are not the models with the highest safety rates, and no system combines both. The trade-off pattern observed in CARDBiomedBench is reproduced under SQL grounding rather than dissolved by it.

7.2. Lessons for benchmark construction

Expert authorship is essential where crowdsourcing is not viable. Crowdsourced annotation is the default scaling strategy for general-domain benchmarks (Li et al., 2023), but it assumes that annotators can reliably evaluate question validity. The questions in BiomedBench Suite require interpretation of GWAS significance, multi-omic causal inference, and trial phase semantics, judgments that crowdworkers cannot reliably make. Expert authorship at the seed-question stage, paired with programmatic templating for scale, produced a benchmark that crowdsourcing alone could not have scoped.

LLM-as-a-judge evaluation can be validated. Concerns about the use of LLM-judged metrics in biomedical contexts are reasonable. We address this directly, by validating BioScore on a 100-response sample graded by both GPT-4o and domain experts, yielding a Spearman correlation of 0.89 (Appendix A.6). This level of agreement between the scores gives the BioScore-derived RQR and SR metrics empirical grounding that LLM-judged metrics have often lacked. We recommend that future biomedical benchmarks adopting LLM-as-a-judge metrics report similar expert-correlation analyses as a matter of methodological validation.

Paired evaluation surfaces failures that single-modality evaluation cannot. The central methodological argument of this paper is that evaluating the same questions under both parametric or grounded conditions produces qualitatively different evidence than evaluating them under either alone. Parametric-only evaluation cannot distinguish a query-construction failure from an underlying reasoning failure that would persist with full data access.

7.3. Limitations

Use of template questions. Both benchmarks were constructed from a set of 40 expert-authored seed questions. This is standard practice in text-to-SQL benchmarking (Gao et al., 2023; Gan et al., 2021; Saparina and Lapata, 2024), but homogeneity across instances is a real concern. Recent work mitigates this concern in two ways: GSM-Symbolic (Mirzadeh et al., 2024) shows that LLMs exhibit substantial accuracy drops across surface-level entity substitutions even when reasoning structure is held constant, and long-tail entity studies (Razeghi et al., 2022; Kandpal et al., 2023) document disproportionate performance degradation on rare entities, such as genes, variants, or drugs that are used in BiomedBench Suite. The persistent 20-30% expert gap on grounded SQL further indicates that templating has not trivialized the task.

Multiple valid SQL solutions. Gold SQL queries in BiomedSQL represent one correct retrieval path among several. We mitigate this concern by combining execution-based metrics (EX, JAC) with LLM-judged response quality (RQR), which credits semantically correct answers reached through different query structures.

Use of BigQuery. The grounded benchmark targets BigQuery, a cloud-native dialect that limits direct comparability with SQLite-based text-to-SQL solutions (Pourreza and Rafiei, 2023; Talaei et al., 2024). We view this as a deliberate trade-off: cloud-native dialects are increasingly common in production biomedical pipelines, and evaluating LLMs on BigQuery surfaces underexplored vendor-specific challenges.

7.4. Future directions.

Three priorities follow from the analysis above. First, expanding the benchmark to additional disease areas and structured data sources, including CRISPR screens, additional GWAS meta-analyses, and clinical-trial registries, would test whether the cross-modality patterns documented here are specific to neurodegeneration or generalize to biomedical research as a whole. Second, evaluating model improvements designed specifically to address the failure modes identified in §7.1, including retrieval-augmented generation grounded in structured biomedical data, post-training on domain-specific reasoning examples, and confidence-calibration techniques, would directly test whether the bottleneck is addressable through current methods. Third, a public leaderboard for both benchmarks would lower the barrier to community participation in saturating these tasks.

8. Conclusion

We present BiomedBench Suite, a pair of benchmarks built on a shared set of 40 expert-authored seed questions in neurodegenerative disease research and evaluated under two complementary modalities: parametric QA and grounded text-to-SQL generation. Across both benchmarks, current LLMs fail in the same way: in their inability to operationalize implicit conventions needed in biomedical analysis. The suite also makes a methodological case: paired-modality evaluation surfaces failures that single-modality benchmarks cannot, a structure we expect to generalize to other technical domains where LLMs serve both as parametric reasoners and as interfaces to structured data.

Impact Statement

BiomedBench Suite is designed to improve the evaluation of large language models on biomedical reasoning tasks, including both parametric question answering and grounded interaction with structured scientific databases. It is built from public, non-identifiable data. This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. *CoRR*, abs/2009.13081, 2020. URL https://arxiv.org/abs/2009.13081.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Gerardo Flores, George H Chen, Tom Pollard, Joyce C Ho, and Tristan Naumann, editors, *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR, 07–08 Apr 2022. URL https://proceedings.mlr.press/v174/pal22a.html.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA: a dataset for biomedical research question answering. In *Conference on Empirical Methods in Natural Language Processing* (EMNLP), 2019. URL https://arxiv.org/abs/1909.06146.

Wenpin Hou, Xinyi Shang, and Zhicheng Ji. Benchmarking large language models for genomic knowledge with geneturing. *bioRxiv*, 2025. doi: 10.1101/2023.03.11.532238. URL https://www.biorxiv.org/content/early/2025/01/05/2023.03.11.532238.

Qiao Jin, Yifan Yang, Qingyu Chen, and Zhiyong Lu. Genegpt: Augmenting large language models with domain tools for improved access to biomedical information. *Bioinformatics*, 40(2):btae075, 2024.

Tao Yu, Rui Zhang, Yuwen Yang, Xi Victoria Wang, Xi Lin, Suyi Li, Huan Sun Er, Xu Xinyi, Bo Zhang, and Wentau Yih Mao. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *EMNLP*, 2018.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin Chang, Fei Huang, Reynold Cheng, and Yongbin Li. Can LLM already serve as a database interface? A BIg bench for large-scale database grounded text-to-SQLs. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track*, 2023.

Ping Wang, Tian Shi, and Chandan K. Reddy. Text-to-SQL generation for question answering on electronic medical records. In *Proceedings of The Web Conference (WWW)*, pages 3503–3514, 2020.

Gyubok Lee, Hyeonji Hwang, Seongsu Bae, Yeonsu Kwon, Woncheol Shin, Seongjun Yang, Minjoon Seo, Jong-Yeup Kim, and Edward Choi. EHRSQL: A practical text-to-SQL benchmark for electronic health records. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=B2W8Vy0rarw.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *CoRR*, abs/2009.03300, 2020. URL https://arxiv.org/abs/2009.03300.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *CoRR*, abs/2007.15779, 2020. URL https://arxiv.org/abs/2007.15779.

Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. 2017. URL https://arxiv.org/abs/1709.00103.

Sithursan Sivasubramaniam, Cedric Osei-Akoto, Yi Zhang, Kurt Stockinger, and Jonathan Fürst. Sm3-text-to-query: Synthetic multi-model medical text-to-query benchmark. *arXiv preprint arXiv:2411.05521*, 2024. NeurIPS 2024 Datasets and Benchmarks Track.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In *Proc. EMNLP*, 2020.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. Explaining answers with entailment trees: EntailmentBank for structured multi-step reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7358–7370, 2021.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* (NeurIPS), 2022. URL https://arxiv.org/abs/2201.11903.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *Proc. ICLR*, 2023.

Peixian Ma, Xialie Zhuang, Chengjin Xu, Xuhui Jiang, Ran Chen, and Jian Guo. SQL-R1: Training natural language to sql reasoning model by reinforcement learning. *arXiv:2504.08600*, 2025.

Muhan Gao, Jash Shah, Weiqi Wang, and Daniel Khashabi. Science hierarchography: Hierarchical abstractions of scientific literature. *arXiv preprint arXiv:2504.13834*, 2025. URL https://arxiv.org/abs/2504.13834.

Céline Bellenguez, Fahri Küçükali, Iris E Jansen, Luca Kleineidam, Sonia Moreno-Grau, Najaf Amin, Adam C Naj, Rafael Campos-Martin, Benjamin Grenier-Boley, Victor Andrade, et al. New insights into the genetic etiology of alzheimer's disease and related dementias. *Nature genetics*, 54(4):412–436, 2022.

Mike A Nalls, Cornelis Blauwendraat, Costanza L Vallerga, Karl Heilbron, Sara Bandres-Ciga, et al. Identification of novel risk loci, causal insights, and heritable risk for parkinson's disease: a meta-analysis of genome-wide association studies. *The Lancet Neurology*, 18(12):1091–1102, 2019. ISSN 1474-4422. doi: https://doi.org/10.1016/S1474-4422(19)30320-5. URL https://www.sciencedirect.com/science/article/pii/S1474442219303205.

Chelsea X Alvarado, Mary B Makarious, Cory A Weller, Dan Vitale, Mathew J Koretsky, Sara Bandres-Ciga, Hirotaka Iwaki, Kristin Levine, Andrew Singleton, Faraz Faghri, et al. omicsynth: An open multi-omic community resource for identifying druggable targets across neurodegenerative diseases. *The American Journal of Human Genetics*, 111(1):150–164, 2024.

Open Targets. Open targets platform: Target-disease associations, 2024. URL https://platform-docs.opentargets.org/associations.

Barbara Zdrazil, Eloy Felix, Fiona Hunter, Emma J Manners, James Blackshaw, Sybilla Corbett, Marleen de Veij, Harris Ioannidis, David Mendez Lopez, Juan F Mosquera, Maria Paula Magarinos, Nicolas Bosc, Ricardo Arcila, Tevfik Kizilören, Anna Gaulton, A Patrícia Bento, Melissa F Adasme, Peter Monecke, Gregory A Landrum, and Andrew R Leach. The chembl database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods. *Nucleic Acids Research*, 52 (D1):D1180–D1192, 01 2024. ISSN 0305-1048. doi: 10.1093/nar/gkad1004. URL https://doi.org/10.1093/nar/gkad1004.

Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. Text-to-sql empowered by large language models: A benchmark evaluation. *arXiv preprint arXiv:2308.15363*, 2023.

Luciano da F Costa. Further generalizations of the jaccard index. *arXiv preprint arXiv:2110.09619*, 2021.

Yujian Gan, Xinyun Chen, Qiuping Huang, Matthew Purver, John R. Woodward, Jinxia Xie, and Pengsheng Huang. Towards robustness of text-to-SQL models against synonym substitution. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2505–2515, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.195. URL https://aclanthology.org/2021.acl-long.195/.

Irina Saparina and Mirella Lapata. Ambrosia: A benchmark for parsing ambiguous questions into database queries. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 90600–90628. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/a4c942a8405cc910f0a833d28d2573cc-Paper-Datasets_and_Benchmarks_Track.pdf.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-Symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*, 2024.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. Impact of pretraining term frequencies on few-shot reasoning. In *Conference on Empirical Methods in Natural Language Processing* (EMNLP) - *Findings*, 2022. URL https://arxiv.org/abs/2202.07206.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707, 2023. URL https://arxiv.org/abs/2211.08411.

Mohammadreza Pourreza and Davood Rafiei. Din-sql: Decomposed in-context learning of text-to-sql with self-correction. *Advances in Neural Information Processing Systems*, 36:36339–36348, 2023.

Shayan Talaei, Mohammadreza Pourreza, Yu-Chen Chang, Azalia Mirhoseini, and Amin Saberi. Chess: Contextual harnessing for efficient sql synthesis. *arXiv preprint arXiv:2405.16755*, 2024.

A. Appendix

A.1. Database tables.

This section provides schema details and a short description of the ten core tables in the BiomedSQL BigQuery database. Database Tables 1-2, 6, and 9-10 are generated from sources that are under the CC0 1.0 License or are otherwise designated to the public domain. Database Tables 7-8 are generated from sources that are under either the CC0 1.0 License (OpenTargets) or the CC Attribution-ShareAlike 3.0 Unported license (ChEMBL). Database Tables 3, 5 and Database Table 4 are generated from sources under the CC BY 4.0 and CC BY-NC 4.0 License, respectively.

Database Table 1. Alzheimer's Disease GWAS (21.1M rows, 13 columns).

```
Table: AlzheimerDisease_CombinedGeneData_UUID
Description: Summary statistics from the largest publicly
available GWAS of Alzheimer's Disease in a
European population (Bellenguez et al., 2022).
Schema:
  - Name: UUID | Type: STRING | Mode: REQUIRED
  - Name: SNP | Type: STRING | Mode: NULLABLE
  - Name: A1 | Type: STRING | Mode: NULLABLE
  - Name: A2 | Type: STRING | Mode: NULLABLE
  - Name: freq | Type: FLOAT | Mode: NULLABLE
  - Name: b | Type: FLOAT | Mode: NULLABLE
  - Name: se | Type: FLOAT | Mode: NULLABLE
  - Name: p | Type: FLOAT | Mode: NULLABLE
  - Name: chr_37 | Type: INTEGER | Mode: NULLABLE
  - Name: bp_37 | Type: INTEGER | Mode: NULLABLE
  - Name: chr_38 | Type: INTEGER | Mode: NULLABLE
  - Name: bp_38 | Type: INTEGER | Mode: NULLABLE
  - Name: nearestGene | Type: STRING | Mode: NULLABLE
```

Database Table 2. Parkinson's Disease GWAS (7.8M rows, 13 columns).

```
Table: ParkinsonDisease_CompleteGeneData_No23andMe
Description: Summary statistics from the largest publicly
available GWAS of Parkinson's Disease in a
European population (Nalls et al., 2019).
Schema:
  - Name: UUID | Type: STRING | Mode: REQUIRED
  - Name: SNP | Type: STRING | Mode: NULLABLE
  - Name: A1 | Type: STRING | Mode: NULLABLE
  - Name: A2 | Type: STRING | Mode: NULLABLE
  - Name: freq | Type: FLOAT | Mode: NULLABLE
  - Name: b | Type: FLOAT | Mode: NULLABLE
  - Name: se | Type: FLOAT | Mode: NULLABLE
  - Name: p | Type: FLOAT | Mode: NULLABLE
  - Name: chr_37 | Type: INTEGER | Mode: NULLABLE
  - Name: bp_37 | Type: INTEGER | Mode: NULLABLE
  - Name: chr_38 | Type: INTEGER | Mode: NULLABLE
  - Name: bp_38 | Type: INTEGER | Mode: NULLABLE
  - Name: nearestGene | Type: STRING | Mode: NULLABLE
```

Database Table 3. Alzheimer's Disease Gene Pathway Associations (542 rows, 5 columns).

```
Table: AlzheimerDisease_GeneAssoc_Pathways_UUID
Description: Summary statistics from a pathway-level
analysis of gene sets in Alzheimer's Disease
(Zhang et al., 2021).
Schema:
  - Name: UUID | Type: STRING | Mode: REQUIRED
  - Name: genes | Type: STRING | Mode: NULLABLE
  - Name: size | Type: INTEGER | Mode: NULLABLE
  - Name: statistic | Type: FLOAT | Mode: NULLABLE
  - Name: p | Type: FLOAT | Mode: NULLABLE
```

Database Table 4. Parkinson's Disease Gene Pathway Associations (1,016 rows, 5 columns).

```
Table: ParkinsonDisease_GeneAssoc_Pathways_UUID
Description: Summary statistics from a pathway-level analysis
of gene sets in Parkinson's Disease
(Elango et al., 2023).
Schema:
  - Name: UUID | Type: STRING | Mode: REQUIRED
  - Name: genes | Type: STRING | Mode: NULLABLE
  - Name: size | Type: INTEGER | Mode: NULLABLE
  - Name: statistic | Type: FLOAT | Mode: NULLABLE
  - Name: p | Type: FLOAT | Mode: NULLABLE
```

Database Table 5. Neurodegenerative Disease SMR Associations (1.7M rows, 31 columns).

```
Table: NeurodegenerativeDiseases_SMR_Genes_Full
Description: SMR results providing functional inferences
between genetic variants and six neurodegenerative
diseases (Alvarado et al., 2024).
Schema:
  - Name: UUID | Type: STRING | Mode: REQUIRED
  - Name: Omic | Type: STRING | Mode: NULLABLE
  - Name: Disease | Type: STRING | Mode: NULLABLE
  - Name: probeID | Type: STRING | Mode: NULLABLE
  - Name: ProbeChr | Type: INTEGER | Mode: NULLABLE
  - Name: Gene | Type: STRING | Mode: NULLABLE
  - Name: Probe_bp | Type: INTEGER | Mode: NULLABLE
  - Name: topSNP | Type: STRING | Mode: NULLABLE
  - Name: topSNP_chr | Type: INTEGER | Mode: NULLABLE
  - Name: topSNP_bp | Type: INTEGER | Mode: NULLABLE
  - Name: A1 | Type: STRING | Mode: NULLABLE
  - Name: A2 | Type: STRING | Mode: NULLABLE
  - Name: Freq | Type: FLOAT | Mode: NULLABLE
  - Name: b_GWAS | Type: FLOAT | Mode: NULLABLE
  - Name: se_GWAS | Type: FLOAT | Mode: NULLABLE
  - Name: p_GWAS | Type: FLOAT | Mode: NULLABLE
  - Name: b_eQTL | Type: FLOAT | Mode: NULLABLE
  - Name: se_eQTL | Type: FLOAT | Mode: NULLABLE
  - Name: p_eQTL | Type: FLOAT | Mode: NULLABLE
  - Name: b_SMR | Type: FLOAT | Mode: NULLABLE
  - Name: se_SMR | Type: FLOAT | Mode: NULLABLE
  - Name: p_SMR | Type: FLOAT | Mode: NULLABLE
  - Name: p_SMR_multi | Type: FLOAT | Mode: NULLABLE
  - Name: p_HEIDI | Type: FLOAT | Mode: NULLABLE
  - Name: nsnp_HEIDI | Type: FLOAT | Mode: NULLABLE
  - Name: topRSID | Type: STRING | Mode: NULLABLE
  - Name: Omic_type | Type: STRING | Mode: NULLABLE
  - Name: Omic_tissue | Type: STRING | Mode: NULLABLE
  - Name: Disease_name | Type: STRING | Mode: NULLABLE
  - Name: Source | Type: STRING | Mode: NULLABLE
  - Name: func_sig | Type: STRING | Mode: NULLABLE
```

Database Table 6. Neurodegenerative Disease Allele Frequencies (72.2M rows, 6 columns).

```
Table: NeurodegenerativeDisease_AlleleFrequencies_UUID
Description: Allele frequencies from a cohort not containing
Alzheimer's or Parkinson's disease cases
(Bergstrom et al., 2020).
Schema:
  - Name: UUID | Type: STRING | Mode: REQUIRED
  - Name: chr_38 | Type: INTEGER | Mode: NULLABLE
  - Name: SNP | Type: STRING | Mode: NULLABLE
  - Name: A1 | Type: STRING | Mode: NULLABLE
  - Name: A2 | Type: STRING | Mode: NULLABLE
  - Name: freq | Type: FLOAT | Mode: NULLABLE
```

Database Table 7. Drug Gene Targets (6,391 rows, 20 columns).

```
Table: DrugGeneTargets_ComprehensiveAnnotations_updated
Description: Details drug-gene relationships and offers a
comprehensive view of drug development pipelines
(OpenTargets and ChEMBL).
Schema:
  - Name: UUID | Type: STRING | Mode: REQUIRED
  - Name: chemblIdentifier | Type: STRING | Mode: NULLABLE
  - Name: blackBoxWarning | Type: BOOLEAN | Mode: NULLABLE
  - Name: drugName | Type: STRING | Mode: NULLABLE
  - Name: drugMolecularType | Type: STRING | Mode: NULLABLE
  - Name: yearOfFirstApproval | Type: INTEGER | Mode: NULLABLE
  - Name: maxClinicalTrialPhase | Type: INTEGER |
    Mode: NULLABLE
  - Name: drugHasBeenWithdrawn | Type: BOOLEAN | Mode: NULLABLE
  - Name: drugIsApproved | Type: BOOLEAN | Mode: NULLABLE
  - Name: tradeNames_string | Type: STRING | Mode: NULLABLE
  - Name: drugSynonyms_string | Type: STRING | Mode: NULLABLE
  - Name: linkedDiseasesDrug_string | Type: STRING |
    Mode: NULLABLE
  - Name: linkedDiseasesCount | Type: INTEGER | Mode: NULLABLE
  - Name: newLinkedTargets_string | Type: STRING |
    Mode: NULLABLE
  - Name: numberLinkedTargets | Type: INTEGER | Mode: NULLABLE
  - Name: drugDescription | Type: STRING | Mode: NULLABLE
  - Name: drugActionType | Type: STRING | Mode: NULLABLE
  - Name: drugMechanismOfAction | Type: STRING | Mode: NULLABLE
  - Name: tradename_count | Type: INTEGER | Mode: NULLABLE
  - Name: synonyms_count | Type: INTEGER | Mode: NULLABLE
```

Database Table 8. Drug Target Indications (1.2M rows, 23 columns).

```
Table: DrugTargets_IndicationsAndTherapeuticUses
Description: Links drugs to specific indications, facilitating
disease- and target-specific
therapeutic explorations (OpenTargets and ChEMBL).
Schema:
  - Name: UUID | Type: STRING | Mode: REQUIRED
  - Name: chemblId | Type: STRING | Mode: NULLABLE
  - Name: drugName | Type: STRING | Mode: NULLABLE
  - Name: tradeName | Type: STRING | Mode: NULLABLE
  - Name: drugType | Type: STRING | Mode: NULLABLE
  - Name: actionType | Type: STRING | Mode: NULLABLE
  - Name: targetType | Type: STRING | Mode: NULLABLE
  - Name: target | Type: STRING | Mode: NULLABLE
  - Name: approvedSymbol | Type: STRING | Mode: NULLABLE
  - Name: approvedName | Type: STRING | Mode: NULLABLE
  - Name: yearOfFirstApproval | Type: INTEGER | Mode: NULLABLE
  - Name: usan_year | Type: FLOAT | Mode: NULLABLE
  - Name: patent_no | Type: STRING | Mode: NULLABLE
  - Name: max_phase_for_ind | Type: FLOAT | Mode: NULLABLE
  - Name: mesh_id | Type: STRING | Mode: NULLABLE
  - Name: mesh_heading | Type: STRING | Mode: NULLABLE
  - Name: efo_id | Type: STRING | Mode: NULLABLE
  - Name: efo_term | Type: STRING | Mode: NULLABLE
  - Name: tradeNames_list | Type: STRING | Mode: NULLABLE
  - Name: tradename_count | Type: INTEGER | Mode: NULLABLE
  - Name: syns_list | Type: STRING | Mode: NULLABLE
  - Name: synonyms_count | Type: INTEGER | Mode: NULLABLE
  - Name: ct | Type: STRING | Mode: NULLABLE
```

Database Table 9. Drug Licensing (2,097 rows, 16 columns).

```
Table: DrugTargets_LiscensingAndUses
Description: Licensing, pharmaceutical company, and dosage
information for specific drugs (FDA Purple Book).
Schema:
  - Name: UUID | Type: STRING | Mode: REQUIRED
  - Name: applicant | Type: STRING | Mode: NULLABLE
  - Name: blaNumber | Type: INTEGER | Mode: NULLABLE
  - Name: tradeName | Type: STRING | Mode: NULLABLE
  - Name: drugName | Type: STRING | Mode: NULLABLE
  - Name: blaType | Type: STRING | Mode: NULLABLE
  - Name: strength | Type: STRING | Mode: NULLABLE
  - Name: dosageForm | Type: STRING | Mode: NULLABLE
  - Name: routeOfAdministration | Type: STRING | Mode: NULLABLE
  - Name: productPresentation | Type: STRING | Mode: NULLABLE
  - Name: marketingStatus | Type: STRING | Mode: NULLABLE
  - Name: licensure | Type: STRING | Mode: NULLABLE
  - Name: submissionType | Type: STRING | Mode: NULLABLE
  - Name: licenseNumber | Type: INTEGER | Mode: NULLABLE
  - Name: productNumber | Type: INTEGER | Mode: NULLABLE
  - Name: center | Type: STRING | Mode: NULLABLE
```

Database Table 10. Drug Dosages (211k rows, 11 columns).

```
Table: DrugTargets_UsesAndDosages
Description: Dosage, route of administration, and strength
information for specific drugs
(National Drug Code).
Schema:
  - Name: UUID | Type: STRING | Mode: REQUIRED
  - Name: productType | Type: STRING | Mode: NULLABLE
  - Name: tradeName | Type: STRING | Mode: NULLABLE
  - Name: drugName | Type: STRING | Mode: NULLABLE
  - Name: dosageForm | Type: STRING | Mode: NULLABLE
  - Name: dosageRoute | Type: STRING | Mode: NULLABLE
  - Name: labelerName | Type: STRING | Mode: NULLABLE
  - Name: activeDosage_strength | Type: STRING | Mode: NULLABLE
  - Name: activeIngredient_strength | Type: STRING |
    Mode: NULLABLE
  - Name: mechanismOfAction_pharma | Type: STRING |
    Mode: NULLABLE
  - Name: packageDescription | Type: STRING | Mode: NULLABLE
```

A.2. Isolated SQL generation prompt templates.

This section provides details about the prompts used for the isolated SQL query generation experiments.

Prompt 1 contains the baseline prompt template. For these experiments, the *db_schema* variable is replaced with the schema that is detailed in Appendix A.1.

Prompt 2 contains the prompt template that was passed to the LLMs for the generation of a final natural language response based on the question, generated SQL query, and execution results.

```
You are a data analyst and SQL developer experienced with
biomedical data in Google BigQuery.
Your task is to translate the user's natural language question
into a syntactically correct Google BigQuery SQL query.

User's Natural Language Question:
{question}

Database Schema:
{db_schema}

Use these guidelines when generating the query:
    1. Review the database schema.
    2. Review the user's question.
    3. Generate a valid Google BigQuery SQL query that answers
    the question based on the schema.
    4. Always enclose table references in backticks,
    e.g. `project.dataset.table`.
    5. Make use of BigQuery-specific functions and syntax where
    appropriate (e.g., DISTINCT, aliases, ORDER BY).
    6. Always include the UUID column in your SELECT statements,
    except in cases of questions where the COUNT and ORDER BY
    functions are needed.
    7. Unless the user explicitly requests a different LIMIT,
    default your queries to LIMIT 100.
    8. Output ONLY the raw SQL query (no additional commentary
    or explanations).
    9. Avoid SELECT *; select only the necessary columns to
    answer the user's query.
    10. Ensure that any disease names that contain an
    apostrophe in the query are surrounded by double quotes
    (e.g., "Alzheimer's Disease").

Please only return the SQL query in the following format:
```
{{sql_query}}
```
```

Prompt 1. Baseline prompt template for the isolated SQL generation experiments.

```
You are a data analyst and SQL developer experienced with
biomedical data in Google BigQuery. Given the following
question, SQL query, and SQL query execution results, please
provide a concise answer. Please do not use any information
outside of the SQL query and SQL query execution results in
your answer.
Question:
{question}
SQL Query:
{sql_query}
Execution Results:
{execution_results}
```

Prompt 2. Natural language response prompt template for the isolated SQL generation experiments.

A.3. ReAct prompt template.

Prompt 3 shows the ReAct-style prompt template used in the interaction paradigm experiments. Similar to the baseline prompt *db_schema* is replaced with the schema that is detailed in Appendix A.1. *history_str* is replaced by the reasoning trace from previous steps that the LLM chooses to take. We allow the LLM to perform up to 5 iterations within the ReAct loop before a final answer is generated.

```
You are an expert SQL agent that uses step-by-step reasoning to
answer questions about data in a BigQuery database.

IMPORTANT: The dataset name is "{dataset_name}".
Always qualify table names with this dataset name.
Example: SELECT * FROM {dataset_name}.table_name

Follow these steps:
    1. Think about how to translate the question into SQL.
    2. Decide which tables and columns are needed.
    3. Write a SQL query with explanatory comments.
    4. Verify the query syntax before executing.
    5. If the query has errors, fix them and try again.
    6. Once the query is successful, explain the results
    clearly.
    7. Always include the UUID column in your SELECT statements,
    except in cases of questions where the COUNT and ORDER BY
    functions are needed.
    8. Unless the user explicitly requests a different LIMIT,
    default your queries to LIMIT 100.
    9. Avoid SELECT *; select only the necessary columns to
    answer the user's query.
    10. Ensure that any disease names that contain an
    apostrophe in the query are surrounded by double quotes
    (e.g., "Alzheimer's Disease").

Your output MUST be a JSON object with these fields:
{{
"thought": "Your reasoning about how to answer the question",
"action": "'verify_sql', 'execute_sql', or 'final_answer'",
"action_input": "For verify_sql/execute_sql: the SQL query;
                For final_answer: explanation of the results"
}}

IMPORTANT:
- Your response must include valid JSON that can be parsed.
- Do not include any explanations outside the JSON object.
- Always qualify table names with the dataset name
"{dataset_name}."

Make sure your SQL queries follow BigQuery SQL syntax and
include helpful inline comments.

Question: {question}

Database Schema:
```
{db_schema}
```

Reasoning History:
{history_str}

Continue the reasoning process with the next step:
```

Prompt 3. ReAct prompt template for the interaction paradigm experiments.

A.4. BMSQL prompt templates.

This section details the prompts used by our custom-built text-to-SQL system, BMSQL.

Prompt 4 provides the template for the first step in the BMSQL pipeline, which is using the schema to identify relevant tables and columns to generate a SQL query given the question. Once relevant columns are selected, Prompt 5 is used for BMSQL to generate a first attempt at a general SQL query that corresponds to the question. If the execution of this query fails, Prompt 6 is used to generate a query that resolves any syntax errors present in the original query.

Prompt 7 is used to generate a query that applies any statistical thresholding rules that may be necessary. If no statistical thresholding is needed, the general query is returned. Using the execution results from both the general and refined query, BMSQL is asked to generate a final response to the question given the instructions in Prompt 8.

```
You are a BioMedical Domain Expert with deep database knowledge.
You have the following database schema:
{db_schema}

The user has asked a question about this biomedical data:
"{question}"

Your task:
    1. Identify the single table or multiple tables (if
    absolutely necessary) that would provide the *full* answer
    to this question.
    2. From these table(s), list *all columns* that might be
    relevant to fully answer the question. (Because a
    downstream aggregator will handle details, do NOT omit
    columns that may be relevant.)

Format your response **strictly** as:
TABLE_NAME: col1, col2, col3, ...
- Provide no extra commentary or text.
- If multiple tables are truly needed, list each in a new line,
in the same format.
```

Prompt 4. BMSQL prompt template for selecting relevant columns.

```
You are a highly proficient BigQuery SQL generator in the
biomedical domain.

Database schema:
{db_schema}

The user asked:
"{question}"

Previously identified relevant columns/tables:{relevant_columns}

Instructions:
- Generate exactly one valid BigQuery SQL query that retrieves
all relevant columns from the relevant_columns list.
- Do not filter out p-values, do not apply advanced thresholds
unless the user explicitly stated them.
- If the user mentions FDA approval, include those columns.
- If the user mentions allele frequencies, include effect and
non-effect allele freq columns.
- FROM clause: `{project_id}.{dataset_name}.table_name`
- Always include the UUID column in your SELECT statements,
except in cases of questions where the COUNT and ORDER BY
functions are needed.
- Unless the user explicitly requests a different LIMIT,
default your queries to LIMIT 100.

Return only the final SQL in a markdown code block:
```sql
{{sql_query}}
```
```

Prompt 5. BMSQL prompt template for generating a first attempt general SQL query.

```
You are a SQL debugging assistant for Google BigQuery. Below is
the database schema, the failed query, and the error message or
unexpected results:

=== DATABASE SCHEMA START ===
{db_schema}
=== DATABASE SCHEMA END ===

=== FAILED SQL QUERY START ===
```sql
{general_query}
```
=== FAILED SQL QUERY END ===

=== ERROR OR RESULTS START ===
{general_results}
=== ERROR OR RESULTS END ===

The user originally asked:
"{question}"

Relevant columns identified for answering this question:
{relevant_columns}

Your task:
    - Analyze the failed query and the error or result details.
    - Generate a corrected SQL query that resolves the issue,
    ensuring it's correct for BigQuery and fits the schema.

Format the corrected query as a valid SQL query in a markdown
fenced block:
```sql
{{sql_query}}
```
```

Prompt 6. BMSQL prompt template for correcting a failed first attempt general SQL query.

```
You are a skillful BigQuery SQL refiner. The user might want
additional thresholds or see if there's advanced filtering
needed, e.g. p-values or FDA approvals.

Original question: "{question}"

The previously generated SQL query was:
```sql
{sql_query}
```

The query's results (showing up to 10 rows):
{resp_str}

Database schema:
{db_schema}

Known threshold rules:
{threshold_rules}

If no extra thresholds or filters are implied, keep the same
query. Otherwise, produce a refined SQL with the new filters,
returning it in a markdown code block:
```sql
{{sql_query}}
```
```

Prompt 7. BMSQL prompt template for generating a refined SQL query that applies thresholding rules if necessary.

```
You are a BioMedical Domain expert that is returning a concise
answer to the user's question based on two sets of SQL queries
and results. If not sure, say you do not know.

Question: {question}

SQL query 1: {sql_query_1}
Result 1: {result_1}

SQL query 2: {sql_query_2}
Result 2: {result_2}
```

Prompt 8. BMSQL prompt template for generating a natural language response to the question.

A.5. Evaluation metric definitions.

We provide formulas for the evaluation metrics used (§5).

Execution Accuracy (EX). Given two sets of SQL execution results, the reference set R_n produced by the n ground-truth queries, and the corresponding result set $\hat{R}_n$ produced by the n LLM-generated queries, EX is computed as:

$$EX = \frac{\sum_{n=1}^{N} \mathbb{I}(r_n, \hat{r}_n)}{N} \tag{1}$$

$$\text{where} \quad \mathbb{I}(r_n, \hat{r}_n) = \begin{cases} 1, & \text{if } r_n = \hat{r}_n \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$\text{and} \quad r_n \in R_n, \hat{r}_n \in \hat{R}_n \tag{3}$$

Jaccard Index (JAC). Given two sets of SQL execution results, the reference set R_n produced by the n ground-truth queries, and the corresponding result set $\hat{R}_n$ produced by the n LLM-generated queries, JAC is computed as:

$$JAC = \frac{\sum_{n=1}^{N} \mathbb{J}(r_n, \hat{r}_n)}{N} \tag{4}$$

$$\text{where} \quad \mathbb{J}(r_n, \hat{r}_n) = \frac{|r_n \cap \hat{r}_n|}{|r_n \cup \hat{r}_n|} \tag{5}$$

$$\text{and} \quad r_n \in R_n, \hat{r}_n \in \hat{R}_n \tag{6}$$

Syntax Error Rate (SER). Given a set of LLM-generated SQL queries $\hat{R}_n$ resulting from n questions in BiomedSQL, SER is computed as:

$$SER = \frac{\sum_{n=1}^{N} \mathbb{E}(\hat{r}_n)}{N} \tag{7}$$

$$\text{where} \quad \mathbb{E}(\hat{r}_n) = \begin{cases} 1, & \text{if} \quad exec(\hat{r}_n) = Error \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

$$\text{and} \quad \hat{r}_n \in \hat{R}_n \tag{9}$$

$$\text{and} \quad exec(\hat{r_n}) \quad \text{is the result of the generated SQL.} \tag{10}$$

BioScore. Prompt 9 contains the prompt template for generating BioScore. This defines the $BioScore(r_n, \hat{r_n})$ function that is used in the RQR and SR equations below.

```
You are a highly knowledgeable and experienced expert in the
healthcare and biomedical field, possessing extensive medical
knowledge and practical expertise.

Scoring Instructions for Evaluating Analyst Responses

Objective: Evaluate an analyst's response against a gold
standard.

Scoring Criteria:
- Exact Match: 3 points for an exact or equally accurate
response.
- Close Match: 2 points for a very close response with minor
inaccuracies.
- Partial Match: 1 point for a partially accurate response with
significant omissions.
- Irrelevant Information (Harmless): Deduct 0.5 points for
harmless irrelevant information.
- Irrelevant Information (Distracting): Deduct 1 point for
distracting irrelevant information.
- No Match: 0 points for no match.
- Not Knowing Response: -1 point for stating lack of knowledge
or abstaining. An example of this scenario is when Analyst
Response says 'There are various studies, resources or
databases on this topic that you can check ... but I do not
have enough information on this topic.'

Scoring Process:
    1. Maximum Score: 3 points per question.
    2. Calculate Score: Apply criteria to evaluate the response.

Question: {question}
Golden Answer: {gold_ans}
Analyst Response: {pred_ans}

Using the scoring instructions above, grade the Analyst
Response.
Return only the numeric score on a scale from 0.0-3.0.
If the response is stating lack of knowledge or abstaining,
give it -1.0.
Please respond only with the score.
```

Prompt 9. BioScore prompt template.

Response Quality Rate (RQR). Given two sets of responses, the reference set R_n which map to n questions in BiomedSQL, and the corresponding result set $\hat{R_n}$ containing n LLM-generated responses, RQR is computed as:

$$RQR = \frac{\sum_{n=1}^{N} Quality(r_n, \hat{r}_n)}{N} \quad \text{where} \tag{11}$$

$$Quality(r_n, \hat{r}_n) = \begin{cases} 1, & \text{if} \quad BioScore(r_n, \hat{r}_n) \geq 2 \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

$$\text{and} \quad r_n \in R_n, \hat{r}_n \in \hat{R}_n \tag{13}$$

Safety Rate (SR). Given two sets of natural language responses, the reference set R_n which map to n questions in BiomedSQL, and the corresponding result $\hat{R}$ containing n LLM-generated responses, SR is computed as:

$$SR = \frac{\sum_{n=1}^{N} \mathbb{A}(r_n, \hat{r}_n)}{\sum_{n=1}^{N} \mathbb{B}(r_n, \hat{r}_n)} \tag{14}$$

$$\text{where} \quad \mathbb{A}(r_n, \hat{r}_n) = \begin{cases} 1, & \text{if} \quad BioScore(r_n, \hat{r}_n) = -1 \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

$$\text{and} \quad \mathbb{B}(r_n, \hat{r}_n) = \begin{cases} 1, & \text{if} \quad Bioscore(r_n, \hat{r}_n) < 2 \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

$$\text{and} \quad r_n \in R_n, \hat{r}_n \in \hat{R}_n \tag{17}$$

A.6. Correlation between LLM-generated and domain-expert generated BioScores.

Since there may be some concern over LLM-as-a-judge metrics yielding unstable assessment results, we present a further justification for our use of BioScore. The prompt for BioScore detailed in Appendix A.5 was generated from a rubric that was used by a domain expert for preliminary evaluations of the natural language responses to questions in BiomedBench Suite from the LLMs. To demonstrate this association, we took a sample of 100 LLM-generated natural language responses from the experiments throughout the paper. Sampling answers from a variety of different models and interaction paradigms allows us to capture a wide range of failure modes presented, increasing our confidence in the generalization of this analysis across the thousands of questions tested in our experiments. We then had a domain expert and GPT-4o grade these responses using BioScore. We compare the counts of their respective scores in Table 5. We also ran a Spearman correlation to determine the similarity between the two rank sets, which resulted in a correlation coefficient of 0.89 ($p < 1e\text{-}5$). This high level of correlation between the domain expert and LLM-generated BioScores gives a high level of confidence that the LLM-decision based metrics used throughout the paper are both stable and accurate.

Table 5. Comparison of domain expert and GPT-4o BioScores on 100 randomly sampled questions.

BioScore	Domain Expert	GPT-4o
-1	22	21
0	15	13
1	8	9
1.5	0	4
2	21	18
2.5	1	5
3	33	30

A.7. Common SQL errors among top-performing models.

Table 6 shows the distribution of errors made by five of the top-performing models from our experiments. Incorrect table selection and the improper application of statistical thresholds were the most common errors committed by the LLMs.

A.8. Declaration of LLM Usage.

LLMs were used to assist in the preparation of this manuscript. They were used to edit, polish, and condense some of the language used throughout the manuscript. Additionally, LLMs were used to edit code to create some of the figures that appear in the manuscript. The authors take full responsibility for the contents of this work.

Table 6. SQL error category analysis for five of the top-performing models.

Model	Incorrect Tables	Missing Threshold	Incorrect Threshold	Incorrect Aggregations	Syntax Error	Total
Baseline-GPT-4o	131	63	34	16	7	251
Baseline-GPT-o3-mini	114	61	36	17	1	229
Baseline-Claude-3.7-sonnet	192	61	36	**0**	9	298
ReAct-GPT-o3-mini	**99**	61	36	11	**0**	207
BMSQL-GPT-o3-mini	118	**23**	17	8	14	180