STRUCTURED COMPOSITION OF VERIFIABLE ATOMIC INSIGHTS FOR TABLE-TO-REPORT GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Table-to-report generation refers to the task of automatically generating article-level analytical reports from relational tables and is an essential capability for automated data science and decision support. Its central challenge lies in systematically discovering verifiable composite insights across tables, attributes, and analytical perspectives, and organizing them into coherent, complete, and traceable evidence chains. Existing methods primarily rely on sequential, reactive data agents or direct Large Language Model(LLM) generation. They suffer from exploration bias: early local observations constrain subsequent actions, causing models to focus prematurely on local analyzes and miss cross-table or cross-dimensional evidence. We propose ComInsight, which reformulates insight discovery as the composition of atomic evidences. We first define an atomic insight as the smallest executable analytical unit conforming to a predefined analysis pattern and enumerate all valid atomic insights from database schema and content. These atoms are then organized into a multi-relational insight graph, where nodes represent verified data facts and edges encode logical, temporal, or hierarchical relations. Finally, a set of composition operators systematically fuses atomic nodes into higher-order composite conclusions. Every composite output is accompanied by executable SQL and fine-grained provenance, ensuring full verifiability. Across three benchmarks InsightBench, DDR-Bench, and T2R-Bench, ComInsight consistently outperforms strong baselines in factual correctness, novelty, and structural completeness. We believe ComInsight offers a reliable, efficient, and explainable path toward table-to-report generation. Source code is available at https://anonymous.4open.science/r/ci-report.

1 INTRODUCTION

Table-to-report generation is the task of automatically synthesizing article-level analytical reports from relational tables, and it constitutes a fundamental capability for automated data science and decision support (Zhang et al., 2025b; Lin et al., 2026). Its central challenge lies not only in generating fluent text but in systematically discovering valuable insights across tables, attributes, and analytical perspectives, and organizing them into coherent, complete, and traceable evidence chains (Liu et al., 2026; Huang et al., 2026).

An individual finding often captures only one aspect of the data, whereas a comprehensive analysis requires connecting multiple findings to reveal relationships that are not apparent from any single finding alone. We refer to such higher-order findings, supported by multiple atomic findings and explicit relationships among them, as **composite insights**. Consider a retail analytics scenario as illustrated in Figure 1. The database contains three tables: sales (order_id, region, product_category, age_group, quarter, revenue, profit), survey (survey_id, region, quarter, satisfaction_score, nps_score), and logistics (shipment_id, region, quarter, delivery_days, delay_flag). A valid analysis should produce a composite insight such as: *"Driven by the 18-30 age group, East China leads in both revenue and satisfaction; in contrast, North China exhibits similar satisfaction but lags significantly in revenue growth, likely because its logistics delay rate (38% vs.*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

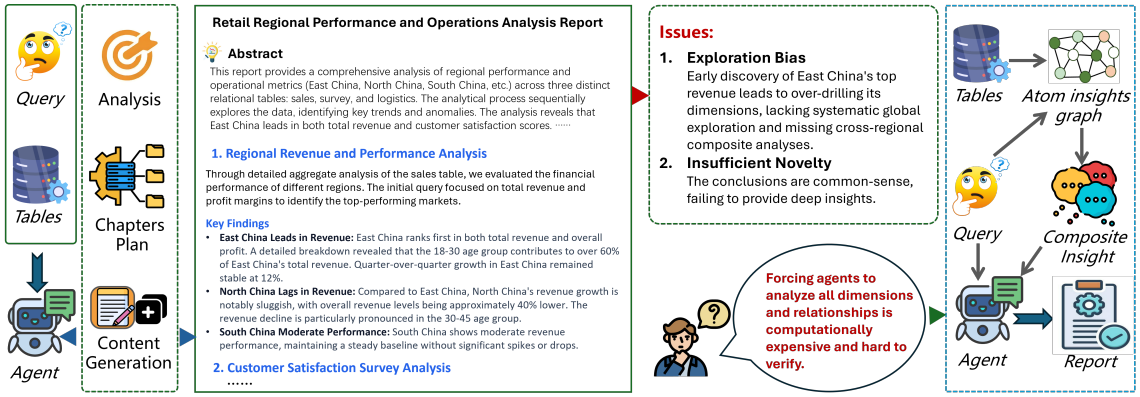

Figure 1: Existing agent workflows suffer from exploration bias and lack of novelty. Exhaustive dimension analysis is unverifiable. The ComInsight framework shifts from sequential exploration to structured composition, using atom insight graphs to generate verifiable composite insights.

East China's 12%) constrains revenue conversion." This conclusion integrates five independent evidence chains drawn from three different tables and involves four distinct analysis patterns (MAX/GROUP_BY, AVG/GROUP_BY, GROUP_BY with filter, and CORR).

Yet existing data agents struggle to produce such insights (Liu et al., 2026; Sahu et al., 2025; Ma et al., 2023). Sequential exploration agents, such as ReAct-style agents(e.g., AgentPoirot) (Sahu et al., 2025), after discovering that "East China has the highest revenue" in early interactions, tend to deepen along this direction, repeatedly querying further details about East China, rather than proactively crossing over to the survey table to examine the satisfaction dimension, let alone further linking to the logistics table's delay rate. DDR-Bench's failure analysis shows that 58% of errors stem from insufficient exploration breadth rather than single-step execution errors (Liu et al., 2026). These failures expose a fundamental limitation: current data agents lack the capability for systematic discovery of composite insights. To address this problem, we propose ComInsight, an evidence-driven framework for composite insight discovery in table-to-report generation. ComInsight follows three steps: it first systematically enumerates all valid atomic insights from the database according to a predefined analysis grammar; it then organizes these insights into a typed evidence graph, where nodes represent verified data facts and edges encode relations; finally, relation-specific composition operators fuse independent atomic insights into higher-order composite conclusions. Every composite output is accompanied by executable SQL, ensuring full verifiability. Our main contributions are as follows:

- **We formalize the problem of composite insight discovery for table-to-report generation** and propose five structural composite types: **comparative, correlative, contextual, temporal, and causal**. Each structural composite type is equipped with explicit composition conditions and reasoning semantics.
- **We propose the ComInsight framework**, comprising three core modules: atomic insight generation, multi-relational graph construction, and composite insight generation, along with a Query Sematic Filter that tailors outputs to user-specific analytical goals.
- **We conduct a systematic evaluation on three benchmarks**: InsightBench, DDR-Bench, and T2R-Bench. Experimental results show that ComInsight significantly outperforms existing baselines in factual correctness, insight novelty, and structural completeness.

2 Related Work

Table Understanding and Reasoning Benchmarks. Early TableQA benchmarks such as WikiTableQuestions (Pasupat & Liang, 2015), HybridQA (Chen et al., 2020), FeTaQA (Nan et al., 2022), and TAT-QA (Zhu et al., 2021) laid the foundation for question answering over tabular data. More recent comprehensive benchmarks include TableBench (Wu et al., 2025), MMTU (Xing et al., 2026), SciTableQA (Ajayi et al., 2025), SciNumQA (Son et al., 2026), TopBench (Ji et al., 2026), and M3TQA (Zha et al., 2024), which extend to complex layouts, multi-table reasoning, multilingual settings, and predictive queries. For table-to-text generation, ToTTo (Parikh et al., 2020) and FreshTab (Onderková et al., 2025) are representative efforts. Despite this diversity, all these benchmarks focus on atomic tasks, answering a single question or generating a single description, and do not address the combination of multiple findings into higher-level narratives.

Data Analysis Agents and Insight Generation. LLM-based agents have spurred benchmarks for end-to-end data analysis requiring multi-step reasoning and tool use, like InfiAgent-DABench (Hu et al., 2024), DataSciBench (Zhang et al., 2025a), UniDataBench (Weng et al., 2026), and WebDS (Hsu et al., 2025). For insight generation, InsightBench (Sahu et al., 2025), InsightEval (Zhu et al., 2026) and MedInsightBench (Zhu et al., 2025) evaluate agents on producing analytical insights, while DDR-Bench (Liu et al., 2026), MMDR-Bench (Huang et al., 2026), and T2R-Bench (Zhang et al., 2025b) target open-ended deep research and article-level report generation. Evaluation has shifted from moved beyond traditional metrics (e.g., BLEU, ROUGE, and BERTScore) toward LLM-as-a-Judge. Systems such as DataShot (Wang et al., 2019), InsightPilot (Ma et al., 2023), InsightLens (Weng et al., 2024), AgentPoirot (Sahu et al., 2025), ReActInsight (Weng et al., 2026), DataCross (Qi et al., 2026), and DABstep (Egg et al., 2025) demonstrate strong exploration and single-insight discovery, but none explicitly composes high-order insights into analytical reports.

3 Preliminaries and Problem Formulation

Table-to-Report Task. We consider a relational database $\mathcal{D}$ consisting of a set of tables $\mathcal{T} = \{T_1, \ldots, T_n\}$, where tables are potentially linked through primary-foreign key or semantic associations. Given $\mathcal{D}$, the task of table-to-report generation is to automatically produce an article-level analytical report $\mathcal{R}$ that summarizes the most important, non-redundant, and verifiable findings in the database. The user query (or report topic) q is optional depending on the specific task. The core problem of table-to-report generation is: *how to define and achieve a comprehensive investigation process that enables a model to discover, compose, and report as many valuable insights in the database as possible, while ensuring traceability, factual correctness, and non-redundancy for each insight?*

Atomic Insight. An atomic insight is defined as an nine-tuple

$$a = (id, src, type, dims, metrics, metadata, sql, result, description),$$

where $type \in \mathcal{T} = \{\text{MAX}, \text{MIN}, \text{AVG}, \text{SUM}, \text{COUNT}, \text{GROUP_BY}, \text{CORR}, \text{TREND}\}$, $\mathcal{T}$ is the set of pattern types. Pattern type is a parameterizable data query template that encapsulates a single data exploration logic.The Patterns and SQL Templates are in Appendix Table 3. $dims$ and $metrics$ specify the analytical context; sql is the executable query; $result$ is the returned value; and $description$ is a natural language statement of the finding.

Composite Insight. An composite insight c is a six-tuple

$$c = (id, type, atoms, summary, confidence, evidence),$$

where $atoms$ is the set of constituent atomic insights with $|atoms| \geq 2$, $type \in \{\text{Comparative}, \text{Correlative}, \text{Contextual}, \text{Temporal}, \text{Causal}\}$, $summary$ is a natural language statement, $evidence$ contains

the SQL queries and results, and $confidence$ measures its credibility in terms of statistical significance and logical consistency. Formally, let $\mathcal{A}$ be the set of all possible atomic insights , and $\mathcal{C}$ be the set of all valid composite insights.

Research Goals. The objective is to construct a report $\mathcal{R}$ such that: **Traceability:** every insight contains a complete evidence chain; **Factual Correctness:** every insight is supported by verifiable atomic insights and executable SQL; **Non-Redundancy:** $\mathcal{R}$ does not contain duplicate or semantically equivalent insights.

4 THE COMINSIGHT FRAMEWORK

4.1 ATOMIC INSIGHT REPRESENTATION AND GENERATION

The generation pipeline is illustrated in Figure 2. Atomic insights are generated through **Column Semantic Recognition, Pattern template Instantiation, Interestingness Filter**(statistical significance filter and LLM-based semantic filter), and **Semantic Deduplication**, yielding a non-redundant set $\mathcal{A}$ of atomic insight units. If the query or topic is given, we apply a **Query Sematic Filter** based on embedding relevance to select top-50 atomic insights before Interestingness Filter. The detailed definition and implementation steps of atomic insights are provided in the Appendix B.1.

Training Data Construction. We build training data by selecting a subset of instances from three benchmarks: 40 datasets from InsightBench (out of 100 instances), 100 tasks from DDR-Bench (out of 291), and 150 industrial tables from T2R-Bench (out of 457), totaling 290 instances. These instances serve two purposes: (i) calibrating the statistical significance filtering rules and thresholds, and (ii) training the lightweight LLM-as-Judge. Experts then annotate their semantic interestingness, yielding 1,200 (labeled interesting / not interesting) samples for supervised fine-tuning (SFT) and 600 preference pairs (Triplet (inputs, chosen, rejected)) for direct preference optimization (DPO).

Statistical significance rules and Interestingness Judge. We first classify training insights into eight pattern types, parse their SQL to extract metrics, dimensions, aggregations, and columns, then select the corresponding statistical test for each type and calibrate thresholds on a validation set by F1, yielding the final significance rules.(e.g., for CORR, retain candidates whose absolute Pearson correlation coefficient $|r|$ exceeds 0.6 and whose significance level p is below 0.05), and rules for the eight types are in Appendix B.1.2. A lightweight LLM-as-Judge (Llama-3-8B), trained via SFT on labeled samples and DPO on 600 preference pairs, evaluates semantic interestingness. In our experiments, the Interestingness Filter module filters about **85%** of atomic insights while preserving high-value discoveries.

4.2 MULTI-RELATIONAL GRAPH CONSTRUCTION

Atomic insights are modeled as nodes in a **multi-relational graph** $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ where nodes $\mathcal{V}$ correspond to atomic insights, $\mathcal{R} = \{\text{Comparative}, \text{Correlative}, \text{Contextual}, \text{Temporal}, \text{Causal}\}$ denotes the five relation types, and $\mathcal{E}$ is the set of typed edges. **The first four relations are treated as undirected edges**. As illustrated in Figure 3-1, each edge is equipped with a type-specific weight that quantifies the strength of the relation. Comparative edges connect same-metric/aggregation atoms with different dimension values, weighted by normalized relative difference and retained if it > 0.15. Correlative edges connect statistically dependent or co-moving trend atoms, and retained if the weight > 0.6. Contextual edges link global aggregates with local segments, weighted by slope or magnitude deviation with sign reversal weighted double. Temporal edges connect adjacent segmented trends partitioned by PELT, weighted by absolute slope difference with a minimum segment length of three. As illustrated in Figure 3-2, **Causal edges are directed** and are established only when two conditions are satisfied simultaneously: (1) a **Granger causality test**

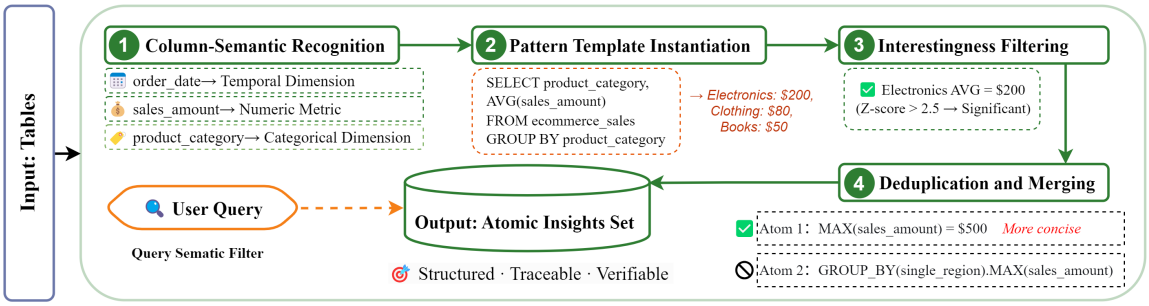

Figure 2: The pipeline of Atomic Insight Generation.

on the corresponding time series of the two atomic insights is statistically significant, i.e., the p-value is below 0.05 after appropriate lag selection; (2) an **LLM-based causal verifier** judges that a plausible causal relation exists, based on the descriptions, SQL results, and domain context, and assigns a mechanism score $s_{\text{mech}} \in [0, 1]$ as well as a confounding-risk score $c_{ij} \in [0, 1]$. Temporal precedence $t_i < t_j$ is required as a prerequisite for the Granger test. The weight of a directed causal edge $i \to j$ is

$$w_{\text{causal}}(i, j) = \mathbb{I}[t_i < t_j] \cdot \mathbb{I}[p_{\text{Granger}} < 0.05] \cdot |r_{ij}|(1 - p_{ij}) \cdot s_{\text{mech}}(i, j) \cdot (1 - c_{ij}), \tag{1}$$

where p_{Granger} is the p-value from the Granger causality test, and r_{ij}, p_{ij} are the adjusted correlation coefficient and its p-value. A directed edge $i \to j$ is retained only if $w_{\text{causal}}(i, j) > 0.6$. This dual requirement ensures that causal edges are both statistically supported and semantically plausible.

The graph is implemented as a MultiGraph, allowing multiple parallel edges between the same node pair to capture different relational facets. The detailed edge weight computation is shown in Appendix B.2.

4.3 Composite Insight Generation

Community-based Construction. We perform community detection on each of the four undirected relation subgraphs. For each relation type, we run the Leiden algorithm on the subgraph to partition atomic insights into communities. Communities with modularity greater than 0.3 and size at least 3 are retained. For each retained community, we prompt an LLM to synthesize a higher-order composite insight that integrates all atomic insights in the community, producing a coherent narrative that reflects the underlying relational structure. The LLM receives the descriptions, types, and SQL results of all constituent atoms, and is instructed to generate a summary that captures the shared theme and relational pattern. The steps and prompts are in Appendix B.4.

Causal Insights Construction. For causal relations, we search directed causal paths of length up to $L_{\max}$ (e.g., 3). Each path is aggregated by an LLM into a causal composite insight that states the causal chain and includes the Granger p-value, mechanism score, and confounding risk. Causal composites are treated as hypotheses and are retained only if every causal edge on the path has been validated by both the Granger test and the LLM judge.

After initial Composite Insight Generation, an LLM-based **Composite Value Evaluator** compares each composite against its constituent atomic insights to determine retain the atoms or composite, which is shown in Figure 3-3. So the final insights set includes generated composite insights and retain atomic insights. More detailed definition and implementation steps of composite insights are provided in the Appendix B.3.

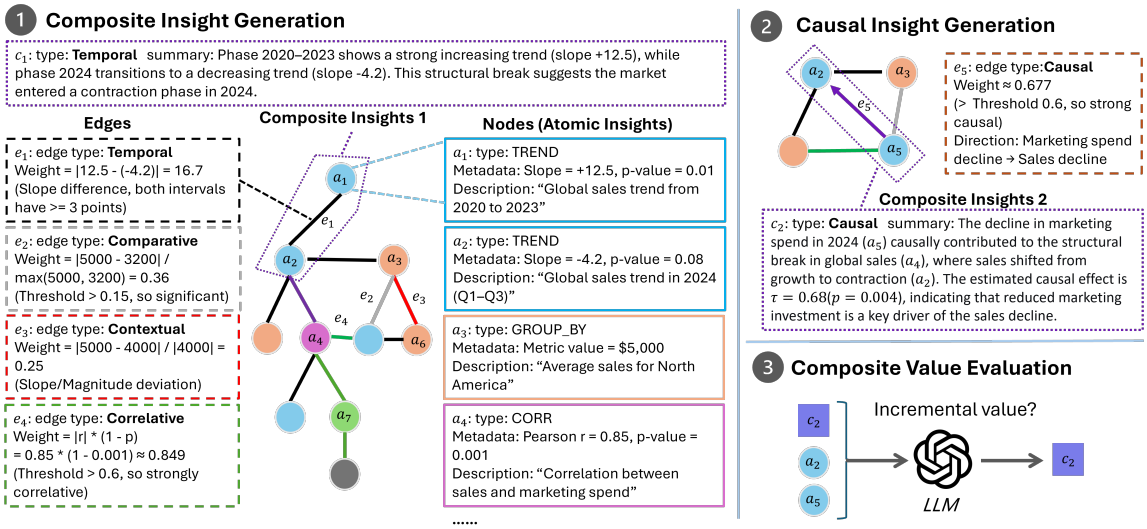

Figure 3: 1. Diagram of the multi-relational graph constructed. Each node represents an atomic insight. Edges between nodes are established according to five relation types. 2. Diagram of Causal insights generation. 3. Composite Value Evaluator.

4.4 Report Generation

Given the set of insights and an optional user query(topic) q, the report generation module first removes insights whose embedding similarity to q falls below a threshold (or keeps all if q is absent). Then we group related insights into thematic sections and uses an LLM to generate coherent narratives that cover all retained insights with every claim backed by SQL evidence. The final report is assembled from these sections and remains fully traceable and verifiable. Details is in Appendix B.5.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate ComInsight on three benchmarks's test sets: **InsightBench**: Consists of 100 datasets with 475 insights. **DDR-Bench** (Deep Data Research Benchmark): Contains 291 tasks and 2058 insights. **T2R-Bench**: Contains 910 questions on 457 real-world industrial tables from 19 industry domains. The LLM-as-Judge is trained on the 290 training instances listed in Section 4.1. All training instances are excluded from test. Results are reported on held-out test splits.

Baselines. **Pandas Agent** (Chase, 2022): A data science agent optimized for single-turn question answering. Given a dataframe and a question, it generates and executes Python code to produce the answer. **AgentPoirot**: The baseline data analysis agent proposed in InsightBench, capable of performing end-to-end data analytics by generating Descriptive, Diagnostic, Predictive, and Prescriptive questions. **DeepSeek-R1** (Guo et al., 2025): The best-performing model on T2R-Bench, achieving an overall score of 62.71%. **Claude 4.5 Sonnet** (Anthropic, 2025): The best-performing proprietary model on DDR-Bench and **GPT-5.5**.

Table 1: Performance comparison on InsightBench. I stands for Insight-level, S is Summary-level, and Eval is LLaMA-3-Eval. **Bold** indicates the best metric, Underline indicates the second best.

Method	I-Eval	S-Eval	I-ROUGE-1	S-ROUGE-1
Pandas Agent (gpt-5.5)	0.54 ± 0.01	0.40 ± 0.04	$\underline{0.35 \pm 0.03}$	0.35 ± 0.01
AgentPoirot (gpt-5.5)	$\underline{0.60 \pm 0.03}$	$\underline{0.44 \pm 0.03}$	0.32 ± 0.02	$\underline{0.37 \pm 0.09}$
DeepSeek-R1	0.47 ± 0.03	0.31 ± 0.02	0.29 ± 0.02	0.31 ± 0.01
Claude 4.5 Sonnet	0.45 ± 0.03	0.27 ± 0.02	0.23 ± 0.02	0.29 ± 0.05
GPT-5.5	0.51 ± 0.02	0.36 ± 0.03	0.31 ± 0.02	0.33 ± 0.04
ComInsight	$\mathbf{0.73 \pm 0.03}$	$\mathbf{0.55 \pm 0.03}$	$\mathbf{0.42 \pm 0.02}$	$\mathbf{0.43 \pm 0.03}$

Evaluation Metrics. We adopt benchmark-specific metrics: For InsightBench, LLaMA-3-Eval (a G-Eval variant based on LLaMA-3-70b) for insight-level and summary-level scores. For DDR-Bench, Checklist-based accuracy, where GPT-5-mini judges whether generated insights support each checklist fact. For T2R-Bench, there are three-dimensional evaluation: Numerical Accuracy Criterion (NAC), Information Coverage Criterion (ICC), and General Evaluation Criterion (GEC), and report their average score.

5.1.1 Implementation Details

ComInsight's atomic insight generation module is based on predefined analysis pattern templates. The multi-relational graph is implemented using NetworkX's MultiGraph structure. All experiments are conducted on 8 NVIDIA RTX4090 GPUs. The LLM used for composite insights generation and report generation is gpt-5.5 (OpenAI API). Sampling temperature is set to 0.0 for reproducibility, with each experiment repeated 5 times reporting average results.

5.2 Main Results

5.2.1 Performance on InsightBench

Table 1 presents the performance comparison on InsightBench. ComInsight significantly outperforms all baselines on both insight-level and summary-level LLaMA-3-Eval scores. ComInsight achieves an insight-level LLaMA-3-Eval score of 0.73, outperforming AgentPoirot (gpt-5.5) by 13.3 percentage points. This significant improvement demonstrates that explicit composite insight generation effectively integrates multiple atomic insights, producing richer and more accurate conclusions than single-step analysis. Meanwhile, GPT-5.5 achieves only 0.51, significantly below ComInsight's 0.73, demonstrating that simple linguistic concatenation cannot replace structured evidence combination reasoning.

5.2.2 Performance on DDR-Bench

Figure 4 presents the performance comparison on DDR-Bench across three scenarios. ComInsight achieves the best performance on MIMIC, GLOBEM and 10-K scenarios. On the MIMIC scenario, ComInsight achieves 41.23% accuracy, outperforming Claude 4.5 Sonnet by 14.3%. On the 10-K scenario, ComInsight reaches 79.34%, a 2.2% improvement over the native model. The improvement on GLOBEM is relatively modest (from 42.31% to 42.87%), likely because GLOBEM's checklist items are closed-form (three-option multiple choice), leaving less room for composite reasoning.

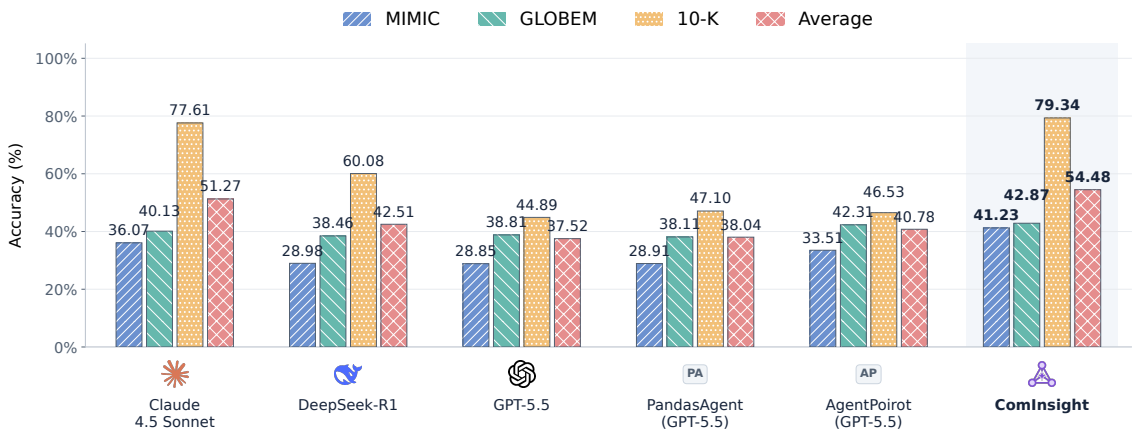

Figure 4: Performance comparison on DDR-Bench.

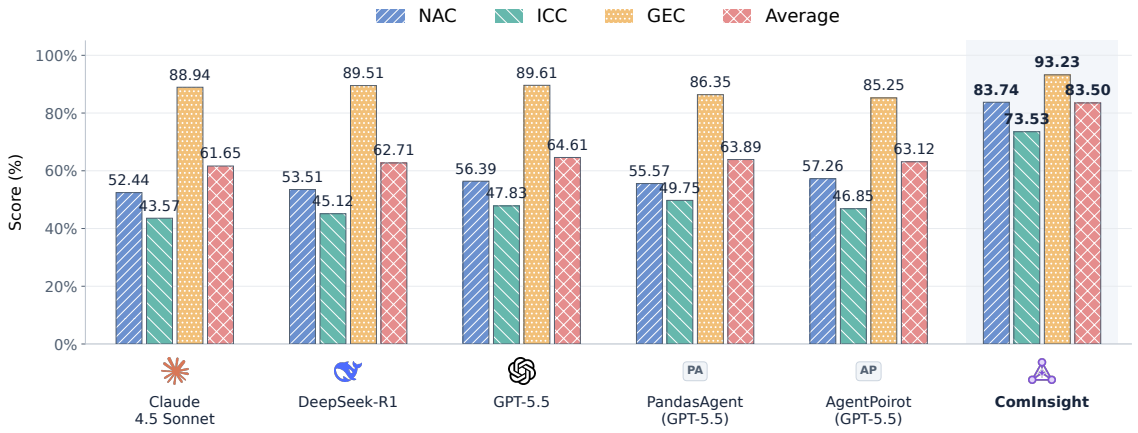

Figure 5: Performance comparison on T2R-Bench.

5.2.3 Performance on T2R-Bench

Figure 5 presents the performance comparison on T2R-Bench. ComInsight outperforms baselines across all three evaluation dimensions. ComInsight achieves an Average score of 83.5% on T2R-Bench, outperforming native GPT-5.5 (64.61%) by 18.89 percentage points. The improvement on Numerical Accuracy Criterion (NAC) is most significant (+ 26.48%), suggesting that explicit evidence chaining in composite insight generation helps reduce numerical hallucinations, the most common error type in T2R-Bench report generation. Information Coverage Criterion (ICC) improves from 49.75% to 73.73%, indicating that composite insights more comprehensively cover report keypoints. This finding aligns with ComInsight's design: through multi-relational graph traversal, the system systematically discovers associations between atomic insights, avoiding omissions of important information.

Table 2: Ablation study on T2R-Bench. ΔAverage is computed against full ComInsight.

Configuration	Average(%)	ΔAverage(%)
Full ComInsight	83.5	–
w/o Composite Insight Generation (Simple LLM Concatenation)	64.61	-18.89
w/o Composite Value Evaluation	71.34	-12.16
w/o Causal Relationship Generation	76.75	-6.75

5.3 Ablation Studies

To isolate the contribution of the three key stages in ComInsight, we conduct ablations on T2R-Bench under the same setting as the main experiment. The evaluated variants are: (1) **w/o Composite Insight Generation (Simple LLM Concatenation)**, which prompts an LLM(GPT-5.5) to concatenate atomic insights into a report; (2) **w/o Composite Value Evaluation**, which retains composite candidates but removes the LLM-based incremental-value judge and (3) **w/o Causal Relationship Generation**, which removes causal assessment and causal composite merging.

On T2R-Bench, full ComInsight achieves an Average of 83.5% ; removing composite insight generation and reverting to simple LLM concatenation reduces Average to 64.61% ($\Delta\text{Avg} = -18.89\%$), indicating that structured composition over the multi-relational graph is the dominant source of improvement; removing composite value evaluation yields 71.34%, showing that the incremental-value judge is important for suppressing redundant or low-utility composites and preserving coverage; and removing causal relationship generation gives 76.75%, suggesting that causal chains contribute less to T2R-Bench numerical accuracy and information coverage but still support reasoning depth and narrative completeness.

5.3.1 Efficiency and Time Analysis

Across the three benchmarks test set, ComInsight generates on average about 28 atomic insights, 142 typed edges, and 41 final composite insights per task. The end-to-end pipeline takes about 102.3 seconds: 14.2s for atomic enumeration and SQL execution, 17.6s for statistical filtering and semantic evaluation, 1.4s for graph construction, 52.8s for composite generation (51.6%, the main bottleneck), and 16.3s for report generation. T2R-Bench is the slowest due to larger tables and more atoms, while InsightBench is the fastest; batching LLM calls and parallelizing composite generation can reduce the time by 4–6×.

6 Conclusion

In conclusion, this paper addresses the exploration bias and limited novelty of existing data agents in table-to-report generation by formalizing composite insight discovery and proposing ComInsight. ComInsight generates verifiable atomic insights, organizes them into a typed multi-relational evidence graph, and applies relation-specific composition operators to fuse them into comparative, correlative, contextual, temporal, and causal composite insights, while query-driven intent alignment and executable SQL provenance ensure relevance, factual correctness, structural completeness, and traceability. Experiments on InsightBench, DDR-Bench, and T2R-Bench demonstrate that ComInsight substantially outperforms existing baselines in factual correctness, novelty, and structural completeness, and ablation studies confirm the critical roles of the multi-relational graph and specialized composition operators. This work thus offers a principled and verifiable structured-composition paradigm for automated, report-oriented data analysis.

7 AI USE STATEMENT

In this work, we used generative AI tools for required-disclosure tasks involved in implementing and evaluating the proposed method: a Llama-3-8B model was fine-tuned via SFT and DPO and used as an LLM-as-Judge for semantic interestingness evaluation; GPT-5.5 was used for composite insight generation and report generation; and LLaMA-3-70B and GPT-5-mini were used as automatic evaluators for benchmark experiments. These uses are part of the ComInsight methodology and evaluation pipeline and are described in Sections 4 and 5. We have not used generative AI tools for generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing research methodology or experiments, cleaning or reformatting datasets, supporting qualitative/thematic data analysis, or interpreting results beyond the LLM-based components explicitly described above. We have reviewed all AI-assisted work. For example, all LLM-generated outputs used as evidence were checked against executable SQL and experimental results, and all references were verified manually. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Kehinde Ajayi, Yi He, Matthew Maisonave, Kris SeekFord, and Jian Wu. Scitableqa: A question-answering benchmark for complex scientific tables. In *International Conference on Theory and Practice of Digital Libraries*, pp. 90–107. Springer, 2025.

Anthropic. Claude 4.5. Anthropic Blog, 2025. URL https://www.anthropic.com/news/claude-sonnet-4-5. Introducing Claude Sonnet 4.5, released 29 Sep 2025.

Harrison Chase. LangChain, October 2022. URL https://github.com/langchain-ai/langchain.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In Trevor C ohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1026–1036, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.91. URL https://aclanthology.org/2020.findings-emnlp.91/.

Alex Egg, Martin Iglesias Goyanes, Friso Kingma, Andreu Mora, Leandro von Werra, and Thomas Wolf. Dabstep: Data agent benchmark for multi-step reasoning. *arXiv preprint arXiv:2506.23719*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Ethan Hsu, Hong Meng Yam, Ines Bouissou, Aaron Murali John, Raj Thota, Josh Koe, Vivek Sarath Putta, GK Dharesan, Alexander Spangher, Shikhar Murty, et al. Webds: An end-to-end benchmark for web-based data science. *arXiv preprint arXiv:2508.01222*, 2025.

Xueyu Hu, Ziyu Zhao, Shuang Wei, Ziwei Chai, Qianli Ma, Guoyin Wang, Xuwu Wang, Jing Su, Jingjing Xu, Ming Zhu, et al. Infiagent-dabench: Evaluating agents on data analysis tasks. *arXiv preprint arXiv:2401.05507*, 2024.

Peizhou Huang, Zixuan Zhong, Zhongwei Wan, Donghao Zhou, Samiul Alam, Xin Wang, Zexin Li, Zhihao Dou, Li Zhu, Jing Xiong, Chaofan Tao, Yan Xu, Dimitrios Dimitriadis, Tuo Zhang, and Mi Zhang. Mmdeepresearch-bench: A benchmark for multimodal deep research agents, 2026. URL https://arxiv.org/abs/2601.12346.

An-Yang Ji, Jun-Peng Jiang, De-Chuan Zhan, and Han-Jia Ye. Topbench: A benchmark for implicit prediction and reasoning over tabular question answering. *arXiv preprint arXiv:2604.28076*, 2026.

Teng Lin, Zhiyang Zhang, Yuyu Luo, and Nan Tang. Monte carlo tree search for table-to-multimodal report generation, 2026. URL https://arxiv.org/abs/2608.04071.

Wei Liu, Peijie Yu, Michele Orini, Yali Du, and Yulan He. Hunt instead of wait: Evaluating deep data research on large language models, 2026. URL https://arxiv.org/abs/2602.02039.

Pingchuan Ma, Rui Ding, Shuai Wang, Shi Han, and Dongmei Zhang. Insightpilot: An llm-empowered automated data exploration system. In *Proceedings of the 2023 conference on empirical methods in natural language processing: system demonstrations*, pp. 346–352, 2023.

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, et al. Fetaqa: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49, 2022.

Kristỳna Onderková, Ondřej Plátek, Zdeněk Kasner, and Ondřej Dušek. Freshtab: Sourcing fresh data for table-to-text generation evaluation. In *Proceedings of the 18th International Natural Language Generation Conference*, pp. 108–121, 2025.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. Totto: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1173–1186, 2020.

Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1470–1480, 2015.

Ruyi Qi, Zhou Liu, and Wentao Zhang. Datacross: A unified benchmark and agent framework for cross-modal heterogeneous data analysis. *arXiv preprint arXiv:2601.21403*, 2026.

Gaurav Sahu, Abhay Puri, Juan A Rodriguez, Amirhossein Abaskohi, Mohammad Chegini, Alexandre Drouin, Perouz Taslakian, Valentina Zantedeschi, Alexandre Lacoste, David Vazquez, et al. Insightbench: Evaluating business analytics agents through multi-step insight generation. In *International Conference on Learning Representations*, volume 2025, pp. 4683–4715, 2025.

Sohyun Son, Yubin Kim, Gangwoo Kim, Donghee Choi, and Mujeen Sung. Scinumqa: A benchmark for complex numerical reasoning over scientific tables. *IEEE Access*, 2026.

Yun Wang, Zhida Sun, Haidong Zhang, Weiwei Cui, Ke Xu, Xiaojuan Ma, and Dongmei Zhang. Datashot: Automatic generation of fact sheets from tabular data. *IEEE transactions on visualization and computer graphics*, 26(1):895–905, 2019.

Han Weng, Zhou Liu, Yuanfeng Song, Xiaoming Yin, Xing Chen, and Wentao Zhang. UniDataBench: Evaluating data analytics agents across structured and unstructured data. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 33755–33780, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1556. URL https://aclanthology.org/2026.acl-long.1556/.

Luoxuan Weng, Xingbo Wang, Junyu Lu, Yingchaojie Feng, Yihan Liu, Haozhe Feng, Danqing Huang, and Wei Chen. Insightlens: Augmenting llm-powered data analysis with interactive insight management and navigation. *arXiv preprint arXiv:2404.01644*, 2024.

Xianjie Wu, Jian Yang, Linzheng Chai, Ge Zhang, Jiaheng Liu, Xeron Du, Di Liang, Daixin Shu, Xianfu Cheng, Tianzhen Sun, et al. Tablebench: A comprehensive and complex benchmark for table question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25497–25506, 2025.

Junjie Xing, Yeye He, Mengyu Zhou, Haoyu Dong, Shi Han, Lingjiao Chen, Dongmei Zhang, Surajit Chaudhuri, and HV Jagadish. Mmtu: A massive multi-task table understanding and reasoning benchmark. *Advances in Neural Information Processing Systems*, 38, 2026.

Zhiyuan Zha, Pengnian Qi, Xigang Bao, Mengyuan Tian, and Biao Qin. M 3 tqa: Multi-view, multi-hop and multi-stage reasoning for temporal question answering. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 10086–10090. IEEE, 2024.

Dan Zhang, Sining Zhoubian, Min Cai, Fengzu Li, Lekang Yang, Wei Wang, Tianjiao Dong, Ziniu Hu, Jie Tang, and Yisong Yue. Datascibench: An llm agent benchmark for data science. *arXiv preprint arXiv:2502.13897*, 2025a.

Jie Zhang, Changzai Pan, Sishi Xiong, Kaiwen Wei, Yu Zhao, Xiangyu Li, Jiaxin Peng, Xiaoyan Gu, Jian Yang, Wenhan Chang, Zhenhe Wu, Jiang Zhong, Shuangyong Song, and Xuelong Li. T2R-BENCH: A benchmark for real world table-to-report task. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 22427–22451, Suzhou, China, November 2025b. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1141. URL https://aclanthology.org/2025.emnlp-main.1141/.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *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)*, pp. 3277–3287, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.254. URL https://aclanthology.org/2021.acl-long.254/.

Zhenghao Zhu, Chuxue Cao, Sirui Han, Yuanfeng Song, Xing Chen, Caleb Chen Cao, and Yike Guo. Medinsightbench: Evaluating medical analytics agents through multi-step insight discovery in multimodal medical data. *arXiv preprint arXiv:2512.13297*, 2025.

Zhenghao Zhu, Yuanfeng Song, Xing Chen, Chengzhong Liu, Cui Yakun, Caleb Chen Cao, Sirui Han, and Yike Guo. InsightEval: An expert-curated benchmark for assessing insight discovery in LLM-driven data agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 34632–34656, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1729. URL https://aclanthology.org/2026.findings-acl.1729/.

A Appendices

B The ComInsight Framework

ComInsight takes relational tables as input and produces a set of high-quality composite insights as output. Its design follows a four-stage progression: *atomic discovery* → *relational modeling* → *structured composition* → *causal verification*. The framework consists of three core modules and two auxiliary layers.

- **Atomic Insight Generation Module.** It performs multi-dimensional analysis pattern enumeration over input tables, followed by statistical salience filtering, semantic interestingness evaluation, and semantic deduplication. The output is a set of high-quality atomic insights that serve as elementary evidence units for composition.
- **Multi-Relational Graph Construction Module.** It models atomic insights as nodes and computes type-specific edge weights for four structural relation types: comparative, correlative, contextual, and temporal. The resulting typed evidence graph provides an interpretable search space for composite discovery.
- **Composite Insight Generation Module.** It performs edge-based traversal and type-specific composition, followed by causal verification, deduplication, and confidence scoring. It outputs five categories of composite insights: comparative, correlative, contextual, temporal, and causal.

In addition, a **Query-driven Intent Alignment Layer** sits between atomic generation and graph construction. Given a user query, it filters the atomic insight pool by semantic relevance, so that subsequent composition focuses on directions genuinely relevant to the user. A **Semantic Interestingness Judge** and a **Causal Verifier** are integrated into the atomic and composite stages, respectively.

B.1 Atomic Insight Representation and Generation

B.1.1 Definition of Atomic Insight

An atomic insight is the smallest executable analytical unit conforming to a predefined analysis grammar. Formally, an atomic insight a is an nine-tuple:

$$a = (id, src, type, dims, metrics, metadata, sql, result, description), \tag{2}$$

where:

- id: a globally unique identifier;
- src: the data source, e.g., a database table or a cross-table view;
- $type \in \mathcal{T}$: the analysis pattern type, where $\mathcal{T} = \{\text{MAX}, \text{MIN}, \text{AVG}, \text{SUM}, \text{COUNT}, \text{GROUP_BY}, \text{CORR}, \text{TREND}\}$;
- $dims$: the set of dimension columns and their filter values, e.g., $\{\text{region} : \text{East}, \text{quarter} : \text{Q1}\}$;
- $metrics$: the set of metric columns and their aggregations, e.g., $\{\text{revenue} : \text{SUM}\}$;
- $metadata$: structured numerical evidence, e.g., "slope $= +12.5$, p-value $= 0.01$";
- sql: the executable query used to reproduce and verify the insight;
- $result$: the numeric or tabular result returned by the SQL query.

Let $\mathcal{A}$ denote the space of all atomic insights. However, not every generated candidate is retained. An atomic insight is included in the final evidence pool only if it passes both statistical salience and semantic interestingness checks.

B.1.2 Generation Pipeline for Atomic Insights

The atomic insight generation module takes arbitrary relational tables as input and produces a curated set of atomic findings through five stages.

Training Data Construction. We build training data by selecting a subset of instances from three benchmarks: 40 datasets from InsightBench (out of 100 instances), 100 tasks from DDR-Bench (out of 291), and 150 industrial tables from T2R-Bench (out of 457), totaling 290 instances. These instances serve two purposes: (i) calibrating the statistical salience filtering rules and thresholds, and (ii) training the lightweight LLM-as-Judge.

Specifically, from these 290 instances we automatically enumerate approximately 3,200 candidate atomic insights. Domain experts then annotate their semantic interestingness, yielding 1,200 supervised fine-tuning (SFT) samples and 600 preference pairs for direct preference optimization (DPO).

Stage 1: Column Semantic Recognition. The system classifies each column into three semantic categories:

- **Dimension Column:** low-cardinality categorical columns (cardinality ≤ 50), or numerical columns with a small value set that semantically serves as grouping criteria, e.g., "region";
- **Metric Column:** high-cardinality numerical columns suitable for aggregation, e.g., SUM, AVG, MAX/MIN;
- **Time Column:** datetime/timestamp columns, or columns whose names contain time-related keywords such as "date", "time", "year", "quarter", or "month".

The classification combines column statistics and naming heuristics.

Stage 2: Pattern Template Instantiation. For each combination of metric columns and dimension columns, the system instantiates parameterized SQL templates for the analysis patterns in $\mathcal{T}$. This stage enumerates all reasonable dimension–metric–pattern combinations and yields a set of raw candidate atomic insights.

Stage 3: Statistical Significance Filtering. We calibrate the significance thresholds for each analysis pattern by processing the insights in the training dataset as follows:

1. **Type classification**: Classify each insight in the training set into one of eight types according to its analysis pattern: MAX, MIN, AVG, SUM, COUNT, GROUP_BY, CORR, TREND.
2. **SQL parsing**: Parse the SQL query of each insight to extract the metric columns, dimension columns, aggregation functions, grouping conditions, time columns, returned values, and metadata.
3. **Statistical test selection**: Choose the corresponding statistical test based on the type:
   - CORR: compute the Pearson correlation coefficient r and its significance p-value;
   - TREND: fit a linear regression of the metric over time to obtain the slope and its significance p-value;
   - GROUP_BY: compute the standardized deviation of each group aggregate from the global mean;
   - MAX/MIN: identify extrema;
   - AVG, SUM, COUNT: compute distributional statistics for subsequent thresholding.
4. **Threshold determination**: On a validation set, select the threshold that optimizes F1 or agreement with expert judgments. The final rules are:
   - **MAX / MIN**: Extrema are inherently interesting; retain all extrema without additional thresholds, i.e., $S_{\text{stat}}(a) = 1$.
   - **AVG**: If it is a global average, retain it directly. If grouped by a dimension, compute the standardized deviation of each group mean from the global mean, $z = |\bar{x}_g - \mu|/(\sigma/\sqrt{n})$, and retain groups with $|z| > 2$.

- **SUM**: If it is a global sum, retain it directly. If grouped, compute the standardized deviation of each group sum from the global sum, or its proportion, and retain groups whose deviation exceeds two standard errors or whose proportion exceeds the 90th percentile.
- **COUNT**: If it is a global count, retain it directly. If grouped, use a binomial test or chi-square test and retain groups with $p < 0.05$.
- **GROUP_BY**: Retain groups whose aggregate deviates from the global mean by more than two standard errors, with multiple-comparison correction.
- **CORR**: Retain candidates with $|r| > 0.6$ and $p < 0.05$.
- **TREND**: Retain candidates with regression $p < 0.05$.

We define a binary statistical salience indicator $S_{\text{stat}}(a) \in \{0, 1\}$, where $S_{\text{stat}}(a) = 1$ if atomic insight a passes the rule for its corresponding type, and 0 otherwise.

Stage 4: Semantic Interestingness Evaluation. Statistical significance alone does not guarantee that an atomic insight is semantically interesting, novel, or actionable. We therefore train a lightweight **LLM-as-Judge** to score the semantic interestingness of each candidate.

We use Llama-3-8B as the judge. The training data come from the 290 instances described above: candidate atomic insights are generated from these instances, and experts annotate their interestingness, forming 1,200 SFT samples and 600 DPO preference pairs.

Base Configuration and LoRA Setup. We adopt QLoRA for parameter-efficient fine-tuning with 4-bit NF4 quantization and bf16 computation. The LoRA configuration is: rank $r = 16$, $\alpha = 16$, dropout $= 0.05$, with target modules `q_proj`, `k_proj`, `v_proj`, `o_proj`. The optimizer is AdamW with weight decay 0.05, and the learning rate schedule is cosine decay with a warmup ratio of 0.1. All experiments are conducted on 8 NVIDIA RTX 4090 GPUs with sampling temperature set to 0.0 for reproducibility.

SFT Stage. We perform supervised fine-tuning on 1,200 expert-annotated samples for 3 epochs. The learning rate is 2×10^{-5}, per-device batch size is 4, gradient accumulation steps is 4, yielding an effective batch size of 128. The maximum sequence length is 2,048 tokens. Loss is computed only on response tokens, not on prompts.

DPO Stage. Starting from the SFT model, we apply DPO on 600 preference pairs for 1 epoch. The learning rate is 5×10^{-7}, per-device batch size is 2, gradient accumulation steps is 8, yielding an effective batch size of 16. The DPO temperature β is set to 0.1, and the maximum sequence length is 2,048 tokens. The number of warmup steps is 100. The training data are derived from 290 benchmark instances (40 from InsightBench, 100 from DDR-Bench, and 150 from T2R-Bench).

Stage 5: Deduplication and Merging. Different analysis patterns may produce semantically equivalent or near-redundant insights, e.g., a MAX result and a GROUP_BY extreme referring to the same fact. We normalize each atomic insight by the triple $(\text{description}, \text{metrics}, \text{type})$ and merge equivalent items, keeping the one with the most concise $description$.

B.1.3 Query Sematic Filter

Given a user query q and the atomic insight set $\mathcal{A}'$, the intent alignment layer computes a relevance score

$$r(q, a) = \text{sim}\big(\text{enc}(q), \text{enc}(a., \text{description} \oplus a.\text{metrics} \oplus a.\text{type})\big), \tag{3}$$

where $\text{enc}(\cdot)$ is a text embedding model, $\oplus$ denotes string concatenation, and $\text{sim}(\cdot, \cdot)$ is cosine similarity. The filtered candidate set is

$$\mathcal{A}'' = \{a \in \mathcal{A}' \mid r(q, a) > \tau\} \cup \{a \in \mathcal{A}' \mid \text{pattern}(a) \in \text{required_patterns}(q)\}\,. \tag{4}$$

We keep top-50 atomic insights.

B.2 Multi-Relational Graph Construction

We organize the filtered atomic insights into a typed multi-relational graph

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R}), \tag{5}$$

where nodes $\mathcal{V} = \mathcal{A}''$ represent atomic insights, and $\mathcal{R} = \{\text{Comparative}, \text{Correlative}, \text{Contextual}, \text{Temporal}\}$ denotes four structural relation types. Edges $\mathcal{E}$ are typed and weighted. The graph is implemented as a MultiGraph, allowing multiple parallel edges between the same node pair to capture different relational facets.

Comparative Edge Weight. A comparative edge connects two atomic insights if they involve the same metric and aggregation but differ in dimension values. Its weight is the normalized relative difference:

$$w_{\text{comp}}(i,j) = \frac{|v_i - v_j|}{\max(v_i, v_j)}, \tag{6}$$

where v_i and v_j are the result values. We retain only edges with $w_{\text{comp}}(i,j) > 0.15$.

Correlative Edge Weight. A correlative edge connects atomic insights that exhibit statistical dependencies, including CORR-type insights and co-moving TREND-type insights. Its weight is

$$w_{\text{corr}}(i,j) = |r_{ij}| \cdot (1 - p_{ij}), \tag{7}$$

where r_{ij} is the Pearson correlation coefficient and p_{ij} is the corresponding p-value. Edges with $w_{\text{corr}}(i,j) > 0.6$ are considered strongly correlative.

Contextual Edge Weight. A contextual edge connects a global aggregate insight with a local segmented insight. Its weight quantifies the deviation of the local pattern from the global baseline:

$$w_{\text{context}}(i,j) = \max\left(\frac{|s_i - s_j|}{|s_j| + \epsilon},\ \mathbb{I}[\text{sign}(s_i) \neq \text{sign}(s_j)] \cdot 2\right), \tag{8}$$

where s_i and s_j are the trend slopes or magnitude values of the local and global insights, and ϵ is a small constant to prevent division by zero.

Temporal Edge Weight. A temporal edge connects segmented trend insights of the same metric across adjacent time intervals partitioned by the PELT change-point detection algorithm. Its weight is

$$w_{\text{temp}}(i,j) = |s_i - s_j| \cdot \mathbb{I}[\text{len}(T_i) \geq 3 \wedge \text{len}(T_j) \geq 3], \tag{9}$$

where s_i and s_j are the trend slopes of adjacent intervals, and $\text{len}(T)$ is the number of time points in the interval. The minimum segment length of three suppresses over-segmentation.

B.3 Composite Insight Generation

B.3.1 Definition of Composite Insight

We extend the composite insight taxonomy to five types:

$$\text{type}(c) \in \{\text{comparative}, \text{correlative}, \text{contextual}, \text{temporal}, \text{causal}\}. \tag{10}$$

A composite insight c is a six-tuple:

$$c = (id, \text{type}, \text{atoms}, \text{summary}, \text{confidence}, \text{evidence}), \tag{11}$$

where:

- id: a globally unique identifier;
- type: one of the five composite categories above;
- atoms $\subseteq \mathcal{A}''$: the set of constituent atomic insights, with $|\text{atoms}| \geq 2$;
- summary: a natural language summary;
- confidence $\in [0, 1]$: a confidence score;
- evidence: the evidence package containing the SQL queries and results of the supporting atoms.

In addition, for every generated composite, we compute a causal score $\text{Causal}(c) \in [0, 1]$. If $\text{Causal}(c) > \tau_{\text{causal}}$, the composite is marked as a *causal* composite insight.

B.4 Community-based Aggregation: Implementation Steps

1. **Subgraph construction.** For each of the four undirected relation types (Comparative, Correlative, Contextual, Temporal), extract a subgraph G_r containing only edges of that type. Nodes are atomic insights; edges are the corresponding relations.
2. **Community detection.** Run the Leiden algorithm on each subgraph G_r to obtain communities. Leiden guarantees internal connectivity and works well on weighted graphs.
3. **Community filtering.** Keep communities with modularity greater than 0.3 and size at least 3. Discard communities that are too small or loosely connected.
4. **Community summary generation.** For each retained community, call an LLM to generate a higher-order composite insight.

```
System: You are a strict reviewer.
User: Score the incremental value of the composite insight
relative to the atomic insights using the following five
criteria:
(a) Novel Synthesis (0--3): Does it form a new synthesized
    conclusion rather than listing atoms?
(b) Information Gain (0--2): Does it contain new
    information not directly stated but reasonably derivable
    from the atoms?
(c) Logical Coherence (0--2): Is it logically consistent
    and free of contradictions?
(d) Relational Explanation (0--2): Does it explain the
    relational pattern or mechanism among atoms?
(e) Decision Utility (0--1): Does it provide practical
    guidance for the analysis goal?
Output the score and a brief reason for each criterion,
then give the total score. If total ≥ 6, answer ``Yes'';
otherwise ``No''.
Atomic insights: {...}
Composite insight: {...}
```

B.4.1 Deduplication and Confidence Scoring

Generated composite candidates undergo two quality-control steps.

Deduplication. We deduplicate composites based on the set of constituent atomic insight IDs. If two composites have identical atomic ID sets, only the one with higher confidence is retained.

Confidence Scoring. The confidence score $\mathrm{Conf}(c) \in [0, 1]$ of a composite insight c is computed as

$$\mathrm{Conf}(c) = \mathrm{consistency}(c) \times \frac{1}{|\mathcal{A}_c|} \sum_{a \in \mathcal{A}_c} \mathrm{sig}(a), \tag{12}$$

where $\mathrm{sig}(a) \in [0, 1]$ is the normalized statistical significance of atomic insight a, and $\mathrm{consistency}(c) \in \{0, 1\}$ is a binary logical-coherence check. The field $\mathrm{evidence}(c)$ records whether the composite is directly validated by a joint SQL query (`direct`) or indirectly through atomic SQLs (`indirect`); it does not enter the confidence product. Only composites with $\mathrm{Conf}(c) > 0.6$ are retained. Indirect composites are additionally audited manually in the evaluation. The causal score $\mathrm{Causal}(c)$ is stored as an additional field, and composites with $\mathrm{Causal}(c) > \tau_{\mathrm{causal}}$ are labeled as causal. This design ensures that the final output is reliable, non-redundant, causally verified, and fully traceable for automated data analysis.

B.5 Report Generation

Given the set of composite insights $\mathcal{C}^*$ and an optional user query q, the report generation module synthesizes an article-level analytical report $\mathcal{R}$ as follows.

1. **Relevance filtering.** If q exists, encode q and the summary of each composite insight c with an embedding model, and compute cosine similarity $rel(c, q)$. Set a threshold τ_{rel} and discard composites with $rel(c, q) < \tau_{\mathrm{rel}}$, keeping only those relevant to the query topic. If q is absent, keep all composites.
2. **Thematic grouping.** Cluster the embeddings of all atomic insights into K themes (K-means clustering). Assign each selected composite to the theme containing the majority of its atoms. Each theme becomes a section of the report.
3. **Narrative generation.** For each section, feed the composites and their evidence (atomic insights, SQL, metadata) into an LLM to generate coherent paragraphs. Ensure all insights are mentioned, transitions are smooth, and every claim is accompanied by evidence references.
4. **Report assembly.** Assemble all sections in logical order into a complete report with title, abstract, and conclusion. Output the final report $\mathcal{R}$, where every claim is traceable to its supporting evidence.

B.6 Basic Analysis Patterns and SQL Templates

In Table 3, the basic Analysis Patterns and SQL Templates is shown.

Table 3: Basic Analysis Patterns and SQL Templates

Pattern Type	Description	SQL Template Example
MAX	Returns maximum value of a metric with associated dimension	`SELECT MAX({metric}) FROM {table}`
MIN	Returns minimum value of a metric with associated dimension	`SELECT MIN({metric}) FROM {table}`
AVG	Computes the average of a metric	`SELECT AVG({metric}) FROM {table}`
SUM	Computes the total of a metric	`SELECT SUM({metric}) FROM {table}`
COUNT	Counts the number of records	`SELECT COUNT(*) FROM {table}`
GROUP_BY	Groups by dimension and computes aggregate per group	`SELECT {dim}, {agg}({metric}) FROM {table} GROUP BY {dim}`
CORR	Computes correlation between two metrics	`SELECT CORR({m1}, {m2}) FROM {table}`
TREND	Performs linear regression of metric over time	`SELECT slope, p_value FROM ({table} time regression)`