# An Empirical Investigation of Robustness in Large Language Models under Tabular Distortions

**Anonymous ACL submission**


## Abstract

Distortions are a natural consequence of tables collected from real-world sources, where scraping, extraction, or manual intervention can introduce semantic and structural inconsistencies. Humans can often identify and mentally correct such errors autonomously before reasoning over the data. In this work, we investigate whether Large Language Models (LLMs) possess a similar capability by evaluating them on 506 distorted table question answering (TQA) samples derived from canonical tables. Our findings reveal systematic weaknesses in current LLM pre-training paradigms, where models exhibit a strong inductive bias toward assuming tables are well-formed, often overlooking distortions or discarding inconsistent entries instead of repairing them. Even frontier models like GPT-5.4 and Opus-4.6 achieve only 68% accuracy under distortion. While distortion-aware prompting partially mitigates these failures, errors persist. Through lightweight finetuning, we demonstrate that models can learn to autonomously detect and repair distorted tables in a human-like manner.


## 1 Introduction

LLMs have shown strong performance on TQA tasks, effectively handling both canonical and diverse table representations across different input modalities (Zhou et al., 2025). In practice, however, tables are often imperfect (Zhu et al., 2025). Formatting errors, misaligned rows or columns and semantic inconsistencies often arise during data collection, conversion between formats (PDF, Excel) (Oro and Ruffolo, 2009), scraping from web (Balakrishnan et al., 2015), OCR from images, etc. While such tables may appear "broken," they often retain a latent internal structure that humans can easily recognize and repair before using them for downstream applications (Figure 1). In this work, we investigate if LLMs can autonomously detect and correct such "weird but solvable" tables.

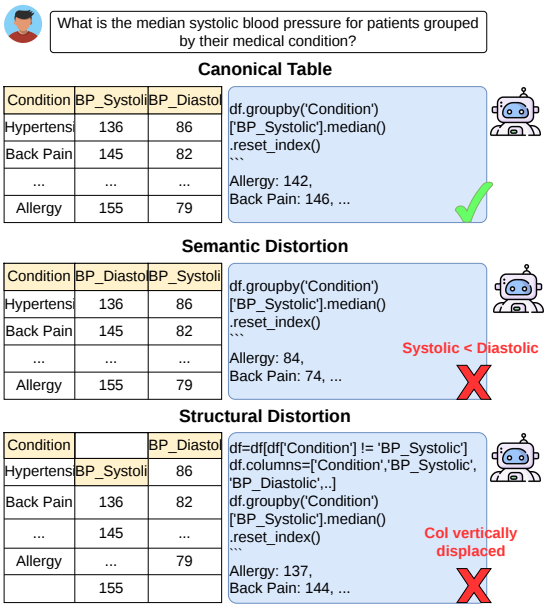


Figure 1: LLMs often overlook distortions in tables. Illustration of *structural* and *semantic* table distortions and their influence on how LLMs reason.

Our study shows error-aware reasoning over tables is not a learned default behavior: when tables are distorted, model accuracy degrades sharply, even when the distortions are visually or semantically evident. Only when models are provided with an explicit prior, via distortion-aware system prompts, do they adjust their reasoning strategies, and even then, recovery is inconsistent and incomplete. To systematically study this phenomenon, we curate a dataset of 506 TQA tasks designed to isolate distortion-related failures. Each canonical table is transformed into multiple *semantic* and *structural* distorted variants, while preserving the original answer, allowing us to directly measure robustness under controlled errors. Through evaluations across model families, input modalities, and execution settings, we uncover consistent failure patterns, most notably a widespread inabil-

ity to handle vertical structural shifts that require reasoning about global table layout. Our findings suggest a fundamental gap in current table understanding capabilities: LLMs either fail to detect or treat distortions as noisy inputs to be cleaned rather than as misaligned structures to be repaired. To alleviate this inductive bias in LLMs, we demonstrate supervised finetuning with distortion-aware reasoning traces can encourage models to inspect table structure, question computation validity, and autonomously repair inconsistencies — behaviors that humans perform inherently.

## 2 Distortions

We define distortions as operational transformations applied to canonical representations that introduce errors or inconsistencies that makes tables logically incorrect. Unlike noise, which is random and unstructured and therefore difficult to recover from, distortions are patterned perturbations that preserve the underlying semantic information and can therefore be repaired or recovered. For the scope of this paper, we restrict ourselves to single operational distortions under two headings – **semantic** and **structural**.

### 2.1 Semantic Distortion

Semantic Distortions alter the semantic interpretation of a table by introducing logical or relational inconsistencies. For example, the second table in Figure 1 contains a *BP_Systolic* value lower than *BP_Diastolic*, which violates real-world medical constraints. Consequently, beyond solving the task, LLMs must also identify such inconsistencies using world knowledge and domain understanding. Typical semantic distortions considered in this study include: (i) character misrepresentation (e.g., $5 \leftrightarrow S$); (ii) insertion of stray characters (e.g., `2‘35`); (iii) decimal separator swaps (e.g., `2,345.65` ↔ `2.345,65`); and (iv) label mismatches (e.g., weight measured in cubic centimeters). More details in Appendix B.1.

### 2.2 Structural Distortion

Structural distortion disrupts the spatial alignment between headers, rows, and cells, thereby introducing inconsistencies in the table structure. For instance, in the third table of Figure 1, the *BP_Systolic* column is vertically shifted downward, causing misalignment with the remaining columns. To correctly solve such tables, the evaluated model must reason about the underlying table geometry and verify whether the structural alignment is semantically consistent before performing computation. Typical structural distortions considered in this study include: (i) *vertical shift*, where a subset of columns is vertically misaligned than the rest; (ii) *horizontal shift*, where certain rows are horizontally displaced relative to the headers; (iii) *content merge*, where multiple row, column or cell values are incorrectly merged into a single entry; (iv) *content split*, where a single entry is fragmented across multiple rows or columns; and (v) *footnote injection*, where footnote markers are inserted within the table content. More details in Appendix B.2.

## 3 Dataset Construction

We construct our dataset[1] by programmatically introducing controlled distortions that change the structural layout and semantic consistency of tables. We generate a collection of 506 *(table, query, answer)* distortions derived from 316 canonical tables from WikiTQ (Pasupat and Liang, 2015). The distorted samples are subsequently validated through a round of LLM-as-Judge followed by Human evaluation, yielding 265 structurally distorted and 241 semantically distorted instances (Appendix A). Importantly, the original query–answer pairs are preserved throughout the transformation process, ensuring that our evaluation isolates the ability of LLMs to identify, reason about, and recover from distortions, rather than being confounded by variations in question complexity or answer distribution.

## 4 Experimental Setup

**Models** Our evaluation covers LLMs over three broad categories: (1) **Finetuned**: TableGpt2-7B (Su et al., 2024), TableLLM-7B (Zhang et al., 2025a); (2) **Open-source**: Qwen2.5-VL-32B (Team, 2025), Deepseek-R1-Distill-Qwen-32B (DeepSeek-AI, 2025); (3) **Close-source**: GPT-5.4 (reasoning: medium) (OpenAI, 2026), Claude-Opus-4.6 (Anthropic, 2026).

**Setup.** The above models are evaluated both with and without access to a Docker-based code execution sandbox that supports file uploads and Python execution. This setup enables a fair comparison between models that rely primarily on direct reasoning and those that solve tasks through code execution. For models without native tool- or function-calling capabilities, generated Python scripts are

[1] Full dataset available at `https://github.com/AIML-Researcher/table-distortion`

| Model | Input | Can. | Dist. Unaware | | | | Dist. Aware | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Distorted | | | Rbst. | Distorted | | | Rbst. |
| | | | Sem† | Str‡ | **All** | | Sem† | Str‡ | **All** | |
| **Table-Finetuned Large Language Models** | | | | | | | | | | |
| TableLLM-7B | Text 🔧 | 34.79 | 12.00 | 3.97 | 7.76 | 22.30 | 11.40 | 3.53 | 7.25 | 20.84 |
| TableGPT2-7B | Text 🔧 | 46.96 | 14.41 | 7.63 | 10.84 | 23.08 | 19.07 | 10.73 | 14.69 | 31.28 |
| **Open-sourced Large Language Models** | | | | | | | | | | |
| Qwen2.5-VL-32B | Text 🔧 | 53.66 | 24.90 | 19.32 | 21.98 | 40.96 | 27.92 | 17.87 | 22.66 | 42.23 |
| Qwen2.5-VL-32B | Image 🔧 | 51.11 | 24.17 | 15.97 | 19.88 | 38.90 | 20.83 | 18.32 | 19.52 | 38.19 |
| Deepseek-R1-Distill Qwen-32B | Text | 74.70 | 60.42 | 50.57 | 55.25 | 73.96 | 60.83 | 51.52 | 55.95 | 74.90 |
| **Close-sourced Large Language Models** | | | | | | | | | | |
| GPT-5.4 | None 🔧 | 83.96 | 68.05 | 61.45 | 64.61 | 76.95 | 82.57 | 67.05 | 74.46 | 88.69 |
| GPT-5.4 | Text | 81.42 | 67.63 | 54.72 | 60.87 | 74.76 | 72.20 | 56.98 | 64.23 | 78.89 |
| GPT-5.4 | Text 🔧 | 83.99 | 69.71 | 60.75 | 65.02 | 77.41 | 82.16 | 66.79 | 74.11 | 88.24 |
| GPT-5.4 | Image | 70.36 | 63.90 | 53.21 | 58.30 | 82.86 | 67.50 | 53.58 | 60.20 | 85.56 |
| GPT-5.4 | Image 🔧 | 81.62 | 69.58 | 62.26 | 65.74 | 80.54 | 78.01 | 66.04 | 71.74 | 87.89 |
| Claude-Opus-4.6 | None 🔧 | 88.32 | 80.91 | 69.81 | 75.10 | 85.03 | 87.97 | 78.87 | 83.20 | 94.20 |
| Claude-Opus-4.6 | Text | 87.98 | 73.22 | 71.76 | 72.46 | 82.36 | 78.33 | 74.72 | 76.44 | 86.88 |
| Claude-Opus-4.6 | Text 🔧 | 87.55 | **84.23** | **73.96** | **78.85** | **90.06** | 90.46 | **78.93** | **84.46** | **96.47** |
| Claude-Opus-4.6 | Image | 85.94 | 66.67 | 59.92 | 63.13 | 73.46 | 73.84 | 62.88 | 68.06 | 79.19 |
| Claude-Opus-4.6 | Image 🔧 | 87.35 | 83.40 | 68.30 | 75.49 | 86.42 | **91.25** | 77.36 | 83.96 | 96.12 |

Table 1: **Shows accuracy degradation (%) when answering identical queries over canonical versus distorted tables.** Distortions are categorized into semantic (Sem†) and structural (Str‡). Robustness ($Rbst\% = Dist./Can.$) quantifies performance preservation under distortion. Models are evaluated across *None*, *Text* and *Image* modes, with and without code sandbox (🔧). Variance across 5 iterations $\approx 0.6857$.

executed separately within the sandbox, and the output used for evaluation. Furthermore, tabular context is provided under three different modalities– (1) **None**: the table file is directly uploaded to the code sandbox without giving any preview in the prompt; (2) **Text**: a markdown representation is provided; (3) **Image**: a PNG image of the entire table is provided. This helps us study whether input modality influences how models detect and handle distortions. We use Markdown instead of other textual forms, like CSV, as it better preserves the structural orientations we cover in the dataset.

**Metrics.** We report performance using exact-match (Pasupat and Liang, 2015). Additionally, we measure Robustness (%) as the ratio of accuracy on distorted tables to accuracy on their corresponding canonical representations. This metric captures the extent to which model performance is preserved under distortion, while also enabling fair comparison across models with varying baseline accuracies.

## 5 Evaluation

Table 1 reports model accuracy for both Canonical and Distorted forms. Table-Finetuned LLMs exhibit the weakest robustness, showing negligible recovery under structural distortions, suggesting that table headers and row-level samples as context during training does not help detect layout-level information. Among open-source models, Qwen2.5 shows moderate robustness ($\approx 40\%$), while DeepSeek performs substantially better (>50%). Closed-source models such as GPT-5.4 and Opus-4.6 perform strongest overall, but still reach only up to 78.85% accuracy under distortion. Across input modalities, Text generally achieves higher accuracy than Image, with this gap being more prominent for structural than semantic, suggesting that current image-based pretraining approaches are ineffective in capturing structural irregularities in tables.

### 5.1 Prompt Steering

We incorporate distortion-handling instructions (exact prompt in Appendix E) in the system prompt and evaluate LLMs. Table-Finetuned and Open-sourced models show little to no improvement, indicating limited capability despite explicit awareness. In contrast, closed-source models demonstrate substantially stronger distortion-awareness, improving average robustness by 7.35% for GPT-5.4 and 7.11% for Opus-4.6 across all input modalities, reaching a best of 96.47% by the latter. However, the gains are considerably larger for semantic than

for structural (+7.64% versus +4.71%), suggesting that structural inconsistencies remain significantly more challenging to detect and repair. This trend is particularly evident in vertical column shifts, where even the best-performing configuration, Opus-4.6 (Text 🔧), achieves only 57.38% (Table 6).

### 5.2 Distortion-aware Finetuning

Motivated by the limitations of prompt steering on Open-sourced models, we fine-tune Qwen2.5-1.5B-Instruct using supervised fine-tuning (SFT) with QLoRA (Dettmers et al., 2023) for 3 epochs under a 5-fold cross validation setup with 80/20 train–test split. All experiments are conducted on a single NVIDIA Tesla T4 (16 GB) GPU. Our training set includes both canonical and distorted tables so the model learns to first detect distortions before attempting repairs, rather than implicitly assuming every table is corrupted. The training outputs follow a structured reasoning format in which the model first inspects the table for structural and semantic consistency, carefully reasons about potential inconsistencies during computation, repairs them if detected, and finally show the answer (more details in Appendix D).

| Qwen2.5-1.5B | Can. | Distorted | | | Rbst. |
|---|---|---|---|---|---|
| | | Sem† | Str‡ | All | |
| Base | 26.09 | 18.97 | 17.65 | 18.48 | 70.83 |
| Finetuned | **31.52** | **22.07** | **25.29** | **23.26** | **73.79** |
| | ± 1.33 | ± 5.64 | ± 9.44 | ± 6.55 | – |

Table 2: Finetuning improves success under distortion.

Table 2 shows that lightweight finetuning improves distortion handling in LLMs, with a 4.78% rise over its base counterpart. Detection and repair of *vertical_shift* increases (+10%), while *label_mismatch*, *content_merge*, and *content_split*, which failed completely, showed noticeable recovery after finetuning (Table 3). Accuracy on canonical tables also improves, suggesting that the model learnt robust reasoning behavior rather than memorizing distortion patterns. The high standard deviations suggest that some folds fail to consistently learn these signals, highlighting the sensitivity of low-resource fine-tuning. However, they provide promising evidence that models can be trained to autonomously handle distortions.

| Distortions | Base (%) | Finetuned (%) |
|---|---|---|
| **Semantic Distortions** | | |
| character_misinterpretation | 17.24 | 15.17 ± 6.26 |
| stray_characters_insertion | 25.00 | 30.00 ± 11.18 |
| label_mismatch | 0.00 | 20.00 ± 14.49 |
| decimal_separator_swap | 31.25 | 33.75 ± 5.59 |
| **Structural Distortions** | | |
| vertical_shift | 16.67 | 26.67 ± 14.91 |
| horizontal_shift | 45.45 | 36.36 ± 17.01 |
| content_merge | 0.00 | 14.55 ± 10.37 |
| content_split | 0.00 | 28.00 ± 10.95 |
| footnote_injection | 0.00 | 0.00 ± 0.00 |

Table 3: Performance comparison of Finetuned against base model across different distortion types.

## 6 Related Works

Prior work has extensively studied table question answering (Chen et al., 2020; Yavuz et al., 2018) focusing on how models reason over tabular content under different representations (Sui et al., 2024; Wu et al., 2025) and input modalities (Zheng et al., 2024; Yang et al., 2025), typically assuming that tables are canonical and well-formed (Herzig et al., 2020; Liu et al., 2022; Jiang et al., 2022). More recent studies have begun to examine robustness by introducing controlled perturbations (Bhandari et al., 2025), while RobuT (Zhao et al., 2023) adversarially alter table rows and columns without changing the underlying semantics to evaluate model stability. Other works compare performance across alternative but valid table representations for the same query (Zhang et al., 2025b), or introduce explicit mechanisms to filter noise and spurious query clauses (Ye et al., 2025). We study rule-based errors and how they can be mitigated towards autonomous rectification than rejection.

## 7 Conclusion

Our evaluation reveals that current LLMs exhibit a strong inductive bias toward understanding tables which prevent them from identifying and repairing both semantic and structural inconsistencies. While distortion-aware prompting improves robustness, recovery remains incomplete and inconsistent. Through lightweight fine-tuning, we demonstrate that these limitations can be alleviated, providing promising evidence that models can be trained to autonomously detect and repair distorted tables, resembling human-like intelligence.

## Limitations

In this study, we focus on single-step distortions rather than multiple compounding transformations. This controlled setup allows us to isolate and analyze the effect of individual distortion patterns without introducing additional confounding factors. Consequently, our evaluation covers only a subset of the much broader space of distorted tables encountered in real-world settings. Furthermore, we restrict our analysis to distortions that preserve the original answer to the question. This design choice enables us to specifically measure the impact of distortions on model reasoning and table understanding without conflating it with changes in question difficulty or answer computation complexity. Extending the analysis to cover a broader space remains an important direction for future work.

## Ethical Considerations

Our dataset was constructed by taking samples from WikiTQ (Pasupat and Liang, 2015) which is publicly available under the license CC-BY-SA-4.0[2]. Human evaluation was conducted across 5 participants from an anonymous IT company. The experts were compensated on a per-query basis at a rate of $2 per query. None of the distortions introduced in this study were designed to target or reinforce biases related to race, gender, ethnicity, or other protected attributes. This was further approved by the ethics review board by the anonymous IT company. All evaluation experiments can be run on a 64GB RAM or a single NVIDIA Tesla V100-32G GPU. Finetuning was done on a single NVIDIA Tesla T4 (16 GB) GPU.

A key societal concern highlighted by our findings is the risk of *silent failures*, where models generate confident but incorrect outputs under table distortions (Figure 1). Such failures are particularly dangerous in high-stakes domains like healthcare and finance, where row or column misalignments can alter the meaning of critical values. Our results show that even frontier models remain vulnerable to structural perturbations, emphasizing the need for verification mechanisms and human oversight in safety-critical applications.

ChatGPT was used solely for grammar polishing and improving writing clarity during the preparation of this paper. It was not involved in the formulation of hypotheses, experimental design, analysis, or the scientific discoveries and conclusions presented in this work.

[2] https://creativecommons.org/licenses/by-sa/4.0/

## References


Anthropic. 2026. Claude opus 4.6. https://www.anthropic.com/news/claude-opus-4-6. Accessed: 2026-05-22.

Sreeram Balakrishnan, Alon Y Halevy, Boulos Harb, Hongrae Lee, Jayant Madhavan, Afshin Rostamizadeh, Warren Shen, Kenneth Wilder, Fei Wu, and Cong Yu. 2015. Applying webtables in practice. In *CIDR*.

Kushal Raj Bhandari, Sixue Xing, Soham Dan, and Jianxi Gao. 2025. Exploring the robustness of language models for tabular question answering via attention analysis. *Preprint*, arXiv:2406.12719.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. 2022. Omnitab: Pretraining with natural and synthetic data for few-shot table-based question answering. *Preprint*, arXiv:2207.03637.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022. Tapex: Table pre-training via learning a neural sql executor. *Preprint*, arXiv:2107.07653.

OpenAI. 2026. Introducing gpt-5.4. https://openai.com/index/introducing-gpt-5-4/. Accessed: 2026-05-22.

Ermelinda Oro and Massimo Ruffolo. 2009. Pdf-trex: An approach for recognizing and extracting tables from pdf documents. In *2009 10th International Conference on Document Analysis and Recognition*, pages 906–910.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. *Preprint*, arXiv:1508.00305.

Aofeng Su, Aowen Wang, Chao Ye, Chen Zhou, Ga Zhang, Gang Chen, Guangcheng Zhu, Haobo Wang, Haokai Xu, Hao Chen, Haoze Li, Haoxuan Lan, Jiaming Tian, Jing Yuan, Junbo Zhao, Junlin Zhou, Kaizhe Shou, Liangyu Zha, Lin Long, and 14 others. 2024. Tablegpt2: A large multimodal model with tabular data integration. *Preprint*, arXiv:2411.02059.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. *Preprint*, arXiv:2305.13062.

Qwen Team. 2025. Qwen2.5-vl.

Pengzuo Wu, Yuhang Yang, Guangcheng Zhu, Chao Ye, Hong Gu, Xu Lu, Ruixuan Xiao, Bowen Bao, Yijing He, Liangyu Zha, Wentao Ye, Junbo Zhao, and Haobo Wang. 2025. Realhitbench: A comprehensive realistic hierarchical table benchmark for evaluating llm-based table analysis. *Preprint*, arXiv:2506.13405.

Bohao Yang, Yingji Zhang, Dong Liu, André Freitas, and Chenghua Lin. 2025. Does table source matter? benchmarking and improving multimodal scientific table understanding and reasoning. *Preprint*, arXiv:2501.13042.

Semih Yavuz, Izzeddin Gur, Yu Su, and Xifeng Yan. 2018. What it takes to achieve 100% condition accuracy on WikiSQL. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1702–1711, Brussels, Belgium. Association for Computational Linguistics.

Shenghao Ye, Yu Guo, Dong Jin, Yikai Shen, Yunpeng Hou, Shuangwu Chen, Jian Yang, and Xiaofeng Jiang. 2025. When tableqa meets noise: A dual denoising framework for complex questions and large-scale tables. *Preprint*, arXiv:2509.17680.

Xiaokang Zhang, Sijia Luo, Bohan Zhang, Zeyao Ma, Jing Zhang, Yang Li, Guanlin Li, Zijun Yao, Kangli Xu, Jinchang Zhou, Daniel Zhang-Li, Jifan Yu, Shu Zhao, Juanzi Li, and Jie Tang. 2025a. Tablellm: Enabling tabular data manipulation by llms in real office usage scenarios. *Preprint*, arXiv:2403.19318.

Yue Zhang, Seiji Maekawa, and Nikita Bhutani. 2025b. Same content, different representations: A controlled study for table qa. *Preprint*, arXiv:2509.22983.

Yilun Zhao, Chen Zhao, Linyong Nan, Zhenting Qi, Wenlin Zhang, Xiangru Tang, Boyu Mi, and Dragomir Radev. 2023. RobuT: A systematic study of table QA robustness against human-annotated adversarial perturbations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6064–6081, Toronto, Canada. Association for Computational Linguistics.

Mingyu Zheng, Xinwei Feng, Qingyi Si, Qiaoqiao She, Zheng Lin, Wenbin Jiang, and Weiping Wang. 2024. Multimodal table understanding. *Preprint*, arXiv:2406.08100.

Wei Zhou, Bolei Ma, Annemarie Friedrich, and Mohsen Mesgar. 2025. Table question answering in the era of large language models: A comprehensive survey of tasks, methods, and evaluation. *Preprint*, arXiv:2510.09671.

Junnan Zhu, Jingyi Wang, Bohan Yu, Xiaoyu Wu, Junbo Li, Lei Wang, and Nan Xu. 2025. Tableeval: A real-world benchmark for complex, multilingual, and multi-structured table question answering. *Preprint*, arXiv:2506.03949.

## A Dataset Construction

In this section, we discuss the specifics of how tables from WikiTQ were transformed to created a verified distorted set of 506 samples for evaluation. We sampled 500 *(table, query, answer)* triplets from the validation split from WikiTQ and applied controlled programmatic transformations to insert different distortion types at locations that would impact the answer computation targets in the table. With these deterministic disturbances added to create multiple variants of the same canonical table, we passed them on to an LLM-as-Judge stage to review if the transformations: (i) does not violate the internal semantics or implied meaning of the table, (ii) retains the query–answer relevance from the canonical representation, and (iii) alters the computation strategy for the given answer. We used Claude-Opus-4.5 model for this purpose. The files that passed the above test was then validated by 5 human experts from an anonymous IT company to ensure correctness and relevance to humans from the stand-point of identifying and fixing these errors. After the manual filtration process, we concluded with 506 samples coming from 316 original canonical tables.

## B Distortion Types

In this section, we describe the various distortions applied to canonical tables in detail. These distortions commonly arise in tables curated or extracted from image and text documents, as well as from errors introduced during manual data entry. While the exact transformations vary depending on the table content, they can be broadly categorized into the following groups.

### B.1 Semantic

We cover 4 broad categories where tables in the wild can be distorted semantically.

**character_misinterpretation** Characters in cell values are swapped with visually similar ones (OCR-style confusions, e.g. O↔0, l↔1, rn↔m). See Figure 4.

**stray_characters_insertion** Random noise characters or symbols are inserted into otherwise valid table values. For example, artifacts such as stains or marks in scanned documents may appear as symbols like an asterisk (∗) within numeric entries, which are subsequently captured by OCR systems. This could also confuse an LLM in altering the scale of values, where for a certain label that only accepts whole numbers gets a floating point number in between.

| Distortion Type | Count |
|---|---|
| **Semantic** | |
| character_misinterpretation | 101 |
| stray_characters_insertion | 38 |
| label_mismatch | 32 |
| decimal_separator_swap | 70 |
| **Structural** | |
| vertical_shift | 61 |
| horizontal_shift | 56 |
| content_merge | 95 |
| content_split | 31 |
| footnote_injection | 22 |

Table 4: Distribution of distortion types across the 506 samples curated for this study.

**label_mismatch** Labels do not align with their corresponding values, or rules which break relation constraints inside the table. For example, systolic pressure scoring lower than diastolic pressure in Figure 1.

**decimal_separator_swap** Decimal and thousands separators are swapped (e.g. 1,234.56 ↔ 1.234,56), changing the numeric reading of a value without touching the digits themselves.

### B.2 Structural

We cover 5 broad categories where tables obtained in the wild can be distorted structurally.

**vertical_shift** Rows are nudged up or down so that values end up under the wrong row label (the row dimension is misaligned with its data).

**horizontal_shift** Columns are nudged left or right so values fall under the wrong header (the column dimension is misaligned with its data).

**content_merge** Multiple cells, rows, or columns are fused into one, collapsing information that was originally separated (see example Figure 2).

**content_split** A single logical table entity is broken across multiple physical rows or columns, fragmenting a record that should stay together (see example Figure 3).

**Canonical**

| Rank | Airport | Passengers handled | % Change 2011 / 12 |
|---|---|---|---|
| 1 | Netherlands - Amsterdam | 105,349 | - |
| 2 | United Kingdom - Belfast-International | 92,502 | - |
| 3 | Portugal - Faro | 71,676 | - |
| 4 | Spain - Barcelona | 66,565 | - |
| 5 | Spain - Alicante | 64,090 | - |
| 6 | Spain - Malaga | 59,175 | - |
| 7 | Ireland - Dublin | 35,524 | - |
| 8 | Jersey - Jersey | 35,169 | 1128 |
| 9 | Ireland - Waterford | 31,907 | 25 |
| 10 | Spain - Palma de Mallorca | 27,718 | - |

Question: between the topped ranked airport, netherlands - amsterdam, & spain - palma de mallorca, what is the difference in the amount of passengers handled?

**Content Merge Distortion**

| Rank | Airport | Passengers handled | % Change 2011 / 12 |
|---|---|---|---|
| 1 2 | Netherlands - Amsterdam United Kingdom - Belfast-International | 105,349 92,502 | - - |
| 3 | Portugal - Faro | 71,676 | - |
| 4 | Spain - Barcelona | 66,565 | - |
| 5 | Spain - Alicante | 64,090 | - |
| 6 | Spain - Malaga | 59,175 | - |
| 7 | Ireland - Dublin | 35,524 | - |
| 8 | Jersey - Jersey | 35,169 | 1128 |
| 9 | Ireland - Waterford | 31,907 | 25 |
| 10 | Spain - Palma de Mallorca | 27,718 | - |

Ans: 77,631

Figure 2: The model must realize that the first and the second row have been merged together and they are not a single row data.

**Canonical**

| Player | Year | Round | Result | Opponent |
|---|---|---|---|---|
| Raymond van Barneveld | 2009 | Quarter-Final | Won | Jelle Klaasen |
| Raymond van Barneveld | 2010 | 2nd Round | Won | Brendan Dolan |
| Adrian Lewis | 2011 | Final | Won | Gary Anderson |
| Dean Winstanley | 2013 | 2nd Round | Lost | Vincent van der Voort |
| Michael van Gerwen | 2013 | Semi Final | Won | James Wade |
| Terry Jenkins | 2014 | 1st Round | Lost | Per Laursen |
| Kyle Anderson | 2014 | 1st Round | Lost | Ian White |

Question: Who was the last to win against his opponent?

**Content Split Distortion**

| Player | Year | Round | Result | Opponent |
|---|---|---|---|---|
| Raymond van Barneveld | 2009 | Quarter-Final | Won | Jelle Klaasen |
| Raymond van Barneveld | 2010 | 2nd Round | Won | Brendan Dolan |
| Adrian Lewis | 2011 | | | |
| | | Final | Won | Gary Anderson |
| Dean Winstanley | 2013 | 2nd Round | Lost | Vincent van der Voort |
| Michael van Gerwen | 2013 | | | |
| | | Semi Final | Won | James Wade |
| Terry Jenkins | 2014 | 1st Round | Lost | Per Laursen |
| Kyle Anderson | 2014 | 1st Round | Lost | Ian White |

Ans: Michael van Gerwen

Figure 3: The model should align the displacement of partial columns into different rows correctly. This should not be interpreted as new data with missing fields.

**Canonical**

| Country | Masters | U.S. Open | The Open | PGA | Total |
|---|---|---|---|---|---|
| United States | 57 | 80 | 42 | 79 | 258 |
| Scotland | 1 | 12 | 41 | 0 | 54 |
| England | 3 | 8 | 22 | 2 | 35 |
| South Africa | 5 | 5 | 10 | 2 | 22 |
| Australia | 1 | 2 | 9 | 4 | 16 |
| Jersey | 0 | 2 | 7 | 0 | 9 |

Question: Is the United States or Scotland better?

**Character Misinterpretation Distortion**

| Country | Masters | U.S. Open | The Open | PGA | Total |
|---|---|---|---|---|---|
| United States | S7 | 80 | 4Z | 79 | Z58 |
| Scot1and | 1 | 12 | 41 | O | S4 |
| Eng1and | 3 | 8 | 22 | Z | 3S |
| South Africa | 5 | 5 | I0 | 2 | Z2 |
| Australia | I | 2 | 9 | 4 | 16 |
| Jersey | 0 | 2 | 7 | 0 | 9 |

Ans: United States

Figure 4: The model must correctly map the country names and numerical values and not simply use a greater than logic and compute directly on the string.

**footnote_injection** Footnote markers, asterisks, or trailing reference notes are inserted into cells, polluting the value with non-data text. These usually get picked from pdf files or documents when the table spans multiple pages and when page number, header or footer can creep in.

## C Evaluation across Distortion Types

We report robustness across individual distortion categories in Tables 5 and 6. The results reveal substantial differences not only across model families, but also between semantic and structural perturbations.

Table-finetuned models continue to perform poorly across nearly all distortion categories. In the distortion unaware setting, both TableLLM and TableGPT2 achieve near-zero accuracy for several structural perturbations, particularly horizontal shifts and content splits. Even under distortion-aware prompting, improvements remain limited and are largely confined to semantic distortions. For example, TableGPT2 improves noticeably on semantic corruptions such as stray character insertion and label mismatch, but structural robustness remains extremely weak, with content split accuracy remaining fixed at only 3.23% and vertical shift accuracy stagnating at 6.67%. This further suggests that prompt adjustments does not help close the fundamental gaps of how these models were finetuned.

Among open-source models, Qwen2.5-VL and DeepSeekR1 exhibit moderate robustness gains under distortion-aware prompting, though their weaknesses remain distortion-specific. Vertical shifts remain particularly challenging across the board: Qwen2.5-VL achieves only 11.67% accuracy even in the aware setting, while DeepSeekR1 remains below 32%. Qwen additionally struggles with structural perturbations more broadly, especially horizontal shifts and content merges, where performance remains below 20% despite prompting. In contrast, DeepSeekR1 handles horizontal shifts substantially better, achieving close to 78% accuracy, suggesting stronger spatial reasoning capabilities. However, prompting yields only marginal gains overall, indicating that awareness alone is insufficient to reliably repair structural inconsistencies.

Closed-source frontier models demonstrate significantly stronger robustness across nearly all distortion types. Across GPT-5.4 and Opus-4.6 variants, semantic distortions such as character misinterpretation, label mismatch, and decimal separator swaps are handled reliably, with many settings exceeding 85–90% accuracy. Structural perturbations, however, remain comparatively harder. In particular, vertical shifts and content merges consistently emerge as the weakest categories across both model families. Even in distortion-aware settings, GPT-5.4 variants average only around 20-40% robustness on vertical shifts, while Opus-4.6, although substantially stronger, still shows noticeably lower performance on this distortion compared to other categories.

Distortion-aware prompting nevertheless provides meaningful gains for frontier models, particularly on semantic corruptions and localized structural inconsistencies. For instance, Opus-4.6 nearly saturates on horizontal shifts and character-level semantic distortions, with several configurations reaching or exceeding 90% robustness. Similarly, GPT-5.4 shows strong improvements on content merges and content splits under aware prompting. Interestingly, these gains are not uniform: vertical shift robustness slightly decreases for some GPT-5.4 configurations in the aware setting, suggesting that prompting can occasionally bias the model toward over-correction or erroneous table restructuring. Overall, the results indicate that while large frontier models can effectively utilize distortion-aware instructions, structural reasoning, particularly row and column alignment under global table perturbations, remains an unresolved challenge even for the strongest systems evaluated.

Furthermore, the choice of input modality noticeably affects robustness for specific distortion types. Text-based inputs consistently outperform image-based settings on semantic distortions such as label mismatch, stray character insertion, and decimal separator swaps, particularly for open-source models like Qwen2.5-VL, suggesting that OCR-induced noise compounds existing semantic corruption. In contrast, image inputs occasionally help preserve coarse structural layout information, leading to competitive or slightly improved performance on distortions such as horizontal shifts and content splits for some closed-source models. GPT-5.4 and Opus-4.6 generally achieve their strongest overall robustness under text-based prompting, while the "None" setting remains surprisingly competitive for semantic distortions, indicating that frontier models can often infer and repair localized inconsistencies without explicit tab-

ular content guidance. However, these gains do not extend uniformly across all structural perturbations. Vertical shifts remain consistently difficult across text, image, and prompt-aware settings alike, highlighting that maintaining global row-column alignment and relational consistency is a fundamental challenge that persists regardless of evaluation framework or modality.

## D Finetuning

We describe the implementation details of finetuning on the distorted table sampples.

**Training Setup.** We fine-tuned *Qwen2.5-1.5B-Instruct* using supervised fine-tuning (SFT) on a mixture of canonical and distorted tables. The 506 distorted samples were first split into an 80/20 train–test partition. Each distorted table was mapped back to its corresponding canonical representation, and both versions were assigned to the same split to avoid leakage. This resulted in a final dataset of 824 samples, with 647 samples used for 5-fold cross-validation during training (518 train and 129 validation samples per fold) and 177 held out for testing. Using both clean and distorted tables prevents the model from overfitting to explicit distortion patterns while encouraging it to distinguish between well-formed and corrupted tables.

Training targets followed a structured reasoning format consisting of three primary stages: SCAN, where the model checks whether the table structure appears logically consistent; COMPUTE, where it derives the answer; and CONCLUDE, where it states the final response. Two additional optional stages were introduced that fire only when the table warrants them: FLAG, if something does not look right, and REPAIR, that applies a fix for the flagged error. Distorted samples follow either one of the following trajectories during reasoning: *upfront* (the anomaly is obvious at first glance, so FLAG fires during SCAN), *mid-computation* (the anomaly only becomes apparent when reading the cell the question depends on, so FLAG fires during COMPUTE), or *none* (the distortion does not affect the cells needed to answer the question and is therefore not mentioned). Canonical samples follow *none* by default. This allows flexibility for the model to detect and de-distort even if something was not apparent superficially in the SCAN stage. When building the target output for training, during REPAIR, we provide only the relevant de-distortion definition for the current sample to reduce memorization and overfitting to other types.

**Implementation Details.** We fine-tuned *Qwen2.5-1.5B-Instruct* using QLoRA (Dettmers et al., 2023) with 4-bit NF4 quantization, double quantization, and fp16 compute precision. LoRA adapters (rank 16, $\alpha = 32$, dropout 0.05) were applied to all attention and MLP projection layers. Training was performed for 3 epochs using an effective batch size of 16, a cosine learning-rate schedule with peak learning rate $2 \times 10^{-4}$, and the `paged_adamw_8bit` optimizer. Experiments used 2048-token sequences, gradient checkpointing, and SDPA attention. All runs were conducted on a single NVIDIA Tesla T4 (16 GB) GPU. Figure 5 shows the training loss curve.

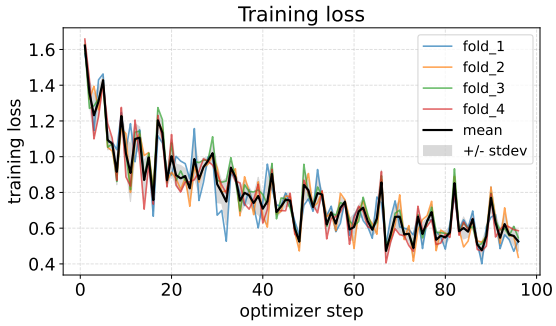


Figure 5: Fine-tuning loss curve.

**Evaluation.** We compare the fine-tuned model against its corresponding base model. Results in Table 2 and Table 3 report the mean and standard deviation across all five cross-validation folds for the fine-tuned model, while the baseline results correspond to greedily decoded outputs from the original base model.

## E Prompts Used

We have used standard prompt across all models and evaluation setups. For finetuned and open-source models, we followed prompt directions that were presented in their model card (TableGPT2-7B[3], TableLLM-7B[4]) to better reflect their capability. We used the prompt in Figure 6 for Distortion-Unaware and prompt in Figure 8 for a Distortion-Aware setting. For the setting where we do not provide a code execution sandbox to the models, we use prompts in Figure 7 and 9 respectively.

[3] https://huggingface.co/tablegpt/TableGPT2-7B
[4] https://huggingface.co/RUCKBReasoning/TableLLM-7b

You are an expert data analyst specializing in solving complex data analytics questions. Your role is to analyze datasets and provide accurate answers to user queries.

**Your Environment:**

- You have access to a code sandbox where you can execute Python scripts
- The relevant dataset (Excel or CSV file) has been uploaded to the sandbox
- You can write and run Python code to analyze the data and answer queries

**Your Responsibilities:**

1. Carefully analyze the user's query to understand the required information
2. Study the table structure and content to ensure accurate reasoning
3. Write clear and efficient Python code to extract insights
4. Execute the code in the sandbox environment
5. If execution fails, debug and retry up to five times
6. Provide a direct answer based on the analysis without suggesting approaches or partial solutions

Figure 6: Distortion-Unaware prompt given to models during evaluation.

You are an expert data analyst specializing in solving data analytics questions based on a tabular data. Your role is to analyze the data and provide accurate answers to user queries.

**Your Environment:**

- You will be provided with the relevant data context required to answer the query.

**Your Responsibilities**:

1. Carefully analyze the user's query to understand what information they need
2. Study the table structure and content to better understand how to answer queries accurately
3. Provide a direct answer based on your analysis. Do not suggest approaches or offer partial solutions

Figure 7: Distortion-UnAware prompt given to models during evaluation without the code execution sandbox.

You are an expert data analyst specializing in solving complex data analytics questions. Your role is to analyze datasets and provide accurate answers to user queries.

**Your Environment:**

- You have access to a code sandbox where you can execute Python scripts
- The relevant dataset (Excel or CSV file) has been uploaded to the sandbox
- You can write and run Python code to analyze the data and answer queries

**Your Responsibilities:**

1. Carefully analyze the user's query to understand the required information
2. Study the table structure and content to ensure accurate reasoning
3. Write clear and efficient Python code to extract insights from the dataset
4. Execute the code in the sandbox environment
5. If execution fails, debug and retry up to five times before concluding failure
6. Provide a direct answer based on the analysis without suggesting approaches or partial solutions
7. If the table exhibits structural or semantic inconsistencies (e.g., shifted rows or columns, semantic misalignment), correct these issues prior to analysis

Figure 8: Distortion-Aware prompt given to models during evaluation.

You are an expert data analyst specializing in solving data analytics questions based on a tabular data. Your role is to analyze the data and provide accurate answers to user queries.

**Your Environment:**

- You will be provided with the relevant data context required to answer the query.

**Your Responsibilities**:

1. Carefully analyze the user's query to understand what information they need
2. Study the table structure and content to better understand how to answer queries accurately
3. Provide a direct answer based on your analysis. Do not suggest approaches or offer partial solutions
4. If you encounter scenarios where the table seems incorrect either structurally or semantically (e.g., shifted rows, shifted columns, semantic misalignment, etc.), correct these issues first before proceeding with your analysis.

Figure 9: Distortion-Aware prompt given to models during evaluation without the code execution sandbox.

| Model | Input | Semantic | | | | Structural | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | CharMis | StrChar | LabMis | DecSwap | Vert | Horz | CMerge | CSplit | Foot |
| **Table-Finetuned Large Language Models** | | | | | | | | | | |
| TableLLM | Text | 10.53 | 11.43 | 15.62 | 12.70 | 3.39 | 0.00 | 6.32 | 0.00 | 9.52 |
| TableGPT2 | Text | 6.00 | 29.73 | 21.88 | 14.93 | 6.67 | 3.70 | 11.58 | 3.23 | 9.09 |
| **Open-source Large Language Models** | | | | | | | | | | |
| Qwen2.5VL | Text 🔧 | 10.89 | 55.26 | 28.12 | 27.14 | 9.84 | 14.29 | 23.40 | 19.35 | 40.91 |
| Qwen2.5VL | Image 🔧 | 13.86 | 47.37 | 31.25 | 23.19 | 3.28 | 12.50 | 21.51 | 22.58 | 27.27 |
| DeepseekR1 | Text | 53.00 | 73.68 | 43.75 | 71.43 | 27.87 | 78.57 | 43.16 | 58.06 | 63.64 |
| **Close-sourced Large Language Models** | | | | | | | | | | |
| GPT-5.4 | None 🔧 | 61.39 | 81.58 | 71.88 | 68.57 | 40.98 | 76.79 | 58.51 | 89.66 | 54.55 |
| GPT-5.4 | Text | 66.34 | 84.21 | 59.38 | 64.29 | 26.23 | 85.71 | 43.16 | 87.10 | 59.09 |
| GPT-5.4 | Text 🔧 | 64.36 | 89.47 | 53.12 | 74.29 | 36.07 | 73.21 | 56.84 | 93.55 | 68.18 |
| GPT-5.4 | Image | 71.29 | 81.58 | 43.75 | 52.86 | 29.51 | 69.64 | 47.37 | 80.65 | 63.64 |
| GPT-5.4 | Image 🔧 | 61.39 | 84.21 | 70.97 | 72.86 | 36.07 | 82.14 | 57.89 | 90.32 | 63.64 |
| Opus-4.6 | None 🔧 | 70.30 | 92.11 | 81.25 | 90.00 | 57.38 | 82.14 | 65.26 | 87.10 | 68.18 |
| Opus-4.6 | Text | 67.00 | 83.78 | 75.00 | 75.71 | 37.70 | 98.18 | 70.53 | 93.33 | 76.19 |
| Opus-4.6 | Text 🔧 | 79.21 | 94.74 | 75.00 | 90.00 | 52.46 | 100.00 | 67.37 | 87.10 | 77.27 |
| Opus-4.6 | Image | 59.00 | 80.00 | 62.50 | 72.86 | 37.70 | 83.64 | 52.69 | 77.42 | 68.18 |
| Opus-4.6 | Image 🔧 | 77.23 | 92.11 | 78.12 | 90.00 | 52.46 | 80.36 | 64.21 | 87.10 | 72.73 |

Table 5: Breakdown of model robustness across individual distortion categories in a **Distortion Unaware** setting.

| Model | Input | Semantic | | | | Structural | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | CharMis | StrChar | LabMis | DecSwap | Vert | Horz | CMerge | CSplit | Foot |
| **Table-Finetuned Large Language Models** | | | | | | | | | | |
| TableLLM | Text 🔧 | 9.38 | 14.29 | 18.75 | 9.23 | 3.39 | 2.00 | 5.26 | 0.00 | 4.76 |
| TableGPT2 | Text 🔧 | 8.00 | 37.84 | 31.25 | 19.40 | 6.67 | 3.77 | 18.95 | 3.23 | 13.64 |
| **Open-source Large Language Models** | | | | | | | | | | |
| Qwen2.5VL | Text 🔧 | 15.84 | 60.53 | 31.25 | 26.09 | 11.67 | 14.29 | 18.95 | 23.33 | 31.82 |
| Qwen2.5VL | Image 🔧 | 7.92 | 55.26 | 28.12 | 17.39 | 6.67 | 16.07 | 17.20 | 45.16 | 22.73 |
| DeepseekR1 | Text | 53.47 | 78.95 | 50.00 | 66.67 | 31.15 | 78.18 | 44.21 | 54.84 | 68.18 |
| **Close-sourced Large Language Models** | | | | | | | | | | |
| GPT-5.4 | None 🔧 | 80.20 | 94.74 | 71.88 | 84.29 | 42.62 | 83.93 | 67.02 | 90.32 | 59.09 |
| GPT-5.4 | Text | 79.21 | 84.21 | 59.38 | 61.43 | 22.95 | 82.14 | 49.47 | 90.32 | 72.73 |
| GPT-5.4 | Text 🔧 | 79.21 | 94.74 | 68.75 | 85.71 | 36.07 | 91.07 | 65.26 | 90.32 | 63.64 |
| GPT-5.4 | Image | 79.21 | 78.95 | 48.39 | 52.86 | 27.87 | 75.00 | 48.42 | 83.87 | 50.00 |
| GPT-5.4 | Image 🔧 | 77.23 | 86.84 | 71.88 | 77.14 | 34.43 | 85.71 | 67.37 | 83.87 | 72.73 |
| Opus-4.6 | None 🔧 | 85.15 | 94.74 | 81.25 | 91.43 | 70.49 | 92.86 | 73.68 | 90.32 | 72.73 |
| Opus-4.6 | Text | 76.24 | 94.59 | 78.12 | 72.86 | 44.26 | 98.21 | 73.68 | 90.32 | 81.82 |
| Opus-4.6 | Text 🔧 | 91.09 | 94.74 | 81.25 | 91.43 | 57.38 | 100.00 | 74.73 | 93.55 | 81.82 |
| Opus-4.6 | Image | 70.41 | 83.78 | 65.62 | 77.14 | 42.62 | 91.07 | 55.32 | 70.97 | 68.18 |
| Opus-4.6 | Image 🔧 | 94.00 | 94.74 | 81.25 | 90.00 | 63.93 | 92.86 | 73.68 | 90.32 | 72.73 |

Table 6: Breakdown of model robustness across individual distortion categories in a **Distortion Aware** setting.