International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Benchmark-Driven Evaluation Frameworks for AI Agent Trustworthiness in Enterprise Analytics

Santosh Reddy Muthyala

Abstract: Enterprise adoption of AI-powered analytics agents has outpaced the methods available to evaluate whether their outputs can be trusted. General-purpose benchmarks such as MMLU [1] and HumanEval [2] measure static knowledge recall and isolated code-generation ability, while text-to-SQL benchmarks such as Spider [3] and BIRD [4] test query generation against simplified schemas. None of these evaluate the full chain of reasoning an enterprise analytics agent must perform: schema interpretation, organization-specific metric computation, calibrated uncertainty, and application of the correct analytical methodology. On BIRD, even the strongest reported system reaches only 40.08% execution accuracy against realistic enterprise schemas, compared with 92.96% for human analysts [4], a gap that becomes consequential once an agent's output informs financial or regulatory decisions rather than exploratory analysis. This article proposes a nine-dimension trustworthiness evaluation framework purpose-built for enterprise analytics agents. Four dimensions are grounded in established trust taxonomies [6]-[8]; five are analytics-specific contributions not addressed by any existing framework: Schema Fidelity, Metric Definition Adherence, Failure Mode Transparency, Business Context Understanding, and Analytical Methodology Competence. The article also proposes a benchmark design methodology combining automated, LLM-as-judge, and domain-expert graders, and a five-phase Continuous Benchmark Evolution Lifecycle for keeping evaluation infrastructure current as schemas, metric definitions, and agent capabilities change.

Keywords: *AI agent trustworthiness; enterprise analytics; text-to-SQL evaluation; benchmark design; LLM-as-judge; responsible AI adoption*

1. Introduction

Analytics agents capable of interpreting business questions, generating SQL, and producing data-driven summaries are moving from pilot projects into production workflows across large organizations. The capability gap driving this shift is not subtle: agents that once required carefully scoped prompts now handle open-ended questions against live production schemas, and the volume of decisions touched by their output has grown accordingly. What has not kept pace is the infrastructure to evaluate whether that output deserves the trust it is often given by default.

Ninety-five real-world databases totaling 33.4 GB make up the BIRD benchmark, deliberately constructed to be closer to enterprise conditions than earlier text-to-SQL datasets [4]. Even so, the best-reported system in the original BIRD paper achieves only 40.08% execution accuracy, against a human baseline of 92.96% [4]. Spider, an earlier and cleaner benchmark built from well-named, modestly sized schemas, reported far higher accuracy under simpler conditions [3], a difference in outcome that tracks the difference in schema realism rather than any change in underlying model capability. Real enterprise data warehouses compound this difficulty further: hundreds of tables, abbreviated or historically inconsistent column names, and business rules embedded in views rather than documented anywhere an agent could retrieve them. General-purpose evaluation suites were not built to surface this kind of failure. MMLU tests static academic knowledge across 57 subjects [1]; HumanEval tests isolated function-level code generation [2]; HELM extends coverage to 42 scenarios across seven metrics including robustness and fairness but does not include enterprise-specific scenarios [5]. None of these ask the questions that matter for an analytics agent operating inside a real organization: Does it preserve numerical precision rather than rounding or approximating figures it

Independent Researcher, USA

should compute exactly? Does it correctly resolve ambiguous column names to the tables and joins an analyst would use? Does it apply the organization's own definition of a metric rather than a generic textbook one? Does it recognize the limits of its own knowledge and say so, rather than producing a fluent but wrong answer? Does it interpret results with enough business context to avoid drawing a misleading conclusion from a correct query? And can it select the right analytical method, whether experimentation, root-cause investigation, forecasting, or model evaluation, for the task in front of it?

This article proposes a nine-dimension trustworthiness evaluation framework that answers these questions directly. The framework builds on established trust taxonomies, including the NIST AI Risk Management Framework's seven trustworthiness characteristics [6], DecodingTrust's eight-category assessment of GPT-family models [7], and TrustLLM's eight defined dimensions, six of which its authors empirically benchmark [8], while introducing five dimensions specific to enterprise analytics that none of these frameworks were designed to capture. The remainder of the article positions this framework against existing evaluation literature (Section 2), presents the nine dimensions in detail (Section 3), proposes a benchmark design and multi-grader scoring methodology (Section 4), and discusses the framework's implications for staged enterprise adoption and long-term benchmark maintenance (Section 5).

2. Related Work

2.1 General-Purpose LLM Benchmarks

The benchmarks most commonly used to compare frontier models share a common limitation for this article's purposes: none targets the analytics workflow specifically. MMLU evaluates multiple-choice knowledge recall across academic subjects using 15,908 questions spanning 57 subjects [1]; it measures what a model knows, not whether it can reason correctly over an unfamiliar enterprise schema. HumanEval evaluates function-level code generation using 164 hand-written Python problems with unit tests [2]; its scope is deliberately narrow, and it does not test SQL generation, multi-file reasoning, or the schema-discovery step that precedes any real analytics query. HELM broadens the lens considerably, covering 42 scenarios across seven metrics, accuracy, calibration, robustness, fairness, bias, toxicity, and efficiency [5], but its scenario set, like MMLU's and HumanEval's, was not constructed with enterprise data infrastructure in mind. MT-Bench, which scores 80 multi-turn conversational questions using an LLM judge, extends evaluation into multi-turn dialogue but at a scale too small and a scope too general to substitute for domain-specific evaluation [11]. AgentBench evaluates language models as agents across eight distinct environments, including a database environment; GPT-4, the strongest model reported in the paper, models struggle specifically with database-grounded, tool-mediated tasks, but none of them was designed to diagnose why, or to certify whether a given deployment meets the bar an enterprise actually needs.

2.2 Text reaches a database-task success rate of only 32.0%, its weakest showing among the eight environments other than web shopping [12]. Read together, these benchmarks establish that current ***-to-SQL and Enterprise Analytics Benchmarks***

Text-to-SQL benchmarks come closer to the analytics use case, and their trajectory is instructive. Spider, an early and influential dataset, evaluates query generation across 200 databases and 10,181 natural-language questions using clean, well-documented schemas [3]. BIRD was constructed specifically to close the realism gap Spider leaves open, using 95 real-world databases with less curated naming conventions [4]; the resulting accuracy drop, from Spider's much higher reported figures down to BIRD's 40.08% for the best system tested, illustrates how much of a model's apparent competence is an artifact of schema cleanliness rather than genuine query-generation skill. Applying a battery of perturbations to this same fragility question, Dr.Spider tests whether accuracy holds up under renamed columns and altered natural-language phrasing, conditions that are routine in production schemas rather than exceptional [13]. DAIL-SQL demonstrates one partial remedy: retrieving relevant schema elements and similar worked examples before generation improves execution accuracy to 86.6% on Spider [14], showing that better context, not just a larger model, closes part of the gap. DIN-SQL applies decomposed, self-correcting in-context learning to the same problem [16]. DS-1000 extends the evaluation lens to data-science code generation beyond SQL, reporting that the best system evaluated in the original paper, Codex-002, reaches only 43.3% overall accuracy and roughly 26.5% on

the Pandas-specific subset [17], evidence that the difficulty of realistic data manipulation tasks is not confined to SQL generation alone.

2.3 Trust Taxonomies

A separate body of work defines what trustworthiness means for AI systems generally, without targeting analytics specifically. The NIST AI Risk Management Framework organizes trustworthiness into seven characteristics: validity and reliability, safety, security and resilience, accountability and transparency, explainability and interpretability, privacy enhancement, and fairness with managed bias [6]. This framework was designed to apply across the full span of AI applications, from computer vision to recommendation systems to conversational agents, and its generality is both its strength and its limitation for the purposes of this article: it tells an organization what categories of trust concern exist, but it offers no operational guidance on how validity and reliability should be measured for an agent whose core task is translating a business question into a correct database query.

DecodingTrust operationalizes eight of these concerns, including robustness, fairness, and privacy, into a concrete assessment protocol for GPT-family models [7]. Its methodology is instructive precisely because it demonstrates what a rigorous, dimension-specific evaluation protocol looks like: rather than asking a single holistic question about whether a model is trustworthy, it constructs targeted test scenarios for each of its eight categories and reports dimension-level scores rather than a single aggregate figure. The analytics-specific framework proposed in Section 3 borrows this same design principle, but DecodingTrust's own eight categories were built around adversarial robustness, toxicity, and privacy leakage in general text generation, none of which maps onto schema fidelity or metric-definition correctness.

TrustLLM defines eight trustworthiness dimensions and empirically benchmarks six of them, given that transparency and accountability resist direct automated measurement [8]. The two dimensions TrustLLM's own authors were unable to benchmark automatically are worth dwelling on: both concern whether a system's reasoning can be inspected and traced to a responsible party, which is conceptually adjacent to Reasoning Transparency in Section 3.1 but framed at the level of a general-purpose language model rather than an agent operating over a specific, auditable schema. This distinction matters because a schema, unlike open-ended text generation, provides a concrete, checkable structure against which reasoning transparency can be partially automated: an agent's stated join logic can be checked against the schema's actual foreign-key relationships, for instance, in a way that a general conversational claim cannot be checked against an ill-defined notion of truth.

These frameworks establish a vocabulary and a measurement discipline this article draws on directly, but none defines dimensions for schema-grounded reasoning, organization-specific metric correctness, or the analytical-methodology competence that separates a capable data scientist from a fluent query generator. That gap is the starting point for the framework proposed in Section 3.

3. Proposed Trustworthiness Evaluation Framework

3.1 Four Core Trust Dimensions

Four of the nine dimensions in the proposed framework are adaptations of concerns already established in the general trust literature, reframed for the analytics context.

Data Fidelity asks whether an agent accurately retrieves and represents underlying data, not merely whether its SQL executes without error. An agent that rounds a reported revenue figure, or substitutes a mean where a median would be the correct aggregation for the question asked, fails this dimension even when its query is syntactically valid.

Hallucination Rate captures how often an agent fabricates information that has no basis in the underlying data or schema. In an analytics context, this takes at least three distinguishable forms: schema hallucination, where the agent references a column or table that does not exist; data hallucination, where it states a numerical value not actually present in any query result; and relationship hallucination, where it assumes a join path or foreign-key relationship that the schema does not support. A survey of hallucination research documents that reported rates vary considerably by domain and task setup [18], underscoring why this dimension needs a domain-specific benchmark rather than a borrowed general-purpose figure.

Reasoning Transparency asks whether an agent's intermediate steps, why it selected a given table, how it constructed a join, what assumptions it made when interpreting an ambiguous question, are inspectable rather than opaque. The LLM-as-judge

paradigm, shown to correlate reasonably well with human evaluators on rubric-based tasks [11], is one practical mechanism for scoring this dimension at scale, discussed further in Section 4.2.

Consistency measures whether an agent produces the same result when asked the same question against an unchanged data state more than once. Response variation even at low sampling temperature is a documented characteristic of current model-serving infrastructure, and for a decision-support tool, inconsistency by itself is a trust failure independent of whether any single response happens to be correct.

3.2 Five Analytics-Specific Dimensions

The remaining five dimensions are, to this article's knowledge, not addressed as a coherent set by any existing published trust framework, though each responds to a documented failure pattern in text-to-SQL and agentic evaluation research.

Schema Fidelity goes beyond hallucination detection to ask whether an agent correctly uses schema elements it has not fabricated. An agent can avoid referencing a nonexistent column while still misusing a real one, joining on the wrong key, or applying a string operation to a column typed as numeric. Dr.Spider's perturbation results, which show that column renaming alone degrades accuracy meaningfully even when the underlying schema logic is unchanged [13], point to how fragile this competency remains even among otherwise capable models.

Metric Definition Adherence asks whether an agent implements an organization's specific definition of a metric rather than a generic interpretation. Active user can mean a unique login, a unique session, a unique engaged session or a unique engaged specific event during a specific period, depending on the organization asking the question; a technically correct query built on the wrong definition produces a number that is precise and wrong at the same time. This dimension cannot be evaluated through schema inspection alone; it requires comparison against a curated, organization-specific set of metric definitions.

Failure Mode Transparency asks whether an agent recognizes the limits of what it can reliably answer and says so, rather than producing a confident but unsupported response. This is arguably the single most consequential dimension for enterprise deployment: a wrong answer delivered with unwarranted confidence is more damaging to downstream trust than an honest admission that it is not confident in a schema mapping and that a human should verify it, yet current models are not well calibrated in this regard [1].

Business Context Understanding evaluates whether an agent grasps enough of the business domain to interpret a result rather than merely compute it. A revenue comparison across quarters that ignores seasonality, or a user-growth figure evaluated against the wrong baseline for a market's maturity, can be numerically correct and analytically misleading at once. This dimension differs from Metric Definition Adherence in that it concerns interpretation of a correctly computed result rather than the mechanics of the computation itself; internal analytics platforms built to surface contextual signals alongside raw metrics, for instance a retrieval layer spanning query history, dashboards, and prior analyses so that an agent's response reflects accumulated organizational context rather than the current question in isolation, are one practical approach to strengthening this dimension in deployed systems, drawn from the author's primary professional experience building such a retrieval-augmented assistant at a billion-user-scale technology platform.

Analytical Methodology Competence asks whether an agent applies the correct analytical approach for the task, not merely whether it can produce syntactically valid SQL. Reading an experiment result correctly requires recognizing statistical significance conventions and common pitfalls such as sample ratio mismatches; diagnosing a metric movement requires decomposing it across dimensions rather than treating every change as an anomaly; forecasting requires distinguishing seasonality from a genuine trend and communicating the resulting uncertainty rather than presenting a point estimate as certain. An agent that writes flawless SQL but applies a simple period-over-period comparison where a controlled experiment design was called for fails this dimension even when every other dimension scores well, a failure mode this article regards as the clearest line between a query-generation tool and a genuine analytics collaborator.

4. Benchmark Design and Evaluation Methodology

4.1 Test Suite Construction

A benchmark capable of exercising all nine dimensions needs test cases spanning several levels of complexity: single-table aggregations that test

basic data fidelity; multi-table joins encoding business rules, which test schema fidelity and join-path reasoning; time-series queries that test handling of date arithmetic and period comparisons; metric-computation queries that specifically test adherence to organization-specific definitions rather than generic ones; and multi-step analytical workflows, where later queries depend on the results of earlier ones, which test reasoning transparency and consistency under compounding conditions. A benchmark limited to any single complexity tier will systematically overstate an agent's readiness for the tiers it does not cover.

Each of these five complexity tiers exercises the nine dimensions unevenly, which is itself a design consideration rather than an incidental fact. Single-table aggregation cases are cheap to construct and to grade automatically, making them well suited to high-volume, continuous regression testing, but they say little about Schema Fidelity or Business Context Understanding, since there is no join-path ambiguity or contextual interpretation required to answer them correctly. Multi-table join cases are the tier where Schema Fidelity failures concentrate in practice, and a benchmark that underweights this tier relative to single-table cases will tend to understate how often an agent misjoins tables or misapplies a business rule encoded only in a WHERE clause. Multi-step analytical workflows are the most expensive tier to construct and grade, because each step's ground truth depends on the correctness of the step before it, but they are also the only tier capable of exercising Consistency and Reasoning Transparency under realistic compounding conditions, since a single-turn query cannot reveal whether an agent's intermediate reasoning remains coherent across a sequence of dependent questions.

A minimum viable suite should draw from multiple enterprise-representative schemas spanning domains such as financial reporting, marketing analytics, and product metrics, with each test case carrying the natural-language question, the schema, at least one valid ground-truth query, the expected result set, and metadata indicating which trust dimensions the case is designed to probe.

4.2 Ground Truth and Scoring

Three methodological choices shape whether a benchmark's scores mean anything. Execution-based evaluation is preferable to exact-match comparison against a single gold query, because multiple structurally different SQL statements can return identical, correct results; scoring against result sets rather than query text avoids penalizing valid alternative formulations. Binary correct or incorrect scoring, on the other hand, understates the difference between a query that is completely wrong and one that returns nearly all correct rows but misses an edge-case filter; partial-credit scoring, for instance using set-similarity measures between predicted and gold result sets with additional penalties for numerical precision errors, better reflects the practical difference between these two failure modes. No single grading method, finally, can score all nine dimensions, because they differ fundamentally in what kind of judgment they require.

Table 1 summarizes how each dimension maps to a grading approach.

Table 1. Mapping of Trust Dimensions to Grading Approach

Grader Type	Dimensions	Scalability	Cost	Calibration Needed
Automated Deterministic	Data Fidelity, Schema Fidelity, Consistency, Hallucination Rate	High - every test case	Low	One-time setup
LLM-as-Judge	Reasoning Transparency, Failure Mode Transparency, Business Context Understanding	Medium - API cost per evaluation	Medium	Periodic human calibration
Domain Expert	Metric Definition Adherence, Analytical Methodology Competence	Low - sampled evaluation	High	Panel calibration for inter-rater reliability

Automated deterministic graders suit dimensions with objectively verifiable ground truth. Data Fidelity can be checked by comparing agent output against gold result values within a numerical tolerance band; Schema Fidelity can be checked through static analysis of the generated query's abstract syntax tree against the schema catalog; Consistency can be checked by running the same

query multiple times and computing variation across runs; Hallucination Rate can be checked by cross-referencing every referenced table and column name against the schema catalog. None of these requires human or model judgment at evaluation time.

LLM-as-judge graders suit dimensions requiring semantic judgment against a defined rubric rather than an exact numerical answer. Reasoning Transparency can be scored by asking a judge model whether an agent's stated rationale explains its table selection and join logic and acknowledges its assumptions, a rubric-based approach shown to correlate reasonably well with human evaluators [11]. Failure Mode Transparency can be scored by presenting an agent with deliberately ambiguous or unanswerable questions and evaluating whether it flags uncertainty rather than answering with unwarranted confidence. Business Context Understanding can be scored by presenting results that require situational interpretation and evaluating whether the agent's explanation accounts for the relevant context.

Domain expert graders are reserved for dimensions where organizational or methodological expertise cannot be reliably encoded into an automated rubric. Metric Definition Adherence often turns on distinctions, such as whether active user excludes automated accounts or whether revenue nets out refunds, that are specific to an organization's own definitions and not something a general-purpose grader can be expected to know. Analytical Methodology Competence requires judging whether an agent chose the statistically appropriate method for a given question, itself a data-science judgment call best made by a data scientist, ideally as part of a small calibrated panel rather than a single unreplicated review.

A production evaluation pipeline combines all three grader types: automated graders run on every test case to provide continuous, low-cost coverage; LLM-as-judge graders run at a scale between full coverage and expert-only review, with periodic human calibration to confirm the judge model's rubric application remains accurate; domain-expert graders run on a sampled basis, sufficient to establish reliable per-dimension scores and catch systematic failures the automated and LLM-based graders miss.

4.3 Composite Trust Score

Individual dimension scores can be combined into a single composite Trust Score using domain-specific weights, expressed as TrustScore equals the sum over i of w sub i times D sub i, where D sub i is the normalized score, on a scale of 0 to 1, for dimension i, and w sub i is a weight reflecting how much that dimension matters for a given analytics domain. Table 2 illustrates suggested weight profiles.

Table 2. Suggested Trust Score Weight Profiles by Analytics Domain

Dimension	Financial Analytics	Marketing Analytics	Product Analytics
Data Fidelity	0.25	0.15	0.20
Hallucination Rate	0.25	0.15	0.20
Reasoning Transparency	0.10	0.10	0.10
Consistency	0.15	0.10	0.10
Schema Fidelity	0.10	0.20	0.15
Metric Definition Adherence	0.05	0.15	0.15
Failure Mode Transparency	0.05	0.05	0.05
Business Context Understanding	0.10	0.10	0.10
Analytical Methodology Competence	0.10	0.10	0.10

Financial analytics weights Data Fidelity and Hallucination Rate most heavily, reflecting the regulatory stakes of numerical error in that domain; marketing and product analytics weight Schema Fidelity and Metric Definition Adherence higher, reflecting how much those domains depend on custom, frequently changing metric definitions. Business Context Understanding and Analytical Methodology Competence carry comparable weight across all three profiles, reflecting that both matter regardless of domain.

A production-scale realization of this kind of composite scoring is achievable: an evaluation pipeline built by the author at a billion-user-scale technology platform ingests more than 100,000 evaluation submissions through automated quality gates and LLM-powered semantic de-duplication before scoring, drawn from the author's primary professional experience, illustrating that dimension-specific, multi-grader scoring at the volume described in this section is a solved engineering problem, not merely a theoretical proposal.

5. Discussion

5.1 Interpretation of Findings from Existing Benchmarks

The benchmark evidence surveyed in Section 2 supports a consistent pattern relevant to the framework proposed here. Accuracy degrades with schema complexity and departs sharply from clean-schema conditions: the gap between Spider's simplified schemas and BIRD's real-world ones is large enough that schema realism, not underlying model capability, appears to be the dominant factor [3], [4]. Column-renaming perturbations alone measurably degrade accuracy, evidence that current systems rely more on surface-level naming cues than on robust schema understanding [13]. This pattern has a direct implication for how the Schema Fidelity dimension should be operationalized in practice: a benchmark that tests only well-named, stable schemas will systematically overstate an agent's real-world Schema Fidelity score, because the fragility Dr.Spider documents is precisely the failure mode a clean-schema test suite cannot surface.

Multi-step reasoning benefits substantially from tool use and language feedback: MINT reports absolute performance gains of 1 to 8 percent per additional turn of tool use and 2 to 17 percent with natural-language feedback [15], which suggests that Reasoning Transparency and iterative self-verification are not merely nice-to-have properties but active levers for improving Schema Fidelity and Data Fidelity in practice. This finding reframes Reasoning Transparency, in particular, as more than an auditability feature: an agent that exposes its intermediate reasoning creates the surface area for the kind of turn-by-turn feedback MINT shows to be effective, whereas an agent that produces only a final answer forecloses that improvement pathway entirely. Retrieval of relevant schema context, as demonstrated by DAIL-SQL's improvement to 86.6% accuracy on Spider [14], reinforces the same conclusion from a different angle: much of what looks like a model-capability gap is actually a context-availability gap, which the framework's Business Context Understanding and Schema Fidelity dimensions are designed to surface rather than obscure behind an aggregate accuracy figure.

Taken together, these findings suggest that the nine proposed dimensions are not independent of one another in practice, even though they are scored separately. Improvements to Reasoning Transparency appear to produce downstream gains in Schema Fidelity and Data Fidelity, and improvements to retrieval and context availability appear to do the same. An organization applying this framework should therefore expect dimension scores to move together to some degree, and a persistent divergence, for instance strong Data Fidelity alongside weak Schema Fidelity, is itself a diagnostic signal worth investigating rather than an artifact of measurement noise.

5.2 The Trust Gap and a Staged Adoption Model

Even where the individual benchmark evidence is encouraging, translating dimension scores into a deployment decision requires more than a single accuracy threshold. Executive surveys point to why: Accenture's 2024 survey of AI risk maturity found that more than half of surveyed companies lack a systematic process for identifying AI risk before deployment, with reliability cited by 45 percent of respondents as one of their top three risk concerns [9]. A separate 2025 Accenture survey found that 77 percent of executives believe AI's genuine business benefits are only realizable once a foundation of trust in its performance exists [10], a belief that a single pass or fail benchmark score cannot, by itself, satisfy.

The nine-dimension framework instead supports a staged deployment model tied to a composite Trust Score threshold rather than a binary readiness decision. At a Trust Score around 0.70, an agent is suited to assisted exploration, where a human analyst verifies every output and the agent functions primarily as an accelerant for hypothesis generation and query drafting. At approximately 0.80, an agent can move to supervised operations, generating operational reports subject to automated sanity checks and human spot-checks on a sampling basis rather than full review. At approximately 0.90, an agent can operate with guardrails in well-defined, stable use cases, producing and distributing analytics autonomously with automated alerting when outputs fall outside expected ranges. Only at

approximately 0.95 does full autonomy, where an agent's output triggers downstream decisions or automated actions without a human review step, become a reasonable proposition, and even then comprehensive audit logging and periodic review remain warranted. This graduated structure reframes enterprise adoption as a continuum calibrated against the framework's own dimension scores, rather than a single go or no-go decision made once and revisited only after a visible failure.

5.3 Treating the Benchmark Itself as a Living System

A benchmark built once and left unchanged degrades in usefulness even if the agent being evaluated does not change at all, because the production environment around it keeps moving: schemas are altered as new data sources are onboarded, metric definitions shift as the business evolves, and agent capabilities improve to the point where previously discriminating test cases stop differentiating good performance from great performance. This article proposes a five-phase Continuous Benchmark Evolution Lifecycle to address that drift directly.

Phase one, benchmark construction, extracts representative schemas, documents metric definitions, and establishes ground truth through expert review, the most labor-intensive phase but the one that produces the durable evaluation asset the remaining phases depend on. Phase two, agent evaluation, runs the suite on a recurring schedule, weekly or triggered by a model update, schema change, or data refresh, scoring each dimension with its appropriate grader and tracking scores over time so that a degradation on any single dimension is visible before it becomes a production incident. Phase three, agent improvement, closes the loop from evaluation back into engineering: a drop in Schema Fidelity motivates an update to the agent's schema-retrieval corpus; weak Analytical Methodology Competence on experimentation tasks motivates targeted few-shot examples or tool integrations; lagging Business Context Understanding motivates enriching the semantic layer with domain documentation.

Phase four, benchmark drift detection, watches for the signals that the benchmark itself has gone stale: score saturation on a dimension, schema drift between the benchmark and production, metric-definition drift, the emergence of new analytical methodologies the benchmark does not yet test, and production failures the benchmark did not anticipate. Phase five, benchmark refresh, acts on those signals by adding test cases sourced from corrected production failures, retiring saturated cases that no longer discriminate performance, updating schemas and metric definitions to match the current environment, and recalibrating grader rubrics, feeding directly back into phase two to complete the cycle.

This lifecycle is not a hypothetical prescription. In the author's experience building and operating a company-wide AI benchmark evaluation platform at a billion-user-scale technology platform, applying an equivalent evolution cycle scaled real-agent-traffic benchmark coverage from approximately 30 percent to more than 85 percent and grew the evaluation set by roughly a factor of three over a sustained period, drawn from the author's primary professional experience, outcomes consistent with what the five-phase model predicts when the benchmark-refresh phase is resourced as a continuing function rather than a one-time project.

5.4 Limitations

The nine-dimension framework and its associated grading methodology are proposed as a structure for organizations to instantiate against their own schemas and metric definitions; this article does not report a controlled empirical evaluation applying all nine dimensions against a single benchmark suite, and the dimension weightings in Table 2 are illustrative starting points rather than values derived from a formal weight-optimization study. LLM-as-judge scoring, while shown to correlate reasonably well with human judgment on rubric-based tasks [11], still requires periodic recalibration to guard against judge-model drift, and domain-expert grading, necessary for the two dimensions that resist automated scoring, introduces a cost and scalability constraint that organizations must plan for rather than treat as incidental.

5.5 Future Work

Future work includes empirically validating the proposed dimension set against a shared, multi-organization benchmark suite; studying how Trust Score thresholds for staged deployment should vary by regulatory context beyond the financial, marketing, and product domains illustrated in Table 2; and developing more automated methods for the two dimensions currently reserved for domain-expert grading, particularly as LLM-as-judge techniques continue to mature toward the reliability automated deterministic grading already offers for the four core dimensions.

6. Conclusion

Enterprise analytics agents are being deployed faster than the methods available to evaluate whether their outputs deserve trust. This article has argued that closing that gap requires moving past general-purpose capability benchmarks toward a dimension-specific framework built for the analytics workflow itself, one that separates data fidelity and hallucination from the harder, analytics-specific questions of schema fidelity, organization-specific metric correctness, calibrated failure signaling, business-context judgment, and methodological competence. The contribution of this article is that combined framework, together with a benchmark design methodology that matches each dimension to the grading approach it actually requires, and a five-phase lifecycle for keeping the benchmark itself current as schemas, metrics, and agent capabilities change. The practical implication is that enterprises do not need to wait for a fully mature agent before building this evaluation infrastructure: the benchmark, the organizational calibration process, and the trust-scoring discipline developed during early deployment are themselves the foundation that later, more capable agents will be measured against.

Author Contributions

Conceptualization, methodology, investigation, writing (original draft, review and editing), visualization: the sole author.

Data Availability

The benchmarks and datasets discussed in this article are publicly available from their respective original sources as cited in the reference list. No new dataset was created for this article.

Conflicts of Interest

The author declares no conflict of interest.

Declaration on the Use of Generative AI and AI-Assisted Technologies

The author used Claude (Anthropic, claude-sonnet model) to assist with manuscript drafting and language refinement. The author takes full responsibility for the accuracy, originality, and integrity of all content presented in this manuscript, and confirms that all data, analysis, and conclusions reflect independent work and verified sources.

References

[1] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring Massive Multitask Language Understanding," in Proc. ICLR, 2021. arXiv:2009.03300.

[2] M. Chen, J. Tworek, H. Jun, et al., "Evaluating Large Language Models Trained on Code," arXiv:2107.03374, 2021.

[3] T. Yu, R. Zhang, K. Yang, et al., "Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task," in Proc. EMNLP, 2018. DOI: 10.18653/v1/D18-1425.

[4] J. Li, B. Hui, G. Qu, et al., "Can LLM Already Serve as A Database Interface? A BIg Bench for Large-Scale Database Grounded Text-to-SQL Evaluation," in Proc. NeurIPS Datasets and Benchmarks Track, 2023. arXiv:2305.03111.

[5] P. Liang, R. Bommasani, et al., "Holistic Evaluation of Language Models," arXiv:2211.09110, 2022.

[6] National Institute of Standards and Technology, "AI Risk Management Framework (AI RMF 1.0)," NIST AI 100-1, 2023. DOI: 10.6028/NIST.AI.100-1.

[7] B. Wang, W. Chen, H. Pei, et al., "DecodingTrust: A Comprehensive Assessment of Trustworthiness in GPT Models," in Proc. NeurIPS, 2023. arXiv:2306.11698.

[8] Y. Huang, L. Sun, H. Wang, et al., "TrustLLM: Trustworthiness in Large Language Models," arXiv:2401.05561, 2024.

[9] Accenture, "From Compliance to Confidence: Embracing a New Mindset to Advance Responsible AI Maturity," Accenture Research, 2024.

[10] Accenture, "Technology Vision 2025," Accenture Research, 2025.

[11] L. Zheng, W. Chiang, Y. Sheng, et al., "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena," in Proc. NeurIPS Datasets and Benchmarks Track, 2023. arXiv:2306.05685.

[12] X. Liu, H. Yu, H. Zhang, et al., "AgentBench: Evaluating LLMs as Agents," in Proc. ICLR, 2024. arXiv:2308.03688.

[13] S. Chang, J. Wang, M. Dong, et al., "Dr.Spider: A Diagnostic Evaluation Benchmark towards Text-to-SQL Robustness," in Proc. ICLR, 2023. arXiv:2301.08881.

[14] D. Gao, H. Wang, Y. Li, et al., "Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation," arXiv:2308.15363, 2023.

[15] X. Wang, Z. Wang, J. Liu, et al., "MINT: Evaluating LLMs in Multi-turn Interaction with Tools and Language Feedback," in Proc. ICLR, 2024. arXiv:2309.10691.
[16] M. Pourreza and D. Rafiei, "DIN-SQL: Decomposed In-Context Learning of Text-to-SQL with Self-Correction," in Proc. NeurIPS, 2023. arXiv:2304.11015.
[17] Y. Lai, C. Li, Y. Wang, et al., "DS-1000: A Natural and Reliable Benchmark for Data Science Code Generation," in Proc. ICML, 2023. arXiv:2211.11501.
[18] L. Huang, W. Yu, W. Ma, et al., "A Survey on Hallucination in Large Language Models," arXiv:2311.05232, 2023.