TabRex: Tabular Referenceless eXplainable Evaluation

Tejas Anvekar[a] **Junha Park**[a] **Aparna Garimella**[b] **Vivek Gupta**[a]

[a]Arizona State University [b]Adobe Research

Project-Page Code

{tanvekar,jpark284,vgupta140}@asu.edu
garimell@adobe.com

Abstract

Evaluating the quality of tables generated by large language models (LLMs) remains an open challenge: existing metrics either flatten tables into text, ignoring structure, or rely on fixed references that limit generalization. We present **TabRex**, a *reference-less, property-driven* framework for evaluating tabular generation via graph-based reasoning. TabRex converts both source text and generated tables into canonical knowledge graphs, aligns them through an LLM-guided matching process, and computes interpretable, rubric-aware scores that quantify structural and factual fidelity. The resulting metric provides controllable trade-offs between *sensitivity* and *specificity*, yielding human-aligned judgments and cell-level error traces. To systematically assess metric robustness, we introduce **TabRex-Bench**, a large-scale benchmark spanning six domains and twelve planner-driven perturbation types across three difficulty tiers. Empirical results show that TabRex achieves the highest correlation with expert rankings, remains stable under harder perturbations, and enables fine-grained model-vs-prompt analysis establishing a new paradigm for *trustworthy, explainable evaluation* of structured generation systems.

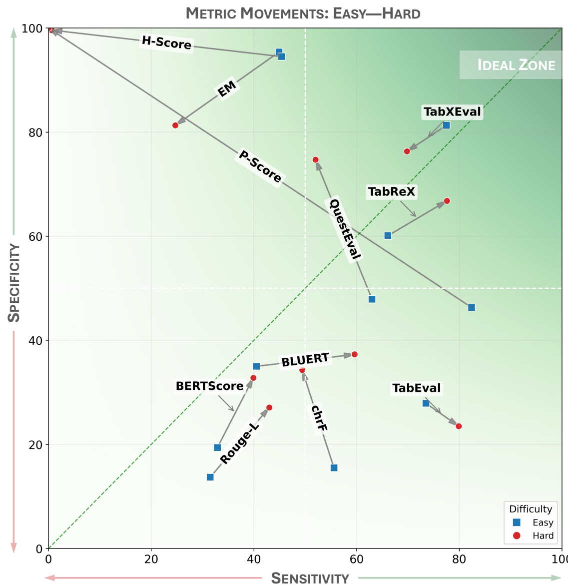

Figure 1: **Metric Movements Across Difficulty Levels.** Arrows show each metric's shift from *easy* (blue) to *hard* (red) perturbations. Axes plot *specificity* (y) vs. *sensitivity* (x), with the green region denoting the balanced *ideal zone*. The dashed diagonal marks the optimal trade-off. TabRex stay near this zone, maintaining right direction even for hard examples.

1 Introduction

Structured data underpins critical workflows across domains such as finance, healthcare, scientific reporting, and logistics. Beyond spreadsheets and relational tables, modern ecosystems rely on JSON records, knowledge graphs, and visual dashboards. These formats enable consistent reasoning and aggregation, yet even a single misplaced column, unit mismatch, or corrupted cell can propagate costly downstream errors.

As large language models (LLMs) increasingly generate or transform structured outputs e.g., converting reports into financial tables, synthesizing patient dashboards, or reformatting analytical data the need for *reliable automatic evaluation* has become a major bottleneck. Unlike free-form text, structured generation demands assessment of not just semantic fidelity but also schema alignment, syntactic consistency, and cell-level correctness.

Most existing metrics, however, flatten tables into plain text. N-gram scores like BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) ignore row-column structure and unit semantics, while embedding-based metrics such as BERTScore (Zhang* et al., 2020) and BLEURT (Sellam et al., 2020) capture semantics but miss structural perturbations. Token-level methods like Exact Match or PARENT (Dhingra

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 47015–47030
July 2-7, 2026

et al., 2019) cannot distinguish harmless reformatting from genuine factual errors. Reference-less QA metrics such as DATAQUESTEVAL (Rebuffel et al., 2021) ground evaluation in source evidence but over-penalize layout changes, and recent TABEVAL (Ramu et al., 2024) and TABXEVAL (Pancholi et al., 2025) improve explainability yet remain limited by small, single-pass benchmarks and one-shot perturbation schemes.

We argue that next-generation evaluation must be both *property-driven* and *personalizable*. Effective metrics should obey key properties-permutation and format invariance, schema- and unit-consistent alignment, monotonic improvement as errors are fixed, and robustness to outliers while allowing tunable trade-offs between *sensitivity* (coverage) and *specificity* (hallucination control). Real-world domains differ in their error tolerance (e.g., precision in finance vs. recall in clinical data), requiring metrics that are domain-agnostic by design yet easily adaptable through interpretable property weights.

To meet these needs, we propose **TABREX**, a graph-based, explainable evaluation framework. TABREX converts both reference text and generated tables into structured graphs via a hybrid pipeline: a rule-based *Table2Graph* converter and an LLM-assisted *Text2Graph* extractor-followed by an LLM-guided *Graph Alignment* that identifies factual correspondences and discrepancies. From these alignments, a *property-driven scoring* function computes interpretable, rubric-aware penalties capturing both structure and content quality, yielding an explainable, reference-less score.

To stress-test metric reliability, we introduce **TABREX-BENCH**, a large-scale benchmark covering six domains (finance, healthcare, hierarchical tables, and narratives) and twelve planner-driven perturbation types across three difficulty levels. Unlike prior one-shot datasets, TABREX-BENCH systematically combines factual and structural edits ranging from benign reformatting to severe semantic corruption enabling robust sensitivity-specificity analysis under realistic perturbation regimes.

In summary, our contributions are:

- **TABREX**: a *reference-less, property-driven* evaluation framework that aligns table–text graphs and computes interpretable, rubric-aware scores.
- **TABREX-BENCH**: a large, systematically perturbed dataset enabling reproducible metric evaluation across domains and difficulty levels.
- Empirical results showing that TABREX achieves strong human correlation and robustness under harder perturbations.
- Rubric-wise analyses demonstrating that TABREX provides explainable diagnostics at both table and cell levels for model–prompt alignment.

2 TABREX

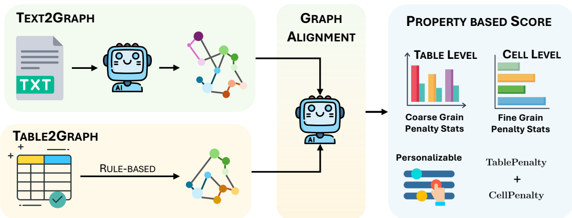

Figure 2: Illustration of proposed **TABREX**. Both source text and generated tables are converted into knowledge graphs via *Text2Graph* and *Table2Graph*, aligned through an LLM-guided *Graph Alignment*, finally scored by a *Property-Driven Scoring* function that aggregates alignment statistics into interpretable, controllable table- and cell-level penalties.

We propose **TABREX**, a unified evaluation framework for tabular generation that converts both candidate table and reference / source text into knowledge graphs and scores them through a small set of *property-driven* signals. This design yields a metric that is *reference-less*, *effective in detecting true discrepancies*, and *explainable* by construction, best illustrated in Figure 2

2.1 Pipeline Overview

Stage 1: Text2Graph and Table2Graph. To enable uniform comparison, TABREX represents both textual summaries and tables as knowledge-graph triplets $[s, p, o]$.

For **text**, we use an LLM guided by a strict entity-centric grammar (Prompt C) to extract minimal atomic facts, where the *subject* is an entity or time slice, the *predicate* a normalized property, and the *object* a canonical value. This design enforces consistent granularity, normalized predicates, and unit-aware values across free-form text: $\mathcal{G}_S = \{(s_i, p_i, o_i) \mid i = 1, \ldots, n\}$.

For **tables**, we apply a lightweight *rule-based unrolling*. Headers define predicates; each row specifies a subject; every non-empty cell yields a triplet $(s_{\text{row}}, p_{\text{header}}, o_{\text{cell}})$. To support diverse table formats, we implemented both `RuleHTMLConverter` and `RuleMDConverter`, and in this work, we use the latter. This deterministic approach is fast, schema-aware, and requires no training.

By converting both modalities into a common, interpretable triplet space, TABREX ensures structural clarity and prepares them for downstream alignment and scoring.

Stage 2: Graph Alignment. In our reference-less setup, we align the graph extracted from the *generated table*, $\mathcal{G}_T$, with that from the *source text*, $\mathcal{G}_S$, so the table can be judged directly against the textual evidence.

Both graphs consist of factual triplets $(s,\ p,\ o)$. The alignment, guided by an LLM prompt (Prompt D), maps triplets in $\mathcal{G}_T$ to their counterparts in $\mathcal{G}_S$.

We adopt a two-step procedure: (i) a deterministic pass aligns triplets with identical or schema-normalized subject–predicate pairs; (ii) an LLM-assisted refinement aligns the remainder, resolving paraphrases, abbreviations, and compound attributes (e.g., “GDP growth (YoY)" $\leftrightarrow$ “growth_rate_2021").

Each matched pair is annotated with a difference vector Δ recording unit-aware numeric gaps, categorical mismatches, and whether a fact is missing in the table or extra relative to the source. The resulting aligned set $\mathcal{A}$ exposes, at the row/column/cell level, the precise correspondences and discrepancies required for property-driven scoring.

Stage 3: Property-Driven Scoring. The aligned set $\mathcal{A}$ provides structured evidence of matches, omissions, and numeric deviations between the table graph $\mathcal{G}_T$ and the source text graph $\mathcal{G}_S$. From these alignments, TABREX derives interpretable statistics counts of missing (MI), extra (EI), and partially matched triplets aggregated over rows, columns, and cells. These alignment-derived quantities directly drive two complementary components capturing structural and factual quality.

$$\begin{aligned}\text{TablePenalty} = \beta_{\text{MI}}\Big(\alpha_r \frac{\text{MI}_r}{N_r} + \alpha_c \frac{\text{MI}_c}{N_c}\Big) \\ + \beta_{\text{EI}}\Big(\alpha_r \frac{\text{EI}_r}{N_r} + \alpha_c \frac{\text{EI}_c}{N_c}\Big),\end{aligned}$$

where N_r and N_c denote the total numbers of rows and columns in $\mathcal{G}_S$, and MI / EI count missing and extra entities, respectively. The *cell-level penalty* captures factual fidelity:

$$\begin{aligned}\text{CellPenalty} = \beta_{\text{MI}}\alpha_{\text{cell}}\frac{\text{MI}_{\text{cell}}}{N_{\text{cell}}} + \beta_{\text{EI}}\alpha_{\text{cell}}\frac{\text{EI}_{\text{cell}}}{N_{\text{cell}}} \\ + \beta_{\text{partial}}\alpha_{\text{cell}}\frac{\Gamma}{N_{\text{cell}}},\end{aligned}$$

where Γ is the sum of normalized numeric deviations over partially aligned cells. The final score combines both components:

$$\mathcal{S}_{\text{TABREX}} = \text{TablePenalty} + \text{CellPenalty}.$$

The weighting parameters (α, β) provide intuitive control over the metric's behavior: increasing β_{MI} favors *sensitivity* (rewarding comprehensive coverage), while increasing β_{EI} favors *specificity* (penalizing hallucinated entries). Because all quantities are derived directly from $\mathcal{A}$, the score remains reference-less, and fully explainable. All the weight configurations and a walk through example is illustrated in Appendix B.

2.2 TABREX-BENCH

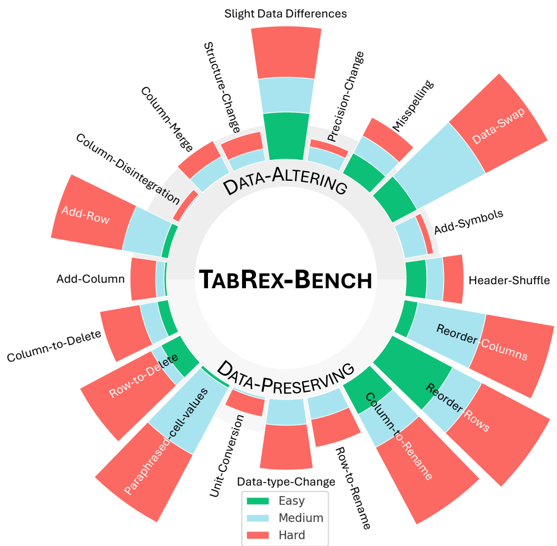

Figure 3: **Perturbation landscape across difficulty and type.** The radial stacked donut visualizes the distribution of perturbation types segmented by difficulty: *Easy* (green), *Medium* (blue), and *Hard* (red). The top and bottom semicircles correspond to *data-altering* and *data-preserving* transformations, respectively.

TABREX-BENCH is a comprehensive benchmark for evaluating tabular metrics under both *data-preserving* and *data-altering* perturbations. Unlike prior resources such as TABXBENCH (Pancholi et al., 2025), which includes only 50 reference tables with 5 perturbations each, TABREX-BENCH spans six heterogeneous datasets FinQA (Chen

Dataset	# of Tables	# Pert / Table	Tables	Avg Row	Avg Col	Avg Cell	Avg Tokens	Avg Num
FinQA	150	12	1950	05.55	02.47	13.22	119.5	33.55
HiTabQA	150	12	1950	20.08	05.60	115.1	434.8	102.7
ToTTo	150	12	1950	24.97	05.49	142.2	361.3	69.63
OpenML med	10	12	120	04.20	11.58	47.94	210.9	23.80
MIMIC-IV	100	12	1200	10.58	03.94	40.84	153.5	26.29
RotoWire	150	12	1950	10.18	05.86	59.50	146.5	14.33
Total	710		9120					

Table 1: Statistics of TABREX-BENCH: Datasets, perturbation counts, and average table and summary characteristics.

et al., 2021), HiTabQA (Cheng et al., 2022), ToTTo (Parikh et al., 2020), OpenML-med (Smith et al., 1988; Centers for Medicare & Medicaid Services, 2019), MIMIC-IV (Johnson et al., 2024), and RotoWire (Wiseman et al., 2017) covering finance, healthcare, hierarchical tables, and narrative-to-table tasks. As summarized in Table 1, the benchmark comprises 710 source tables, each expanded with 12 perturbations, yielding 9,120 perturbed instances spanning compact clinical sheets to large multi-column tables.

Figure 3 illustrates the perturbation composition. We define two complementary perturbation groups: *Data-Preserving* (Group 0) alters layout or presentation e.g., row or header reordering, unit conversion, or paraphrasing without changing factual content; *Data-Altering* (Group 1) introduces semantic modifications such as adding or deleting rows/columns, swapping numeric values, or injecting noise and misspellings. Each group is further stratified into three difficulty tiers (*Easy*, *Medium*, *Hard*), supporting controlled analyses of metric robustness as perturbation severity increases.

A key innovation over prior work is our **planner-driven perturbation generation**. Rather than issuing separate LLM calls for each edit, TABREX-BENCH employs an LLM-based planner (Prompt B) that generates executable code to produce all 12 perturbations across both groups and difficulty levels in a single pass, yielding more diverse and reproducible variants. Each perturbed table is also paired with a concise, fact aligned **table-level summary** (Prompt A) and stats for the avg # token and Numerical data present are given in Table 1, enabling the evaluation of reference-less metrics assessing factual consistency between tables and summaries an aspect not present in TABXBENCH.

All perturbations and summaries were initially generated through this planner-driven pipeline and validated on 20% of the data, achieving inter-annotator agreement of 87% for summaries and 91% for perturbations, ensuring correctness and diversity. By combining broad domain coverage, structured perturbation design, paired summaries, and tiered difficulty, TABREX-BENCH enables rigorous evaluation of metric robustness, sensitivity, and human alignment across both reference-based and reference-less settings.

3 Experiments

To assess the efficacy of TABREX, we conduct experiments using our synthetic benchmark TABREX-BENCH. All results are reported with `GPT-5-nano` (Team, 2025b), evaluating both components of TABREX: *Text2Graph* and *Graph Alignment* using proposed TABREX-BENCH dataset.

Baselines. We compare TABREX against a diverse set of automatic evaluation metrics grouped by methodological design. *Deterministic* metrics: Exact Match (EM), CHRF, and ROUGE-L: compute token- or character-level overlaps, offering reproducible yet surface-biased comparisons. *Algorithmic* metrics such as H-SCORE perform structured alignment and rule-based matching without relying on neural embeddings, offering deterministic, training-free evaluation. *Neural* metrics such as BERTSCORE and BLUERT leverage contextual embeddings to capture semantic similarity but may exhibit variability across runs. Among recent LLM-based approaches, we include P-SCORE (an LLM-judged quality metric producing 0–10 scores) and TABEVAL, which flattens tables via an LLM and measures entailment using RoBERTa-MNLI. We also evaluate the state-of-the-art TABXEVAL, a two-phase rubric-based framework that first aligns tables structurally (*TabAlign*) and then performs semantic and syntactic comparison (*TabCompare*) for interpretable, human-aligned evaluation. Finally, we benchmark the reference-less QUESTEVAL, which generates question–answer pairs from both the source and the generated text or table, performs cross-validation using two LLM calls, and computes F1 scores to measure factual and semantic consistency.

LLMs. We conduct all experiments using `GPT-5-nano`, `Gemma-3` (4B/27B-Instruct) (Team, 2025a), and `InternVL3.5` (8B-

Instruct/Thinking) (Wang et al., 2025). Unless stated otherwise, we employ uniform decoding settings across models, using their default temperature, top-k, and top-p parameters. All gpu-intensive experiments were conducted on NVIDIA-2×H100s. The full prompts for *Text2Graph* (Prompt C) and *Graph Alignment* (Prompt D) are provided in Appendix D.

3.1 Correlation Analysis of Metrics Category.

Metric	$\rho_S\uparrow$	$\tau_K\uparrow$	$\tau_w\uparrow$	RBO ↑	$\zeta_F\downarrow$	$\pi_t\downarrow$
Non-LLM Based (w/ Ref)						
EM	45.88	39.38	39.51	43.33	47.49	58.40
CHRF	41.76	34.55	31.61	39.39	49.26	01.64
ROUGE-L	31.18	26.69	22.56	37.65	55.94	01.97
BLUERT	44.66	37.64	36.09	39.57	48.09	00.77
BERTSCORE	36.21	30.66	27.96	38.11	53.25	00.92
H-SCORE	56.87	47.97	51.73	41.11	40.02	00.99
LLM-Based (w/ Ref)						
P-SCORE	49.24	40.00	37.43	40.73	43.93	07.39
TABEVAL	49.01	39.22	34.21	41.11	43.06	00.63
TABXEVAL	**80.27**	**72.37**	**66.87**	**47.54**	**20.94**	45.33
(w/o Ref)						
QUESTEVAL	62.93	52.29	51.71	42.70	35.04	03.03
TABREX	**74.51**	**64.24**	**62.28**	**44.85**	**27.01**	13.59

Table 2: Correlation of automatic evaluation metrics with human rankings across synthetic perturbation sets. Higher values of Spearman's rank correlation (ρ_S), Kendall's tau (τ_K), weighted Kendall's tau (τ_w), and Rank-Biased Overlap (RBO) indicate stronger monotonic and positional agreement with human orderings (↑), while lower values of Spearman's footrule distance (ζ_F) and tie ratio (π_t) denote better rank stability and finer discriminative resolution (↓). The proposed TABREX achieves the best overall consistency with human judgment.

Table 2 reports the correlation between automatic evaluation metrics and human judgments over the synthetic perturbation benchmark. Each ground-truth (GT) table was paired with twelve systematically perturbed variants six preserving factual content (labels 0: *1-easy*, *1-medium*, *1-hard*) and six introducing data alterations (labels 1: *1-easy*, *1-medium*, *1-hard*). Human annotators ranked these variants by perceived semantic and factual fidelity to the GT, providing a gold human order for correlation analysis. Metrics are grouped by family Non-LLM, LLM-based, and reference-less to examine their consistency and robustness under controlled perturbations.

(a) Non-LLM metrics. such as EM, CHRF, and ROUGE-L show limited alignment with human judgment. Their Spearman's (ρ_S) and Kendall's (τ_K) values remain low ($\rho_S < 0.45$, $\tau_K < 0.35$), indicating that rank orderings diverge substantially from human perception. Sentence-level embedding metrics (BLEURT, BERTSCORE) capture partial semantic similarity but exhibit modest RBO (≈0.39) and high footrule distances ($\zeta_F \approx 4553$), reflecting poor rank stability. Their near-zero tie ratios ($\pi_t < 2\%$) further suggest coarse differentiation, failing to separate semantically close variants.

(b) LLM-based metrics. such as P-SCORE, TABEVAL, and TABXEVAL show notably higher agreement with human preferences ($\rho_S \approx 0.490.80$, $\tau_K \approx 0.390.72$). Among them, TABXEVAL achieves the strongest overall correlation ($\rho_S = 0.80$, $\tau_K = 0.72$), confirming that instruction-tuned evaluators capture perturbation sensitivity effectively. However, its elevated tie ratio ($\pi_t = 45.3\%$) and moderate rank dispersion ($\zeta_F = 20.9$) indicate frequent scoring saturation, where distinct variants receive identical judgments reducing discriminative precision even when global trends align.

(c) Reference-less metrics. Without access to reference tables, QUESTEVAL maintains moderate alignment ($\rho_S = 0.63$, $\tau_K = 0.52$) by generating QA pairs from both the source and system outputs, yet exhibits instability under data-altering perturbations. In contrast, **our metric** achieves the most balanced performance across all dimensions Spearman's $\rho_S = 0.75$, Kendall's $\tau_K = 0.64$, and weighted $\tau_w = 0.62$ while also maintaining competitive RBO (44.9) and low rank dispersion ($\zeta_F = 27.0$). Its moderate tie ratio ($\pi_t = 13.6\%$) indicates finer discriminative granularity, avoiding overconfidence and reflecting human-perceived difficulty progression. Together, these findings highlight that our method preserves ordinal consistency across perturbation severity while generalizing robustly in the absence of reference data.

(d) Ensemble of Scores. We further benchmarked ensemble baselines that aggregate complementary metrics using either simple averaging (*Mean*) or harmonic averaging (*Harmonic*). These ensembles span three families: *Lex-Emb* (EM, ROUGE-L, BERTSCORE, BLEURT, CHRF), *LLM* (P-SCORE, H-SCORE), and *Hybrid* (TABXEVAL, QUESTEVAL). While the best-performing ensemble, *LLM (Harmonic)*, achieves $\rho_S = 0.56$ and $\tau_K = 0.47$, it still lags behind our TABREX, which attains $\rho_S = 0.75$ and $\tau_K = 0.64$ with lower rank dispersion. This highlights that naive aggregation of diverse metrics cannot match

Metric	$\rho_S \uparrow$	$\tau_K \uparrow$	$\tau_w \uparrow$	RBO ↑	$\zeta_F \downarrow$	$\pi_t \downarrow$
Ensemble Baselines						
Lex-Emb (M)	38.43	32.65	30.17	38.52	52.15	00.49
Lex-Emb (H)	29.80	24.00	19.68	37.65	55.04	00.63
LLM (M)	48.49	39.21	36.94	40.56	44.38	00.42
LLM (H)	56.00	46.93	50.64	40.95	40.63	00.42
Hybrid (M)	32.04	24.94	20.29	37.03	51.51	01.13
Hybrid (H)	54.03	42.71	32.61	42.31	40.11	01.13
TabReX	**74.51**	**64.24**	**62.28**	**44.85**	**27.01**	13.59

Table 3: Comparison of ensemble baselines with the proposed TabReX. Ensembles combine metric families: Lex-Emb (lexical + embedding), LLM (LLM-based), and Hybrid (reference + reference-less) using either simple Mean (M) or Harmonic (H) aggregation. All ensemble variants fall short of TabReX, which achieves the highest correlation with human rankings and better rank stability.

the targeted, reference-less reasoning of TabReX, which better aligns with human judgment across perturbation severities.

3.2 Can TabReX Generalize Across Perturbation Regimes?

A robust evaluation metric must remain reliable not only in standard (*easy*) settings but also under *hard perturbations* tables with subtle misalignments, semantic shifts, or fine-grained numeric errors. Using our proposed TabReX-Bench, we sample both *easy* and *hard* cases across data-preserving and data-changing perturbations to compute true-positive and true-negative rates (sensitivity and specificity). Figure 1 plots each metric's *trajectory* on the specificity–sensitivity plane as difficulty increases, revealing whether it remains stable or degrades under stress.

Embedding-Driven Metrics. Many popular metrics (e.g., BERTScore, BLUERT, TabEval) rely on neural embeddings rather than surface-level string matching. For example, TabEval first unrolls tables into natural-language atomic statements using an LLM, then applies RoBERTa-MNLI (Liu et al., 2019) to score entailment between candidate and reference statements. Such embedding-based approaches capture deeper semantics, yet as Figure 1 shows, they still exhibit large drops in sensitivity or specificity under harder perturbations.

Stability vs. Fragility. Metrics with only *short arrow movements* from easy to hard cases (e.g., TabXEval, TabReX) demonstrate stable trade-offs and thus *robust generalization*. Interestingly, even though TabXEval sits in the ideal zone, its trajectory drifts slightly *away* from the optimal direction as difficulty rises. By contrast, metrics such as EM, H-Score, and even the LLM-based P-Score experience *sharp drops in sensitivity*, revealing an over-reliance on surface-level cues-showing that an LLM backbone alone does not guarantee proper alignment.

Reference-less Metrics. Both QuestEval and our proposed TabReX evaluate tables without explicit references, instead judging how well a candidate table supports automatically generated questions. QuestEval employs an LLM for question generation and a QA module to assess semantic fidelity, but its reliance on generic QA signals often penalizes harmless re-orderings or formatting changes. In contrast, TabReX tailors question generation to tabular structure and integrates explicit reasoning over extracted facts, enabling it to better separate meaningful discrepancies from superficial variations. As shown in Figure 1, this specialization helps TabReX stay closer to the ideal zone even under tougher perturbations, reflecting stronger alignment with human judgment.

Towards Trustworthy Evaluation. These results highlight the importance of *balanced, difficulty-robust metrics* for downstream evaluation. As generative table models encounter noisier, real-world data, reliable metrics must *reward genuine comprehension* rather than superficial matches. The ability of TabReX to remain in the green "ideal zone" across difficulty levels-despite being *reference-less* underscores its suitability for *high-stakes domains* such as scientific reporting and financial auditing, where both *false alarms* and *missed discrepancies* can be costly.

3.3 Evaluation on Text-to-Table Task

To assess robustness of our method in realistic reference-less settings, we evaluate its performance on text-to-table generation across diverse domains including finance, healthcare, and sports. Generated tables are produced by strong open and proprietary LLMs (`Gemma-3-(4/27B)`, and `InternVL-3.5-thinking (on/off)`). Humans ranking generated tables across models and prompting strategies (zero-shot, CoT, Map&Make).

Expert annotators ranked the model outputs along three axes *structural correctness*, *factual fidelity*, and *semantic coverage*. We then measured how well automatic metrics correlate with these human rankings (detailed in Appendix A) using Spearman's ρ_S, Kendall's τ_K, and rank-biased overlap (RBO).

Table 4: Correlation of automatic metrics with human rankings on real-world text-to-table generation. TabRex achieves the highest alignment across all correlation metrics.

Metric	$\rho_S \uparrow$	$\tau_K \uparrow$	RBO $\uparrow$
EM	-0.01	0.01	0.33
ROUGE-L	0.33	0.25	0.29
BERTScore	0.26	0.19	0.38
BLEURT	0.29	0.20	0.39
CHRF	0.25	0.19	0.36
QuestEval (ref-less)	0.28	0.20	0.39
TabEval (ref-based)	0.25	0.19	0.36
TabXEval (ref-based)	0.24	0.17	0.37
TabRex (ref-less)	**0.39**	**0.30**	**0.41**

Observations. Surface- and embedding-based metrics (e.g., ROUGE-L, BERTScore, BLEURT) exhibit weak correlation with human preferences, primarily due to their sensitivity to lexical and formatting variation. QuestEval performs better but remains brittle to domain-specific structure shifts such as nested headers or missing subtables. In contrast, TabRex achieves the strongest correlations across all measures **Spearman's** $\rho = 0.39$, **Kendall's** $\tau_b = 0.30$, and **RBO=0.41** demonstrating superior alignment with expert judgments. Its graph-based reasoning captures factual and structural consistency more effectively, validating its reliability as a *reference-less* evaluator for real-world table generation systems.

3.4 Rubric-wise Model–Prompt Alignment

TabRex rubric-aware scoring enables coarse to fine-grained comparison across *models* (e.g., Gemma 8B vs. 27B, InternVL-Thinking On vs. Off) and *prompting strategies* (Zero-Shot, Chain-of-Thought, Map&Make (Ahuja et al., 2025)), measured at both *cell-level* and *table-level* granularity (Figure 4).

Cell-level alignment (top row). Larger models (e.g., Gemma 27B) show clear gains in local fidelity especially for numeric and structural rubrics but only modest improvement in semantic consistency. Reasoning-oriented ("Thinking") variants improve precision on numeric and structural dimensions yet often underperform on partial or contextual agreement, suggesting over-cautious reasoning can reduce semantic coverage. Chain-of-Thought prompting enhances numeric correctness but sometimes amplifies inconsistency, while Map&Make maintains more balanced yet slightly conservative performance.

Table-level alignment (bottom row). At a global scale, model size yields diminishing returns: Gemma 27B's advantage narrows, and "Thinking" variants do not consistently outperform standard modes. Zero-shot improves row-column coherence but increases rubric variance. Map&Make achieves steadier rubric alignment, indicating stronger integration of local reasoning into structural organization.

Insights. Overall, three trends emerge: (1) larger models enhance fine-grained (cell-level) fidelity but not global coherence; (2) "Thinking" reasoning improves precision but limits coverage, favoring accuracy over breadth; and (3) prompt design particularly Map&Make contributes as much as model scale to balanced rubric alignment.

These results illustrate how a referenceless, explainable evaluation metric can reveal the strengths and weaknesses of models and prompting strategies across hierarchical levels. Such rubric-aware scorers enable targeted analysis and can support verifiable reward modeling (Shao et al., 2024) for improved alignment.

4 Comparison with Related Work

From Text-to-Table to Structural Benchmarks. Early text-to-table datasets such as RotoWire for basketball summaries (Wiseman et al., 2017), E2E for restaurant descriptions (Novikova et al., 2017), WikiBio for infobox biographies (Lebret et al., 2016), and WikiTableText (Pasupat and Liang, 2015) provided important initial testbeds but offered limited schema diversity and often encouraged hallucinated or under-structured outputs. Recent resources, including StructBench (Gu et al., 2025) and TanQ (Akhtar et al., 2025), introduced challenging phenomena such as header permutations, schema reshuffling, and multi-hop reasoning. These benchmarks exposed fundamental weaknesses in both generation models and evaluation metrics, motivating the need for metrics that go beyond surface overlap and can reason about structural and semantic fidelity.

Metric Families: From Overlap to Explainability. Conventional reference-based metrics: BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004), METEOR (Banerjee and Lavie, 2005), chrF (Popović, 2015), and even embedding-based

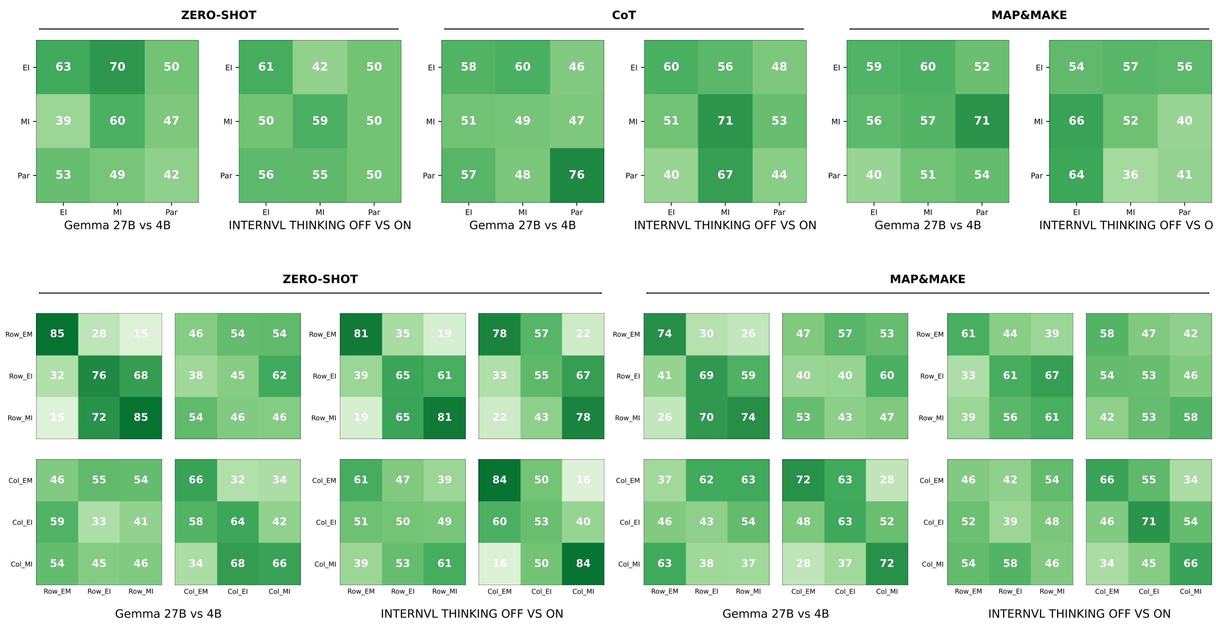

Figure 4: **Rubric-wise alignment across models and prompting strategies.** Top row: cell-level agreement within model across prompts. Bottom row: table-level agreement. Model size and reasoning style influence local precision more than structural coherence, while prompt strategy (like Map&Make (Ahuja et al., 2025)) drives balanced alignment across rubric dimensions.

BERTScore (Zhang* et al., 2020) treat tables as flat text, often ignoring header alignment, units, or cell hierarchy. PARENT (Dhingra et al., 2019) partly grounds evaluation in the input source but still struggles with schema-level changes. Algorithmic and LLM-assisted metrics such as H-Score and P-Score (Tang et al., 2024) move toward structural sensitivity but differ in design: the former computes heuristic, rule-based structural and content similarity, while the latter leverages LLM judgments; both offer limited interpretability. TabEval (Ramu et al., 2024) improves semantic coverage by decomposing tables into atomic statements and applying textual entailment, yet incurs NLI overheads and often over-penalizes harmless layout differences. The recent **TabXEval** (Pancholi et al., 2025) represents a step-change: its two-phase design *TabAlign* for structural alignment and *TabCompare* for semantic/syntactic checks delivers interpretable cell-level diagnostics and consistently balances *sensitivity* and *specificity*, achieving strong human correlation and placing it in the "Goldilocks" zone for robust evaluation.

Reference-less Evaluation and Remaining Gaps. Metrics such as QuestEval and Data-QuestEval (Rebuffel et al., 2021) demonstrate that reference-less evaluation is viable by generating and answering questions over the source data, showing strong alignment with humans in data-to-text tasks. However, their reliance on generic QA signals often misses table-specific structural errors, unit inconsistencies, or localized discrepancies. Despite advances from overlap-based to LLM-driven and rubric-based methods, most existing approaches still emphasize either semantics or structure and condense diverse errors into a single opaque score, limiting error traceability and robustness under realistic perturbations.

5 Conclusion and Future Work

We introduced **TabReX**, a property-driven, reference-less framework for evaluating tabular generation through graph-based reasoning and interpretable, rubric-aware scoring. By unifying structured alignment, factual comparison, and sensitivity–specificity control within a single pipeline, TabReX delivers consistent, human-aligned judgments that remain robust under domain shifts and perturbation difficulty. Our accompanying benchmark, **TabReX-Bench**, establishes a new standard for systematic stress testing of table metrics across six diverse domains and twelve controlled perturbation types.

Experiments demonstrate that TabReX not only correlates most strongly with human evaluations but also provides fine-grained, explainable diagnostics at both cell and table levels enabling actionable analysis of model and prompt behaviors. Beyond outperforming reference-based and LLM-judge baselines, it shows that reliable table evaluation is possible without explicit references by reasoning over grounded factual graphs.

Future work will focus on extending TABREX to richer structural formats such as hierarchical or multi-modal tables, and on distilling its LLM components into lightweight, domain-adaptive evaluators for scalable deployment. We envision TABREX as a foundation for *trustworthy, interpretable evaluation* in structured generation supporting better model selection, alignment, and reward learning across real-world applications.

6 Limitations

While TABREX achieves robust and interpretable evaluation, it has a few limitations. It relies on large language models for fact extraction and alignment, which adds computational cost and mild variability due to model stochasticity. The current implementation supports only structured digital tables (e.g., HTML, Markdown) and cannot yet handle tables embedded in images or PDFs requiring OCR or visual parsing. Finally, although TABREX-BENCH spans six diverse domains, it remains limited to English and synthetic perturbations, leaving real-world noise, multilingual data, and complex layouts for future exploration.

7 Ethics Statement

The authors affirm that this work adheres to the highest ethical standards in research and publication. Ethical considerations have been meticulously addressed to ensure responsible conduct and the fair application of computational linguistics methodologies. Our findings are aligned with experimental data, and while some degree of stochasticity is inherent in black-box Large Language Models (LLMs), we mitigate this variability by maintaining fixed parameters such as temperature, top_p, and top_k. Furthermore, our use of LLMs, including `GPT-5-nano`, `Gemma`, and `InternVL`, complies with their respective usage policies. To refine the clarity and grammatical accuracy of the text, AI based tools such as Grammarly and ChatGPT were employed. Additionally, human annotators who are also among the authors actively contributed to data labeling and verification, ensuring high-quality annotations. To the best of our knowledge, this study introduces no additional ethical risks.

Maintenance and Adoption Plan

To support long-term use and community adoption, we release all artifacts under permissive terms and commit to active maintenance. The TABREX reference implementation (including *Text2Graph*, *Graph Alignment*, and *Property-Driven Scoring*) is open-sourced at `https://github.com/CoRAL-ASU/TabReX` under the MIT License, and TABREX-BENCH is hosted on the Hugging Face Hub with versioned revisions so that published numbers remain reproducible even as the dataset grows.

Ongoing maintenance is coordinated by **Junha Park** (`jpark284@asu.edu`) as the primary point of contact, with bug reports, feature requests, and new-domain contributions handled through GitHub Issues and Pull Requests. We will publish tagged releases for breaking changes and track upstream LLM judge deprecations (e.g., swapping `gpt-5-nano` for successor models) so that TABREX continues to function as provider APIs evolve. Contributions of additional evaluation domains, perturbation types, and alternative judge back-ends (including the embedding-based matcher described in Appendix C) are explicitly welcomed via the contribution guidelines in the repository.

Acknowledgments

We thank the Complex Data Analysis and Reasoning Lab at Arizona State University for computational support. We also thank Adobe Research for supporting this work. Finally, we like to thank the anonymous reviewers for valuable feedbacks which helped improving the manuscript.

References

Naman Ahuja, Fenil Bardoliya, Chitta Baral, and Vivek Gupta. 2025. Map&make: Schema guided text to table generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30249–30262, Vienna, Austria. Association for Computational Linguistics.

Mubashara Akhtar, Chenxi Pang, Andreea Marzoca, Yasemin Altun, and Julian Martin Eisenschlos. 2025. TANQ: An open domain dataset of table answered questions. *Preprint*, arXiv:2405.07765.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Pro-*

ceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Centers for Medicare & Medicaid Services. 2019. Medical charges. Retrieved from OpenML (ID: 43928).

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021. FinQA: A Dataset of Numerical Reasoning over Financial Data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. 2022. HiTab: A hierarchical table dataset for question answering and natural language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1094–1110, Dublin, Ireland. Association for Computational Linguistics.

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4884–4895, Florence, Italy. Association for Computational Linguistics.

Zhouhong Gu, Haoning Ye, Xingzhou Chen, Zeyang Zhou, Hongwei Feng, and Yanghua Xiao. 2025. StrucText-eval: Evaluating large language model's reasoning ability in structure-rich text. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 223–244, Vienna, Austria. Association for Computational Linguistics.

Alistair Johnson, Luca Bulgarelli, Tom Pollard, Brian Gow, Benjamin Moody, Steven Horng, Leo Anthony Celi, and Roger Mark. 2024. MIMIC-IV (version 3.1). RRID:SCR_007345.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E Dataset: New Challenges For End-to-End Generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

Vihang Pancholi, Jainit Sushil Bafna, Tejas Anvekar, Manish Shrivastava, and Vivek Gupta. 2025. TabX-Eval: Why this is a bad table? an eXhaustive rubric for table evaluation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22913–22934, Vienna, Austria. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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

Panupong Pasupat and Percy Liang. 2015. 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)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Pritika Ramu, Aparna Garimella, and Sambaran Bandyopadhyay. 2024. Is this a bad table? a closer look at the evaluation of table generation from text. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22206–22216, Miami, Florida, USA. Association for Computational Linguistics.

Clement Rebuffel, Thomas Scialom, Laure Soulier, Benjamin Piwowarski, Sylvain Lamprier, Jacopo Staiano, Geoffrey Scoutheeten, and Patrick Gallinari. 2021. Data-QuestEval: A Referenceless Metric for Data-to-Text Semantic Evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8029–8036, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. Bleurt: Learning robust metrics for text generation. *Preprint*, arXiv:2004.04696.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

J.W. Smith, J.E. Everhart, W.C. Dickson, W.C. Knowler, and R.S. Johannes. 1988. Using the adap learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Symposium on Computer Applications and Medical Care*, pages 261–265. IEEE Computer Society Press.

Xiangru Tang, Yiming Zong, Jason Phang, Yilun Zhao, Wangchunshu Zhou, Arman Cohan, and Mark Gerstein. 2024. Struc-bench: Are large language models good at generating complex structured tabular data? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 12–34, Mexico City, Mexico. Association for Computational Linguistics.

Gemma Team. 2025a. Gemma 3.

OpenAI Team. 2025b. GPT-5 System Card.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, and 1 others. 2025. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in Data-to-Document Generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Appendix

A Human Evaluation Protocol

Human annotators were instructed to evaluate the similarity of generated tables to the gold (ground-truth) tables whenever available or against source text following a consistent rubric. Each annotation batch contained one gold table and five generated candidates. Annotators ranked candidates from 1 (best) to 5/12 (depending on task) (worst) based on their structural and contextual fidelity to the gold table.

Structural Factors. Annotators prioritized structural integrity in the following order: (1) *Column Missing* - tables omitting columns were penalized most heavily; (2) *Column Extra* - extra columns ranked lower in case of ties; (3) *Row Missing* and (4) *Row Extra* - missing or spurious rows reduced rank; (5) *Cell Missing* and (6) *Cell Extra* - missing or redundant cells influenced ranking proportionally; (7) *Partial Mismatching Severity* - deviations in value accuracy or format were also considered.

Contextual Factors. Within equal structural quality, contextual accuracy guided ranking: (1) string-value mismatches, (2) numeric, boolean, or date-time inaccuracies, (3) inconsistencies in list-type entries, and (4) deviations in other less common data types.

Tie-Breaking. In case of ties, rankings were determined by the number of affected cells within rows and columns. Column headers with semantically incorrect or mismatched meanings were treated as "wrong columns" and penalized equivalently to missing columns.

This rubric ensured consistent and interpretable human rankings aligned with the metric's property-driven principles.

B Walk-Through Example of TabRex

For full details of the formalism, please refer to the main paper. Here we provide only the default hyperparameters and a worked example to show how the score is computed in the reference-less setting.

Symbol	Meaning	Value
β_{MI}	Weight for Missing Information (MI)	1.0
β_{EI}	Weight for Extra Information (EI)	0.9
β_{partial}	Weight for partially correct cell values	0.8
α_r	Row-level (subject) structural weight	0.9
α_c	Column-level (predicate) structural weight	1.0
α_{cell}	Cell-level (object) structural weight	0.8
ω_p	Scaling factor for partial deviation γ	0.9

Table 5: Default TabRex hyperparameters.

Hyperparameters.

Setup. Let $\mathcal{G}_S$ be the source-text evidence graph and $\mathcal{G}_T$ the generated-table graph. All counts below are measured relative to $\mathcal{G}_S$. Assume

$$N_r = 5, \quad N_c = 4, \quad N_{\text{cell}} = 20,$$

with discrepancies:

$$\mathrm{MI}_r = 1, \quad \mathrm{EI}_c = 1, \quad \mathrm{MI}_{\text{cell}} = 2, \quad \mathrm{EI}_{\text{cell}} = 1,$$

and two partially aligned cells with normalized deviations 0.2 and 0.5.

Step 1: Table-level penalty.

$$\begin{aligned}\text{TablePenalty} &= \beta_{\mathrm{MI}}\alpha_r\frac{\mathrm{MI}_r}{N_r} + \beta_{\mathrm{MI}}\alpha_c\frac{\mathrm{MI}_c}{N_c} \\ &\quad + \beta_{\mathrm{EI}}\alpha_r\frac{\mathrm{EI}_r}{N_r} + \beta_{\mathrm{EI}}\alpha_c\frac{\mathrm{EI}_c}{N_c} \\ &= 1.0(0.9\tfrac{1}{5}) + 0.9(1.0\tfrac{1}{4}) \\ &= 0.18 + 0.225 = 0.405.\end{aligned}$$

Step 2: Cell-level penalty. Partial-match deviations:

$$\gamma_1 = \omega_p \cdot 0.2 = 0.18, \quad \gamma_2 = \omega_p \cdot 0.5 = 0.45,$$
$$\sum_i \gamma_i = 0.63.$$

$$\begin{aligned}\text{CellPenalty} &= \beta_{\mathrm{MI}}\alpha_{\text{cell}}\frac{\mathrm{MI}_{\text{cell}}}{N_{\text{cell}}} + \beta_{\mathrm{EI}}\alpha_{\text{cell}}\frac{\mathrm{EI}_{\text{cell}}}{N_{\text{cell}}} \\ &\quad + \beta_{\text{partial}}\alpha_{\text{cell}}\frac{1}{N_{\text{cell}}}\sum_i \gamma_i \\ &= 1.0\times0.8\times\tfrac{2}{20} + 0.9\times0.8\times\tfrac{1}{20} \\ &\quad + 0.8\times0.8\times\tfrac{0.63}{20} \\ &= 0.08 + 0.036 + 0.0202 = 0.1362.\end{aligned}$$

Step 3: Final score.

$$\begin{aligned}\mathcal{S}_{\text{TabRex}} &= \text{TablePenalty} + \text{CellPenalty} \\ &= 0.405 + 0.1362 = 0.5412.\end{aligned}$$

Interpretation. The example shows that both structural discrepancies (missing rows, extra columns) and factual deviations (partially mismatched cell values) jointly contribute to the final reference-less TabRex score.

C Validation of LLM-based KG Extraction

A natural concern with TabRex is that the summary-KG extractor is itself an LLM: if the extractor is unreliable, every downstream score inherits that noise. We address this concern with an empirical study on a 30-sample stratified validation set (6 domains $\times$ 5 summaries per domain, sampled uniformly at random).

Gold KG Construction. Human annotators manually authored the gold KG for each of the 30 summaries, producing **364 triples** in total (mean 12.1, range 623). Triples follow a `(subject, predicate, object)` schema.

Extraction Quality. We compare three extractors against the gold KG: (i) **gpt-5-nano**, (ii) **Qwen3.5-27B**, and (iii) **Gemma-4-26B**. We report micro-averaged triple P/R/F1 under four matchers of increasing tolerance:

- **Exact** - normalized string equality on all three positions.
- **Cross** - token-bag Jaccard ≥ 0.6 allowing components to slide across subject/predicate/object positions.
- **Embed** - whole-triple cosine on `"s | p | o"` using `BAAI/bge-base-en-v1.5` (L2-normalized), with greedy 1-to-1 matching at threshold 0.85. This matcher is *LLM-free* and therefore independent of any judge model.
- **Semantic** - LLM-judge alignment using the same Graph Alignment prompt (Prompt D) employed by TabRex.

Model	Exact F1	Cross F1	Embed F1	Semantic F1
gpt-5-nano	0.262	0.450	**0.854**	**0.914**
Qwen3.5-27B	0.242	0.450	**0.903**	**0.869**
Gemma-4-26B-A4B	0.261	0.361	**0.810**	**0.840**

Table 6: Triple-level P/R/F1 of three LLM extractors against the gold KG under four matchers of increasing tolerance.

Exact and cross F1 are low because the extractors paraphrase entities and predicates freely (e.g. *"has_value"* vs. *"recorded_value"*; *"Q2 2023"* vs. *"second quarter of 2023"*)-exactly the kind of variation TabRex's alignment stage is designed to absorb. Under the two paraphrase-tolerant matchers, all three extractors recover 0.810.91 F1, and the two matchers agree within 0.06 F1 on every model despite being built on independent signals (LLM judgment vs. dense embeddings). This cross-check rules out the circularity risk that a single LLM judge might over-credit its own family's paraphrases.

Does Extractor Choice Move the Final TabRex Score? This is the bottom-line check. For each of the 30 samples we run the full TabRex pipeline **four times**, varying only the summary-KG source: gold KG, gpt-5-nano KG, Qwen3.5-27B KG, and Gemma-4-26B KG. The table graph and scoring rule are fixed; alignments are cached by a hash of `(judge, summary-KG, table-KG)`.

Mean TabRex scores differ from gold by less than 1% (gold 39.71; extractors

Prompt A: Table Summary Generation

```
# System Prompt
You are a neutral data narrator for arbitrary domains. Write a cohesive, flowing paragraph (4-7 sentences) describing the
    ↪ information in a markdown table. Do not ask for additional data or refuse; never mention the table itself or
    ↪ formatting. Avoid lists, bullets, colons, or name:value patterns; use full sentences and connect ideas. Summarize
    ↪ salient figures, ranges, extremes, comparisons, and notable trends. Light interpretation is allowed if consistent
    ↪ with the numbers.Do not mention the table or its structure. Plain text only.


# User Prompt
f"Markdown Table: {markdown_table}"
```

Model	Entity Coverage	Faithfulness
gpt-5-nano	0.794	0.884
Qwen3.5-27B	0.866	0.892
Gemma-4-26B	0.808	0.922

Table 7: Auxiliary per-sample metrics (macro-averaged over 30 samples). **Entity coverage** is the fraction of unique gold subjects whose tokens are recovered (by $\geq 80\%$ token overlap) in the extractor's subjects or predicates. **Faithfulness** is the fraction of extracted-triple objects whose numeric values (or all non-trivial tokens) also appear in the source summary, measuring whether the extractor hallucinates values not grounded in the input.

LLM-extracted KG	Spearman ρ	Kendall τ	p-value (ρ)
gpt-5-nano	**+0.822**	+0.689	2.5×10^{-8}
Qwen3.5-27B	**+0.711**	+0.571	1.1×10^{-5}
Gemma-4-26B	**+0.830**	+0.666	1.4×10^{-8}

Table 8: Rank correlation between TabRex scores computed with the gold summary KG and with each LLM-extracted summary KG across the 30 validation samples.

$39.43/40.04/39.98$). Rank correlations are strong and highly significant for all three extractors, including two open-weight models from different families. The final TabRex score is therefore stable under extractor choice-the LLM extractor is a well-controlled component rather than a metric threat.

D Prompt Templates

Prompt B: Perturbation Planning

````
# Allowed Types by Group (overview mapping)
```python
group_to_types = {
    "0": ["header_shuffle", "reorder_columns", "reorder_rows", "columns_to_rename", "rows_to_rename", "data_type_change", "unit_conversion", "paraphrased_cell_values",],

    "1": ["columns_to_delete", "rows_to_delete", "add_columns", "add_rows", "column_disintegration", "columns_merge", "structure_change", "slight_data_differences", "precision_change", "misspellings", "data_swap", "add_symbols", "remove_symbols"],
    }
```
# System Prompt
You are an expert Python programmer. Generate two outputs - a Python function and a JSON array - separated by a marker. Core goal is to produce perturbed tables under two philosophies:

**Group 0 ("Semantically Identical")** - alter presentation without changing facts.
Allowed: `reorder_rows`, `header_shuffle`, `paraphrased_cell_values`, `data_type_change`, `unit_conversion`. Do not add/delete rows or columns.
Difficulty levels:  [easy --> one simple change, medium --> two combined changes, hard --> three or more complex changes]

**Unit Conversion Rules**
- Convert only when the unit is in cell text (e.g., ``12 km --> 7.456 mi").
- Update headers and recompute totals if affected.
- Ensure |v - f^{-1}(v')| <= max(1e-6, 0.001*|v|).

**Paraphrase & Format Rules**
- Preserve meaning, entities, and tokens.
- Format/rounding variation <= 0.1%.
- Totals/percentages must stay numerically identical.

**Group 1 ("Semantically Different")** - break meaning and falsify facts. Combine weak perturbations (`misspellings`, `precision_change`) only with strong ones (`data_swap`, `delete_rows`, etc.).

**Quantified Impact**
- `slight_data_differences`: +5-10% (easy), +20-50% (hard).
- `data_swap`: swap entire columns.
- `add_rows` / `delete_rows`: modify >= 20% of rows.
- `delete_columns`: remove key or total column.

## Output Specification

Your output must follow this structure exactly:
1. Python block defining `apply_perturbations()` (closed by ```).
2. Separator line:  `---JSON---`
3. Raw JSON array (no markdown fences).

### Python Section
Define:
```python
def apply_perturbations():
    ...
```
It must return a list of dicts with: `{'perturbed_table': <markdown>, 'metadata': <object>}`
**Metadata fields:** `slot_id`, `group`, `difficulty`, `selected_types`, `applied_order`.
Use helpers only:
`create_markdown_table`, `safe_float`, `add_noise`, `safe_round`, `parse_markdown_table`.

Always use `parse_markdown_table(markdown_text)` - never manual ``|" splitting. Preserve empty headers and columns. Check for `None` after `safe_float` before math.
**Operation order:**
1. Structural --> 2. Layout --> 3. Naming --> 4. Content/format
Avoid conflicts:  Don't apply `add_symbols` + `remove_symbols` in one slot. Don't rename then delete the same column. Perform merges before renames.

**Result Example**
```python
results.append({
  "perturbed_table": create_markdown_table(headers, data_rows),
  "metadata": {"slot_id": slot_id, "group": group, "difficulty": difficulty, "selected_types": selected_types, "applied_order": applied_order}
})
```

# User Prompt (generation-time wrapper)
Here is the markdown table and the `{len(plan_slots)}` plan slots to implement. Return a list of dicts, each with `perturbed_table` and `metadata`.

```markdown
{markdown_table}
```
````

Prompt C: Text2Graph

```
# System Prompt
You are a precise data structuring agent. Convert information from any source (text or table) into a standardized knowledge
      ↪ graph of [subject, property, value] triplets.

--- THE GOLDEN RULE: STANDARDIZED GRAMMAR ---
Follow this strict grammar for every triplet.

1. The subject is the PRIMARY ENTITY:
   - Choose the main entity the fact is about (e.g., "aboriginal population", "non-aboriginal population").
   - ENTITY-CENTRIC MODELING: Prefer specific entities (years, items, categories) over general ones.
     - GOOD: "2020", "product_a", "category_x"
     - AVOID: "company_data", "financial_info"
   - For time-based data: use the time period as the subject (e.g., "2020", "q1_2021", "january").
   - For categorical data: use the category as the subject (e.g., "electronics", "clothing", "services").

2. The predicate is a NORMALIZED PROPERTY KEY:
   - Combine the core concept and its condition using underscores: concept_condition.
   - Use lowercase throughout.
   - Maintain consistent patterns for similar concepts (e.g., "revenue_2020", "revenue_2021").
   - Do not add prefixes like "total_", "combined_", "gross_".
   - Examples:
     - Core: "participation rate", Condition: "married or common-law" --> `participation_rate_married_or_common_law`
     - Core: "employment rate difference", Condition: "single or previously married" --> `
      ↪ employment_rate_difference_single_or_previously_married`

3. The object is the CLEAN VALUE:
   - Use the most atomic data point (e.g., "81.9%", "7.0 percentage points").
   - Preserve units and formatting.
   - Use "-" if missing.

--- EXAMPLE OF APPLYING THE GRAMMAR ---

Source: "For the non-aboriginal population, the unemployment rate for those who are single or previously married was 8.2%."
Step 1: Subject --> "non-aboriginal population"
Step 2: Predicate --> `unemployment_rate_single_or_previously_married`
Step 3: Object --> "8.2%"
Final Triplet--> ["non-aboriginal population", "unemployment_rate_single_or_previously_married", "8.2%"]

--- ENTITY-CENTRIC MODELING EXAMPLES ---

GOOD (time-based):
["2020", "debt_amount", "100000"]
["2021", "debt_amount", "120000"]

BAD (concept-centric):
["debt_data", "amount_2020", "100000"]

GOOD (category-based):
["electronics", "sales_volume", "50000"]
["clothing", "sales_volume", "30000"]

BAD (concept-centric):
["product_sales", "electronics_volume", "50000"]
["product_sales", "clothing_volume", "30000"]

--- STRICT FORMAT CHECKLIST ---
- Output ONLY JSON arrays of [subject, predicate, object].
- Each triplet must have exactly 3 elements.
- Do NOT paraphrase, re-case, or stem subjects/predicates - copy labels verbatim from the source.
- Preserve punctuation, spaces, and capitalization.
- Use "-" for unknown values.
- Do NOT invent or omit data.
- Examples:
  - GOOD: ["2014", "copenhagen_shipment_volume", "448.6 million"]
  - BAD: ["2014", "copenhagen_shipment_volume"]
  - BAD: {"subject": "2014", "predicate": "...", "object": "..."} (wrong format)


# User Prompt
Follow the standardized grammar from the System Prompt to convert the given input into triplets.

CRITICAL REMINDERS:
1. Use ENTITY-CENTRIC modeling make specific years, categories, or items the subjects.
2. For time-based data: use years or periods (e.g., "2020", "q1_2021").
3. For categorical data: use categories (e.g., "electronics", "clothing").
4. Avoid using one central subject for all facts.
5. Use consistent, minimal predicates (e.g., "amount", "value", "count").
6. Use "-" for missing data.

Task Input:
Input Summary:
{summary}
```

Prompt D: Graph Alignment

````
# System Prompt
You are a structured reasoning engine comparing two knowledge graphs: **T1 (summary_graph)** and **T2 (table_graph)**.
Your goal: align their triplets `[subject, predicate, object]` semantically and output a structured JSON comparison.


### ALIGNMENT PRINCIPLES
Content-first alignment - never by position.
Example: match ``Product Alpha" <--> ``Product Alpha" even if order differs.
- Match facts based on meaning (subject/predicate semantics).
- Report unmatched ones as Missing (MI) or Extra (EI).


### OUTPUT STRUCTURE
Each aligned pair becomes:
```json
{
  "aligned_triplet": ["subject1/subject2", "predicate1/predicate2", "object1/object2"],
  "object_metadata": {"datatype": "...", "entitytype": "...", "unit": "...", "difference": "...", "missing_extra_info": "..."}
}
```

Rules:
- Copy strings verbatim from source (no rephrasing, re-casing, or normalization).
- Always include `/` in each component (use `-` for missing).
- Allow cross-component semantic matches (e.g., ``2014" <--> ``shipment_volume_2014").
- Each source triplet is used once.


### OBJECT METADATA
Compute difference and add context:
- If numeric with scale units (thousand/million/billion) --> convert and return absolute numeric difference.
  - `"448.6 million"` vs `"449 million"` --> `"difference": "400000"`
- If non-numeric or missing --> `"difference": "-"`
- Mark `"missing_extra_info"` as `"MI"` or `"EI"` when one side absent.

Range Handling: Single vs range --> min distance. Overlapping ranges --> `"difference": "0"`.

### OUTPUT RULES
- Output ONLY JSON matching `FinalComparisonResult`.
- Every `aligned_triplet` has exactly three entries (each with one `/`).
- `difference` is numeric-only (no text, commas, or units).
- Use `"-"` for truly unmatched components.

### Example
Input:
```json
T1: [["2014", "copenhagen_shipment_volume", "448.6 million"]]
T2: [["copenhagen", "shipment_volume_2014", "449 million"], ["copenhagen", "average_growth_rate", "6.96%"]]
```

Output:
```json
{
  "aligned_facts": [
    {
      "aligned_triplet": ["2014/copenhagen", "copenhagen_shipment_volume/shipment_volume_2014", "448.6 million/449 million"], "object_metadata": {"difference": "400000", "missing_extra_info": "None"}
    },
    {
      "aligned_triplet": ["copenhagen/copenhagen", "-/average_growth_rate", "-/6.96%"], "object_metadata": {"difference": "-", "missing_extra_info": "EI"}
    }
  ]
}
```

# User Prompt
Apply the alignment process above to the following graphs:
Input Graph T1 (summary_graph):
```json
{json.dumps(summary_graph, indent=2)}
```
Input Graph T2 (table_graph):
```json
{json.dumps(table_graph, indent=2)}
```
Unit Hints (optional):
```json
{_unit_hints_json}
```
Use hints only for internal numeric scaling (e.g., "thousand" --> *1,000). Do not modify string outputs.
````