

# DualBench100: A Methodology for Evaluating Accounting Assistants

## Abstract

DualBench100 evaluates accounting assistants with 101 synthetic cases across eight accounting categories. Each case specifies a task, a reference answer, and a scoring rule. Four deterministic methods assess classification, structured accounting responses, numerical answers, and multiple-choice knowledge. This paper defines the case composition, evaluation procedure, performance measures, and statistical comparison method. Repeated trials measure score variation. Paired bootstrap intervals estimate differences between configurations on matched cases. Response quality, execution coverage, latency, and estimated cost provide separate measures of performance. The procedure requires documented inputs, evaluation conditions, and eligibility rules. Scores describe the tested configuration and case set. They do not establish general accounting correctness or successful completion of every financial operation. This paper presents methods and dataset characteristics; it reports no new model performance results.

## Process at a glance

The procedure uses fixed evaluation inputs and scoring rules. Each comparison identifies the tested conditions and retains the observations used for analysis.

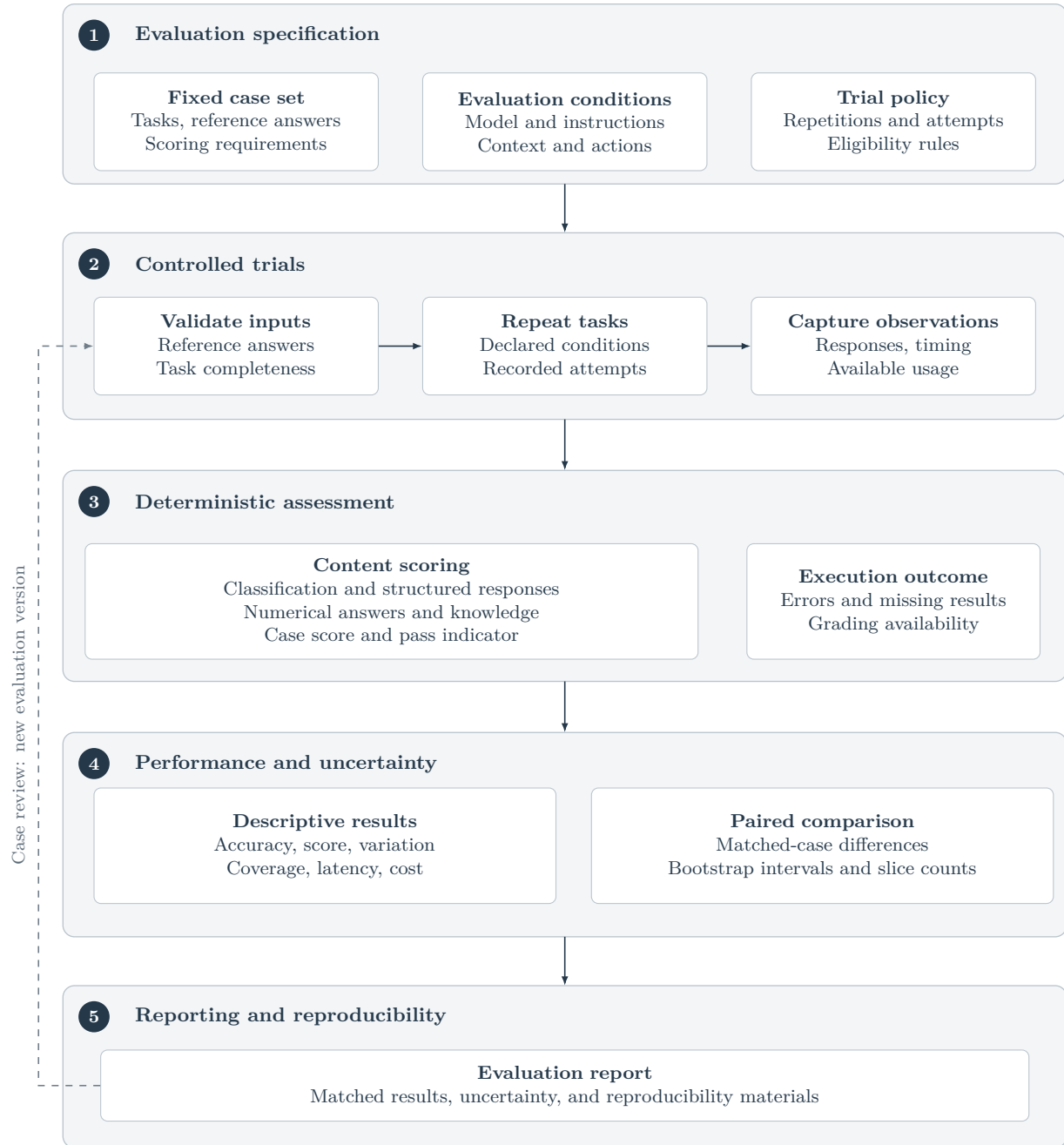


**Figure 1.** Evaluation design. Solid arrows show forward flow. The dashed loop marks case review; changes create a new evaluation version.

**Quick lookup:** Define the case set and conditions, validate inputs, execute repeated trials, score responses, and compare matched cases. Report errors and uncertainty with the performance measures.

# 1 Introduction

Accounting assistants must interpret financial scenarios and produce useful responses under specified conditions. Evaluation therefore needs explicit tasks, reference answers, and acceptance criteria. A plausible explanation alone cannot establish a correct accounting result.

DualBench100 measures whether an assistant’s response satisfies the requirements of a defined accounting case. The evaluated configuration includes the model, task instructions, available context, and permitted actions. Scores characterize that configuration. They do not isolate the contribution of model training from the other evaluation conditions.

A content pass, a correct accounting record, and a completed financial operation are different outcomes. The benchmark’s scoring rules establish the first outcome. Additional evidence is required to establish the others.

## 1.1 Terms

- A **case** contains one task and its reference answer.
- A **repetition** evaluates the selected case set once.
- A **trial** evaluates one case within one repetition.
- An **attempt** is an execution within a trial, including a retry when the declared policy permits one.
- A **configuration** specifies the conditions under which an assistant is evaluated.
- A **corpus** is a versioned collection of cases.

Attempts and repetitions are not interchangeable. Additional attempts can affect cost and latency without adding independent trials.

## 1.2 Related research

AgentBench evaluates language-model agents in interactive environments (Liu et al., 2023). FinanceBench evaluates financial question answering with reference answers and supporting evidence (Islam et al., 2023). These benchmarks illustrate different evaluation objectives.

Spider evaluates text-to-SQL across database schemas (Yu et al., 2018). BIRD emphasizes database contents and query execution quality (Li et al., 2023). Their distinction between an expressed answer and an executed result also matters for accounting evaluation.

Reliable comparisons require documented evaluation conditions and careful interpretation (Lones, 2024). Bootstrap resampling provides a basis for estimating uncertainty from observed results (Efron, 1979). Section 5 defines the paired procedure used in this evaluation profile.

Scores from different benchmarks are not directly comparable. Their case sets, available context, permitted actions, and scoring rules can differ.

# 2 Case composition

## 2.1 Accounting categories

The corpus described here contains 101 unique cases across eight categories:

- Transaction classification: 13 cases.
- Journal entry creation: 13 cases.
- Financial reporting: 13 cases.
- Bank reconciliation: 12 cases.
- Accounts payable: 13 cases.
- Accounts receivable: 12 cases.
- Month-end close: 12 cases.
- Accounting knowledge: 13 cases.

These categories cover distinct accounting tasks. Classification cases identify an appropriate account or treatment. Journal-entry cases assess record structure and financial direction. Reporting cases assess calculations and interpretation. Reconciliation cases assess balance adjustments and matching logic.

Payable and receivable cases assess transaction treatment and payment scenarios. Close cases assess period-end adjustments. Knowledge cases assess accounting concepts through fixed answer choices.

The difficulty labels comprise 37 easy, 36 medium, and 28 hard cases. These labels reflect author judgment. They are not difficulty estimates fitted from measured assistant performance.

## 2.2 Reference answers and scoring methods

Each case specifies the task input, reference answer, and scoring requirements. The task input or permitted reference context must contain the information needed to answer the question. A scenario description that is not provided during evaluation cannot supply missing task information.

The scoring distribution is:

- Structured accounting response: 44 cases.
- Numerical answer: 29 cases.
- Classification or intent: 15 cases.
- Multiple-choice knowledge: 13 cases.

All 101 cases use deterministic scoring. Given the same response and scoring specification, a deterministic scorer produces the same result. This property does not imply deterministic assistant responses.

The cases use synthetic accounting scenarios. Their category mixture defines a fixed evaluation population. It does not estimate the frequency of tasks in any particular business.

## 2.3 Version identity

An evaluation identifies the exact case set and reference answers. A changed task, reference value, or scoring rule creates a different evaluation condition. A benchmark name alone does not identify the evaluated corpus.

The report records the release identifier, case count, and checksum of the evaluation materials. It also identifies any subset, excluded case, or altered expectation. A category subset must not be reported as a complete-corpus result.

Reference answers require review before model evaluation. This review checks numerical calculations, task completeness, and consistency between the expected response and scoring rule. Corrections remain distinguishable from changes to the evaluated assistant.

# 3 Evaluation procedure

The following procedure defines the conditions needed for an interpretable comparison. A publication states which conditions it applies and reports any deviation.

## 3.1 Step 1: Define the evaluation question

State the behavior being measured. Examples include account classification, numerical calculation, record drafting, or task completion with permitted actions.

**Reason:** Different outcomes require different evidence. A response-content score cannot independently prove completion of a financial operation.

## 3.2 Step 2: Fix the case set and acceptance criteria

Select the corpus version, cases, reference answers, and scoring thresholds before execution. Declare the number of repetitions and the error-counting policy.

**Reason:** Changes made after observing results can change the denominator or favor one evaluated configuration.

### 3.3 Step 3: Specify equivalent test conditions

Record model versions, task instructions, available context, permitted actions, and requested inference settings. Identify every evaluated model when a configuration uses more than one.

Keep those conditions equivalent across the comparison except for the declared experimental variable. Record actual settings when they differ from requested settings.

**Reason:** A difference in context or available actions can change an answer independently of the model being compared.

### 3.4 Step 4: Control the accounting context

Establish the accounting information available during each trial. Keep account terminology, amounts, dates, currencies, and relevant relationships consistent with the task specification.

Actions can change that context. Use equivalent starting conditions for comparable trials, or declare that trials share evolving state. Report execution order when state can affect later cases.

**Reason:** A changed balance or record can change the correct answer. Equivalent task wording alone does not establish equivalent conditions.

### 3.5 Step 5: Execute the declared trial policy

Submit the task and record its observable outputs. If retries are permitted, use the same declared policy across compared configurations. Record the attempt count and execution outcome.

Evaluate action-enabled and response-only configurations under their stated conditions. A described action does not establish that the action was performed.

**Reason:** Retries and permitted actions can affect accuracy, latency, and cost. They form part of the evaluated condition.

### 3.6 Step 6: Preserve observations before scoring

Retain the final response, any structured response, the case identifier, the repetition, and the execution outcome. Record elapsed time and available usage information. Keep sufficient evidence to inspect a disputed grade.

**Reason:** Aggregate scores cannot explain a failure or establish whether a scorer used the intended evidence.

### 3.7 Step 7: Apply the fixed scoring rules

Calculate the case score and pass indicator using the selected deterministic method. Distinguish an incorrect response from an execution error or unavailable grade.

Do not revise a reference answer merely because a model disagrees with it. Resolve suspected case defects through a separate review.

**Reason:** Consistent scoring permits matched comparisons. Separate review preserves the distinction between benchmark defects and assistant failures.

### 3.8 Step 8: Summarize and inspect matched results

Calculate repeated-trial summaries and question-level variation. Verify that every scheduled trial has a result or an explicit missing-result record. Compare configurations on matching cases and expectations.

Report error counts, eligible denominators, and uncertainty with the headline measures. Inspect individual changed cases before attributing an aggregate difference to improved accounting ability.

**Reason:** Missing observations and conditional reporting can alter the apparent result. Per-case evidence supports interpretation of the aggregate measures.

## 4 Deterministic scoring profile

Each case receives a score  $s$  between 0 and 1 and a binary pass indicator  $c$ . Classification, structured-response, and numerical methods use a pass threshold of 0.8. Multiple-choice scoring is binary.

The threshold permits partial credit. A pass therefore does not require every expected element to match. Reports retain both the score and the pass indicator.

### 4.1 Classification and intent

Classification scoring checks whether required terms appear affirmatively in the response. A mention that states an account or entity was not found does not establish an affirmative match.

For  $K$  required terms and  $M$  affirmative matches:

$$s = \frac{M}{K}, \quad c = \mathbf{1}[s \geq 0.8]. \quad (1)$$

Publish the required terms, normalization, and negation policy with the scoring specification. These rules remain fixed within a comparison.

Lexical matching measures the declared terms. It does not provide unrestricted semantic validation. For example, an account-name match does not establish a correct account type unless the case also checks that type.

### 4.2 Structured accounting responses

Structured scoring compares the expected response fields. These can include the record type, account references, debit and credit amounts, and total amount. Each expected top-level field receives equal weight.

A line collection receives the fraction of expected lines matched. Matching uses the declared account and amount requirements. An observed line can satisfy one expected line. Additional lines do not automatically reduce this partial-credit score.

For an expected field set  $F$  and field credits  $m_f$  between 0 and 1:

$$s = \frac{1}{|F|} \sum_{f \in F} m_f, \quad c = \mathbf{1}[s \geq 0.8]. \quad (2)$$

Direct monetary comparisons use an absolute difference below 0.01. Treatment of absent and zero debit or credit amounts follows the declared scoring specification. Format normalization and permitted naming equivalents also require fixed rules.

Textual evidence can supply credit when the scoring specification permits it. A matching account name and amount can support partial credit without proving their complete field association. Textual evidence alone does not establish debit or credit direction, record balance, or completed record creation.

Consider an illustrative response with three checked fields. Suppose record type and total receive full credit, while one of two expected lines matches. Its score is:

$$s = \frac{1 + 1 + 0.5}{3} = \frac{5}{6} \approx 0.833. \quad (3)$$

This response passes the 0.8 threshold, although one expected line remains unmatched. The example illustrates the difference between benchmark acceptance and complete record correctness.

### 4.3 Numerical answers

Numerical scoring compares reference values with extracted numerical candidates. Representation normalization accounts for declared currency, grouping, sign, and percentage conventions. Each extracted candidate can satisfy one expected value.

For reference value  $e$  and candidate value  $a$ , the comparison rule is:

$$\text{match}(e, a) = \begin{cases} |a| < 0.01, & e = 0, \\ |e - a| < 0.01 \vee |e - a|/|e| < 0.005, & e \neq 0. \end{cases} \quad (4)$$

For  $V$  expected values and  $M$  matches, the score is  $M/V$ . The response passes when that score is at least 0.8.

The relative tolerance is 0.5%. Its permitted absolute difference increases with the magnitude of the expected value. This tolerance is a benchmark convention, not a universal financial materiality threshold.

Value matching does not independently prove that a number describes the correct financial measure. A correct amount under an incorrect heading can satisfy a value-based check. Keep the full response and inspect the case’s required associations when interpreting results.

## 4.4 Multiple-choice knowledge

Multiple-choice cases provide fixed answer options. The selected answer is compared with the reference selection. A correct selection receives score 1; other selections receive score 0.

This method gives a deterministic knowledge check. It does not measure the ability to construct the answer without options.

## 4.5 Limits of a content pass

The four methods evaluate declared response requirements. They do not collectively prove successful execution, complete financial validity, or accounting correctness beyond those requirements.

A publication therefore identifies the scoring specification and its permitted fallback rules. Additional validation of critical accounting fields remains a separate outcome. A higher benchmark score is evidence about the tested cases, not a guarantee for every accounting task.

# 5 Performance measures and statistical comparison

## 5.1 Accuracy and mean score

For repetition  $r$ , let  $G_r$  contain the trials with defined grades. Let  $c_{qr}$  be the pass indicator and  $s_{qr}$  the score for case  $q$ .

$$A_r = \frac{\sum_{q \in G_r} c_{qr}}{|G_r|}, \quad S_r = \frac{\sum_{q \in G_r} s_{qr}}{|G_r|}. \quad (5)$$

The eligible set requires an explicit policy. An execution failure with no usable answer can receive a zero score. An unavailable grade is reported separately. Partial response content can receive a grade without establishing successful execution.

When no eligible grades exist, report accuracy and mean score as unavailable. Do not substitute a zero observation. Identify eligible repetitions before calculating an experiment-level mean.

Across  $R$  repetitions, the reported means and score variation are:

$$\bar{A} = \frac{1}{R} \sum_r A_r, \quad \bar{S} = \frac{1}{R} \sum_r S_r, \quad \sigma_S = \sqrt{\frac{1}{R} \sum_r (S_r - \bar{S})^2}. \quad (6)$$

This score deviation uses divisor  $R$ . It describes variation between repetition-level mean scores. It is not a confidence interval across cases.

Case-level averages weight each eligible question equally. They do not weight the eight accounting categories equally. Report category and difficulty results with their case counts.

## 5.2 Execution coverage and errors

Distinguish three observations: a graded task failure, an execution error, and an unavailable grade. They answer different questions about performance.

Recorded execution coverage is:

$$C = \frac{\text{trials without recorded execution or grading errors}}{\text{scheduled trials}}. \quad (7)$$

Coverage measures the declared execution records. It does not independently verify every completed financial operation. Missing trials require explicit accounting before analysis.

When the available evidence supports completion status, an additional measure can report complete executions that also pass the content criteria. Label that measure separately from content accuracy.

An accuracy calculated only from completed or gradable trials is conditional on that selection. Report the excluded counts and coverage. Higher conditional accuracy with lower coverage does not establish better overall performance.

### 5.3 Response latency

Response latency measures elapsed execution time until a final response. Evaluation preparation and scoring are separate intervals. Action time and retry waits contribute to execution latency when the declared timing policy includes them.

Use equivalent timing boundaries in a comparison. Differences in initialization, communication, or retry policy can affect the measured result independently of model speed.

Calculate mean latency from valid, positive measurements and report the number of eligible observations. A zero placeholder for a missing measurement is not evidence of an immediate response. The mean across repetitions is the mean of the repetition-level means.

Record total evaluation duration separately when it matters. Response latency does not establish throughput, time to first output, or latency under an operational workload.

### 5.4 Usage and estimated cost

Record available usage by model and sum totals across repetitions. A total over the experiment is different from an average cost per trial.

Use valid provider-reported charges when available. Otherwise apply an identified public price schedule. Record its effective date, currency, model identity, and the basis of the estimate.

For input tokens  $I$ , cached-read tokens  $C$ , cache-write tokens  $W$ , output tokens  $O$ , and prices per million tokens:

$$\text{cost} = \frac{(I - C - W)p_I + Cp_C + Wp_W + Op_O}{10^6}. \quad (8)$$

The token categories must follow the provider’s accounting conventions. Unpriced usage does not mean zero cost. Missing cache rates and incomplete usage need explicit labels.

Discarded attempts, setup calls, and grading can incur charges outside a captured response record. Report whether those charges are included. A response-based estimate is not automatically a complete invoice for the experiment.

### 5.5 Question-level score variation

For each case, record the observed score minimum, maximum, and number of eligible repetitions. Equal observed scores establish score agreement within the experiment. They do not establish identical responses or actions.

At least two eligible observations are needed to examine repeated score variation. Error exclusions must remain visible. A case with one eligible result cannot establish repeated consistency.

Repetition count is a declared sampling choice. More repetitions can expose variability, but no fixed count guarantees a stable ranking for every case set.

### 5.6 Paired bootstrap comparison

Compare configurations on matching case inputs, reference answers, and scoring rules. Collapse each case’s eligible repetitions into a mean score and a strict-majority pass indicator:

$$\tilde{s}_q = \text{mean}_r(s_{qr}), \quad \tilde{c}_q = \mathbf{1}[\text{mean}_r(c_{qr}) > 0.5]. \quad (9)$$

A tied pass rate does not satisfy the strict majority. The majority indicator and mean repetition accuracy are different summaries. Their labels remain distinct.

For configurations  $a$  and  $b$ , calculate paired score differences  $d_q = \tilde{s}_{qb} - \tilde{s}_{qa}$ . Accuracy differences use the corresponding majority indicators. Only matched cases enter the paired analysis.

Resample the case-level differences with replacement 10,000 times. Calculate the mean difference in each resample. Use its 2.5th and 97.5th percentiles for a 95% percentile interval. Record the random seed for reproduction (Efron, 1979).

An interval entirely above zero supports a positive difference under this procedure. An interval entirely below zero supports a negative difference. An interval that includes zero does not provide clear directional evidence. It does not establish equivalence.

Slices with fewer than 10 matched cases receive descriptive treatment rather than a directional classification. This minimum is a reporting convention, not a statistical power guarantee.

The procedure resamples cases after repetition-level aggregation. It is not a joint analysis of every source of execution variability. Related cases can violate an independent-case interpretation. Per-category intervals also require attention to multiple comparisons.

Report the matched case count, missing observations, eligibility rules, and interval method. Verify that the retained repetitions are complete before interpreting a difference.

## 6 Reporting and reproducibility

A reproducible report identifies the corpus version, selected cases, reference answers, scoring rules, and evaluation conditions. It retains trial-level observations rather than only aggregate scores.

The report includes:

- Model versions, task instructions, available context, and permitted actions.
- Requested and actual inference settings.
- Repetition count, attempt policy, and control of task state.
- Per-case scores, pass indicators, execution outcomes, and missing results.
- Accuracy, score variation, category counts, and the eligible denominators.
- Timing boundaries, available usage, cost basis, and excluded charges.
- Paired differences, interval settings, and the random seed.

Independent reproduction requires an archived case set, reference answers, task-input specification, and complete scoring specification. A benchmark title or summary percentage cannot replace those materials.

If a task or scoring rule changes after review, identify the changed version. If a configuration changes during execution, identify the affected observations. Do not combine incompatible conditions into an unexplained summary measure.

## 7 Interpretation limits

Synthetic tasks cover a specified range of accounting scenarios. Their distribution does not represent every organization, transaction, or accounting policy. Author-assigned difficulty also does not establish universal task difficulty.

Partial credit, lexical checks, and numerical tolerances support consistent scoring. They can also accept responses with incomplete associations or unchecked financial details. Interpret each score through the declared acceptance criteria.

Results measure the evaluated configuration under the reported conditions. They do not directly estimate performance under different context, instructions, available actions, or workload.

Repeated-trial means describe the observed experiment. Paired bootstrap intervals describe case-level differences under the stated assumptions. Conditional error exclusions, related cases, and exploratory slices limit broader conclusions.

This methods paper contains no measured ranking, accuracy improvement, latency improvement, or cost saving for a model. Any empirical claim requires the corresponding observations and reporting conditions.

## 8 Conclusion

DualBench100 provides a defined set of accounting tasks and deterministic response checks. Explicit reference answers and acceptance criteria make its scores interpretable. Repeated trials describe variation, while paired comparisons estimate differences between matched configurations.

The principal reporting requirement is traceability from each aggregate measure to its case observations and evaluation conditions. Content quality, execution coverage, latency, and estimated cost remain separate quantities. Their combined interpretation supports a technically defensible assessment of accounting assistants.

## References

- Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1), 1–26.  
<https://doi.org/10.1214/aos/1176344552>
- Islam, P., Kannappan, A., Kiela, D., Qian, R., Scherrer, N., & Vidgen, B. (2023). *FinanceBench: A new benchmark for financial question answering* [Preprint]. arXiv.  
<https://doi.org/10.48550/arXiv.2311.11944>
- Li, J., Hui, B., Qu, G., Yang, J., Li, B., Li, B., Wang, B., Qin, B., Cao, R., Geng, R., Huo, N., Zhou, X., Ma, C., Li, G., Chang, K. C. C., Huang, F., Cheng, R., & Li, Y. (2023). *Can LLM already serve as a database interface? A big bench for large-scale database grounded text-to-SQLs* [Preprint]. arXiv.  
<https://doi.org/10.48550/arXiv.2305.03111>
- Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., Gu, Y., Ding, H., Men, K., Yang, K., Zhang, S., Deng, X., Zeng, A., Du, Z., Zhang, C., Shen, S., Zhang, T., Su, Y., Sun, H., . . . Tang, J. (2023). *AgentBench: Evaluating LLMs as agents* [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2308.03688>
- Lones, M. A. (2024). Avoiding common machine learning pitfalls. *Patterns*, 5(10), Article 101046.  
<https://doi.org/10.1016/j.patter.2024.101046>
- Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I., Yao, Q., Roman, S., Zhang, Z., & Radev, D. (2018). Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In E. Riloff, D. Chiang, J. Hockenmaier, & J. Tsujii (Eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 3911–3921). Association for Computational Linguistics. <https://doi.org/10.18653/v1/D18-1425>