THE POWER OF ORDER: FOOLING LLMS WITH ADVERSARIAL TABLE PERMUTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) have achieved remarkable success and are increasingly deployed in critical applications involving tabular data, such as Table Question Answering (TQA). However, their robustness to the structure of this input remains a critical, unaddressed question. This paper demonstrates that modern LLMs exhibit a significant vulnerability to the layout of tabular data. Specifically, we show that semantically-invariant permutations of rows and columns—rearrangements that do not alter the table's underlying information—are sometimes sufficient to cause incorrect or inconsistent model outputs. To systematically probe this vulnerability, we introduce **Adversarial Table Permutation (ATP)**, a novel, gradient-based attack that efficiently identifies worst-case permutations designed to maximally disrupt model performance. Our extensive experiments demonstrate that ATP significantly degrades the performance of a wide range of LLMs. This reveals a pervasive vulnerability across different model sizes and architectures, including the most recent and popular models. Our findings expose a fundamental weakness in how current LLMs process structured data, underscoring the urgent need to develop permutation-robust models for reliable, real-world applications.

1 INTRODUCTION

Large language models (LLMs) have demonstrated powerful reasoning capabilities, leading to significant advancements in tasks involving structured data. A key area of progress is **table question answering (TQA)**, where models interpret and extract information from tables to answer natural-language questions (Deng et al., 2024; Liu et al., 2024). The dominant paradigm for this task involves **linearizing** the table—converting its rows and columns into a serialized text format—and including it directly in the model's prompt (Zhang et al., 2024; Jiang et al., 2023; Ye et al., 2023a). This approach effectively leverages the native text-processing power of LLMs, allowing them to achieve state-of-the-art performance on some TQA benchmarks without needing specialized architectural modifications.

Despite its practical effectiveness, this linearization strategy introduces a fundamental **semantic-structural mismatch**. Tables are inherently **permutation-invariant**; their underlying relational information remains unchanged regardless of the order of their rows or columns. In stark contrast, the transformer-based architectures of LLMs are fundamentally **order-sensitive**, processing input as a strict sequence of tokens (Shi et al., 2024; Wang et al., 2025). This discrepancy creates a critical vulnerability. Because the model's understanding is tied to a superficial textual order, two tables containing identical information but presented in different layouts can elicit inconsistent and potentially incorrect outputs. This fragility undermines the reliability of LLMs in high-stakes applications and motivates a deeper investigation into their structural robustness.

Although previous work (Yang et al., 2022; Wang & Sun, 2022) demonstrated that row and column order can influence model predictions, its analysis has key limitations that restrict its applicability to modern systems. The methodology was primarily empirical, relying on random permutations to observe output changes without providing a systematic understanding of *how* specific layouts affect model reasoning. Furthermore, this research concentrated on BERT-style models for representation learning, a paradigm fundamentally different from the now-prevalent decoder-only LLM setting, where tables are processed generatively as part of an in-context prompt. This focus on older architectures offers limited guidance on the robustness of the large-scale models used in today's applications.

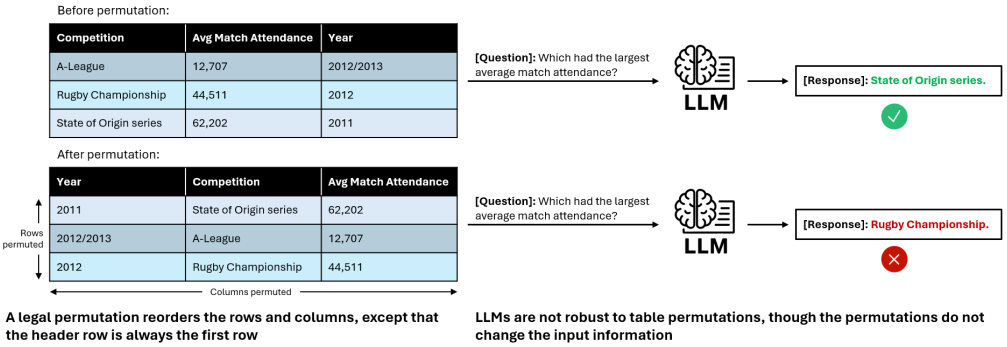


Figure 1: An illustration of the attack space for tabular inputs. A legal permutation reorders the rows and columns, except that the header row is always the first row. Such seemingly simple permutations do not change the information of the table, but are sometimes sufficient to fool modern LLMs.

In this work, we present a more rigorous investigation. We first demonstrate the susceptibility of modern LLMs to row and column permutations and then formalize this permutation sensitivity for TQA. Building on this, we introduce **Adversarial Table Permutation (ATP)**, a gradient-based attack that finds worst-case permutations by relaxing the discrete search problem into a continuous space. Across a range of instruction-tuned LLMs, ATP consistently uncovers worst-case permutations that significantly degrade prediction consistency and task performance (an example can be found in Section A.3). These adversarial layouts are transferable across model families and prompting styles and remain effective under commonly used prompting strategies. Our findings reveal a fundamental structural vulnerability in the prevailing "linearize-then-prompt" paradigm, underscoring the urgent need for more robust table reasoning techniques with LLMs.

We summarize our contributions as follows:

- We formalize the vulnerability of modern LLMs to permutations in tabular inputs, demonstrating that even random shuffling of rows and columns is sometimes sufficient to degrade model performance.
- To systematically expose this weakness, we propose the **Adversarial Table Permutation (ATP)** attack, a novel, gradient-based method that efficiently finds the worst-case permutations that cause a target model to fail. Our method is quite general and serves as a module that works for any open source LLM that takes embeddings, position ids, and attention masks as input.
- We conduct extensive experiments showing that ATP successfully degrades the performance of a wide range of LLMs, including those of various sizes and architectures. These findings reveal a critical design flaw in how current models handle structured data, underscoring the need for more robust architectures for real-world tabular applications.

2 Problem Setting

2.1 Attack Space for Tabular Input: Row and Column Permutations

Given tabular data as input, a robust model is expected to produce outputs that are **invariant** to row and column permutations that preserve the table's semantic meaning. Specifically, given a table with $n+1$ rows and m columns where the first is a header row, one can arbitrarily permute the remaining n data rows and all m columns without changing the underlying relational information. An example of such a semantically equivalent permutation can be seen in Figure 1, where the original and permuted tables contain the exact same information.

Formally, we define this attack space in the context of Table Question Answering (TQA) tasks, where we have i.i.d. samples of an input table $\mathbf{T}$, a question $\mathbf{Q}$, and an answer $\mathbf{A}$ from a given data distribution. Here, $\mathbf{Q}$ and $\mathbf{A}$ are both sequences of words, while $\mathbf{T}$ is represented as an $(n+1) \times m$ matrix where each cell contains a sequence of words. Let Π_k be the set of all $k \times k$ permutation matrices. Then the attack space for the input table $\mathbf{T}$ is defined as:

$$\boldsymbol{P}_r \mathbf{T} \boldsymbol{P}_c, \quad \text{s.t., } \boldsymbol{P}_r \in \hat{\Pi}_{n+1}, \boldsymbol{P}_c \in \Pi_m, \tag{1}$$

$$\text{where } \hat{\Pi}_{n+1} := \{\boldsymbol{P} \in \Pi_{n+1} : \boldsymbol{P}_{[0,0]=1}\}. \tag{2}$$

In Equation (1), the matrix $\boldsymbol{P}_r$ permutes the rows of $\mathbf{T}$ and $\boldsymbol{P}_c$ permutes the columns. The constraint $\boldsymbol{P}_{[0,0]} = 1$ in the definition of $\hat{\Pi}_{n+1}$ ensures that the header row is always the first row. Given this formulation, we next discuss the key research problems this work aims to address.

2.2 Are LLMs Robust Against Table Permutations?

The key motivation of this work is to investigate to what extent current LLMs are robust against row and column permutations of the input table. This can be decomposed into three key research questions: (i) Are current LLMs robust to random table permutations? (ii) How to generate the worst-case table permutation to fool a LLM? (iii) To what extent current LLMs are robust to the worst-case table permutations? We will first formalize (i) and (ii) in the rest of this section and then address (ii) in Section 3 by proposing a novel attack method, and finally answer (i) and (iii) in our experiments in Section 5.

Consider a LLM that parametrizes a probability mass function over the natural language space, as $P_{\text{model}}(\cdot)$. To employ the LLM for TQA tasks, we generate the model response by sampling from the parameterized conditional distribution, as $\hat{\mathbf{A}} \sim P_{\text{model}}(\cdot|\ \mathbf{T}, \mathbf{Q})$. Then we evaluate to what extent $\hat{\mathbf{A}}$ is semantically aligned with the ground truth $\mathbf{A}$, by some evaluation metrics $\mathcal{M}(\mathbf{A}, \hat{\mathbf{A}})$ (the bigger the better alignment).

Thus, the research problem (i) can be formulated as the calculation of the following

$$\mathbb{E}_{\hat{\mathbf{A}} \sim P_{\text{model}}(\cdot|\ \boldsymbol{P}_r \mathbf{T} \boldsymbol{P}_c, \mathbf{Q}),\ \boldsymbol{P}_r \sim \mathcal{U}_r,\ \boldsymbol{P}_c \sim \mathcal{U}_c}\ \mathcal{M}(\mathbf{A}, \hat{\mathbf{A}}), \tag{3}$$

where $\mathcal{U}_r$ and $\mathcal{U}_c$ are uniform distribution over $\hat{\Pi}_{n+1}$ and Π_m, respectively.

As for (ii), it can be formulated as finding the worst case combination of row and column permutations $(\boldsymbol{P}_r^*, \boldsymbol{P}_c^*)$ to fool a victim model P_{model}, as,

$$(\boldsymbol{P}_r^*, \boldsymbol{P}_c^*) = \underset{\boldsymbol{P}_r \in \hat{\Pi}_{n+1}, \boldsymbol{P}_c \in \Pi_m}{\arg\min}\ \mathbb{E}_{\hat{\mathbf{A}} \sim P_{\text{model}}(\cdot|\ \boldsymbol{P}_r \mathbf{T} \boldsymbol{P}_c, \mathbf{Q})}\ \mathcal{M}(\mathbf{A}, \hat{\mathbf{A}}), \tag{4}$$

and then evaluate the performance under such worst case permutations, as

$$\mathbb{E}_{\hat{\mathbf{A}}^* \sim P_{\text{model}}(\cdot|\ \boldsymbol{P}_r^* \mathbf{T} \boldsymbol{P}_c^*, \mathbf{Q})}\ \mathcal{M}(\mathbf{A}, \hat{\mathbf{A}}^*). \tag{5}$$

By Equation (3), it is straightforward to evaluate the robustness of a LLM against random table permutations. As a contrast, the optimization problem in Equation (4) is highly nontrivial. Solving the combinatorial optimization problem in Equation (4) directly in the permutation space $\hat{\Pi}_{n+1}$ and Π_m is NP-hard. The computation complexity grows exponentially with the shape of the input table. For example, when $n = m = 8$, there are around 1.6×10^9 different kinds of combinations of row and column permutations, and this number increases to 1.3×10^{11} when n and m are increased by only 1. Therefore, it is crucial to have a more effective way to find the worst case permutation, and we propose our novel method in what follows.

3 Adversarial Table Permutation (ATP) Attack

The core challenge in finding the worst-case permutation, as formulated in Equation (4), is that it requires optimizing over a vast and discrete space of permutation matrices, which is computationally intractable for tables of non-trivial size. To overcome this hurdle, our proposed Adversarial Table Permutation (ATP) attack reframes this discrete problem into a continuous one that can be solved efficiently with gradient-based methods. This is achieved through two key relaxations, which are detailed below. An illustration of the overall process can be found in Figure 2.

3.1 Relaxing the Discrete Problem into a Differentiable One

From a Non-Differentiable Metric to a Differentiable Loss. Our first step is to transform the optimization objective into a continuous, differentiable form. Specifically, the evaluation metric $\mathcal{M}(\mathbf{A}, \hat{\mathbf{A}})$ is often non-differentiable and requires sampling model outputs $\hat{\mathbf{A}}$, leading to noisy and unstable gradients. We replace this objective with the maximization of the standard cross-entropy

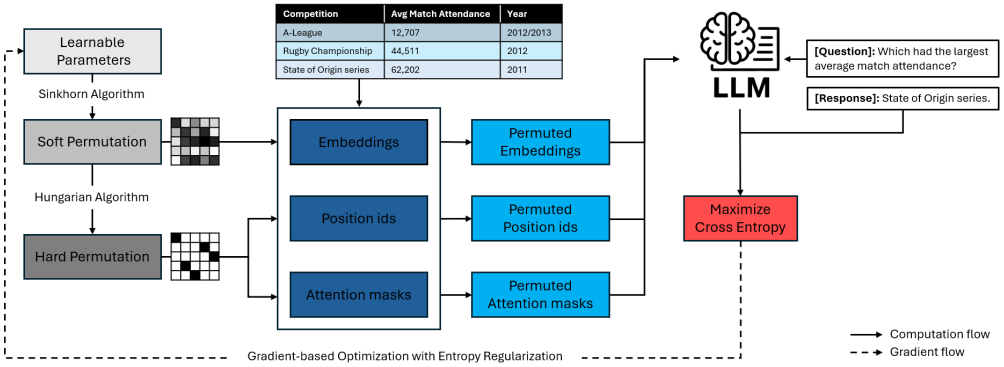


Figure 2: An illustration of the whole procedure of ATP attack, where soft permutations (doubly stochastic matrices) and hard permutations are parameterized by learnable parameters, and then used to permute embeddings of the input table and position ids and attention masks of the table, respectively. The permuted input is fed to LLM together with the question and ground truth response to calculate cross entropy loss. Finally the gradient of the loss is employed to update the learnable parameters for maximizing the cross entropy and thus fooling the victim model.

loss $\mathcal{L}_{\text{CE}}$ of the ground-truth answer $\mathbf{A}$. This provides a differentiable proxy that directly measures the alignment of model output with the ground truth answer. The optimization problem thus becomes

$$(\boldsymbol{P}_r^*, \boldsymbol{P}_c^*) = \underset{\boldsymbol{P}_r \in \hat{\Pi}_{n+1}, \boldsymbol{P}_c \in \Pi_m}{\arg\max} \mathcal{L}_{\text{CE}}(P_{\text{model}}, \boldsymbol{P}_r \mathbf{T} \boldsymbol{P}_c, \mathbf{Q}, \mathbf{A}), \tag{6}$$

where the cross-entropy loss is as follows:

$$\mathcal{L}_{\text{CE}}(\cdot) = -\sum_{t=0}^{|\mathbf{A}|-1} \log P_{\text{model}}(\mathbf{A}_{[t]} | \boldsymbol{P}_r \mathbf{T} \boldsymbol{P}_c, \mathbf{Q}, \mathbf{A}_{[:t]}). \tag{7}$$

Here $\mathcal{L}_{\text{CE}}$ is the sum of the cross-entropy for each token $\mathbf{A}_{[t]}$ conditioned on the correct context. It provides a stable, gradient-friendly objective without the need for sampling.

From Permutation Matrices to Doubly Stochastic Matrices. While the objective is now differentiable, the search space of permutation matrices ($\hat{\Pi}_{n+1}$ and Π_m) remains discrete. To create a continuous search space, we perform a convex relaxation to relax the set of permutation matrices to its convex hull. The reasons to consider convex hull lies in that the convex hull is the minimal convex and continuous superset of the original discrete space, which facilitates efficient optimization.

Theorem 1 (Birkhoff-von Neumann (Birkhoff, 1946)). *The convex hull of the set of $n \times n$ permutation matrices, Π_n, is the set of $n \times n$ doubly stochastic matrices, $\mathbb{D}_n$.*

By Theorem 1, the convex hull of Π_m is the set of all $m \times m$ doubly stochastic matrices, $\mathbb{D}_m$. As for relaxing $\hat{\Pi}_{n+1}$, we further define $\hat{\mathbb{D}}_{n+1} = \{\boldsymbol{D} \in \mathbb{D}_{n+1} : \boldsymbol{D}_{[0,0]} = 1\}$, the set of all $n \times n$ doubly stochastic matrices whose upper-left entry is always 1. As such we can now optimize over the continuous and convex sets of doubly stochastic matrices $\hat{\mathbb{D}}_{n+1}$ and $\mathbb{D}_m$.

Specifically, we first parametrize two unconstrained real matrices $\boldsymbol{\theta}_r \in \mathbb{R}^{n \times n}$ and $\boldsymbol{\theta}_c \in \mathbb{R}^{m \times m}$, and then transform they to two "soft" permutation matrices $\boldsymbol{D}_r$ and $\boldsymbol{D}_c$, by leveraging the differentiable log-Sinkhorn algorithm (Sinkhorn, 1964; Adams & Zemel, 2011; Mena et al., 2018), as

$$\boldsymbol{S}^0(\boldsymbol{\theta}) = \boldsymbol{\theta}, \ \boldsymbol{S}^{i+1}(\boldsymbol{\theta}) = \boldsymbol{N}_c(\boldsymbol{N}_r(\boldsymbol{S}^i(\boldsymbol{\theta}))), \ \boldsymbol{S}(\boldsymbol{\theta}) = \exp(\lim_{i \to \infty} \boldsymbol{S}^i(\boldsymbol{\theta})), \tag{8}$$

where $\boldsymbol{N}_r$ and $\boldsymbol{N}_c$ are row normalization and column normalization, respectively, as

$$\boldsymbol{N}_r(\boldsymbol{\theta})_{[i,:]} = \boldsymbol{\theta}_{[i,:]} - \sum_j \boldsymbol{\theta}_{[i,j]}, \quad \boldsymbol{N}_c(\boldsymbol{\theta})_{[:,j]} = \boldsymbol{\theta}_{[:,j]} - \sum_i \boldsymbol{\theta}_{[i,j]}. \tag{9}$$

By the theorem in Sinkhorn (1964), we have $\boldsymbol{S}(\boldsymbol{\theta}_r) \in \mathbb{D}_n$ and $\boldsymbol{S}(\boldsymbol{\theta}_c) \in \mathbb{D}_m$, and then we define the soft permutation matrices $\boldsymbol{D}_r$ and $\boldsymbol{D}_c$, as

$$\boldsymbol{D}_{r[1:,1:]} = \boldsymbol{S}(\boldsymbol{\theta}_r), \ \boldsymbol{D}_{r[0,0]} = 1, \ \boldsymbol{D}_{r[0,1:]} = \boldsymbol{D}_{r[1:,0]} = 0, \ \text{ and } \boldsymbol{D}_c = \boldsymbol{S}(\boldsymbol{\theta}_c), \tag{10}$$

where $\boldsymbol{D}_r$ is designed to ensure the header row remains fixed, and thus we have $\boldsymbol{D}_r \in \hat{\mathbb{D}}_{n+1}$ and $\boldsymbol{D}_c \in \mathbb{D}_m$. This allows us to search for the optimal permutation in a continuous space using gradients with respect to $\boldsymbol{\theta}_r$ and $\boldsymbol{\theta}_c$.

3.2 Projecting Back to the Permutation Space

The relaxation in Section 3.1 allows for gradient-based optimization, but it introduces a new challenge: the optimized matrices $\boldsymbol{D}_r$ and $\boldsymbol{D}_c$ are "soft" permutations, not the hard, discrete permutations required for a valid attack. Furthermore, an LLM's input for table $\mathbf{T}$ consists of multiple components: continuous token embeddings $\mathbf{T}^{\text{emb}}$ and discrete position ids $\mathbf{T}^{\text{pos}}$ and attention masks $\mathbf{T}^{\text{att}}$. While soft permutations can be applied to continuous embeddings, they cannot be applied to discrete inputs.

To address this, we apply the permutations differently based on the input type. We use the soft doubly stochastic matrices for the embeddings and project them back to the nearest permutation matrices for the discrete components. Such a projection is captured by a maximum weight matching problem that can be solved by the Hungarian algorithm (Kuhn, 1955; 1956), as

$$\text{Proj}_n(\boldsymbol{D}) = \underset{\boldsymbol{P}\in\Pi_n}{\arg\max} \langle \boldsymbol{P}, \boldsymbol{D}\rangle_F, \boldsymbol{D} \in \mathbb{D}_n, \tag{11}$$

where $\langle\cdot,\cdot\rangle_F$ is the Frobenius inner product and the subscript n in $\text{Proj}_n(\cdot)$ is dropped when the context is clear. The input table to the model is thus permuted in a hybrid mode, as

$$P_{\text{model}}(\cdot|\ \boldsymbol{D}_r\mathbf{T}^{\text{emb}}\boldsymbol{D}_c,\ \text{Proj}(\boldsymbol{D}_r)\mathbf{T}^{\text{pos}}\text{Proj}(\boldsymbol{D}_c),\ \text{Proj}(\boldsymbol{D}_r)\mathbf{T}^{\text{att}}\text{Proj}(\boldsymbol{D}_c),\ \mathbf{Q}). \tag{12}$$

This hybrid approach allows us to maintain a differentiable optimization pipeline while ensuring the final generated attack is valid and correctly manipulates all aspects of the model's input.

3.3 Regularization: Information Entropy-based Over Temperature-based

Another key challenge in our relaxed optimization is to ensure that the resulting doubly stochastic matrices, $\boldsymbol{D}_r$ and $\boldsymbol{D}_c$, are close to actual permutation matrices. Without this constraint, the soft permutations during optimization could converge to solutions far from any single permutation (e.g., a uniform matrix), creating a significant gap between the loss measured during optimization and the attack's true effectiveness. To this end, we must encourage the soft matrices to be "sharp" and structurally similar to a hard permutation.

A classic strategy is to follow Gumbel-Softmax (Jang et al., 2016; Maddison et al., 2016) to introduce a temperature parameter τ into the Sinkhorn algorithm, as $\boldsymbol{S}(\boldsymbol{\theta}/\tau)$. Lowering τ is analogous to pushing the optimization towards low-entropy solutions and thus closer to a hard permutation. However, in practice, we found that to encourage $\boldsymbol{S}(\boldsymbol{\theta}/\tau)$ to be sufficiently close to a hard permutation, it relies on a pretty small temperature τ (e.g., $\tau \leq 0.05$); such a small τ introduces significant computational instability. This issue is exacerbated when using low-precision floating-point formats like float16 or bfloat16, which are however very common for LLMs, especially in memory-constrained scenarios.

To circumvent this instability while still achieving the same goal, we incorporate the information entropy $\mathcal{H}(\cdot)$ as an explicit regularization term in our final objective, defined as follows:

$$\mathcal{H}(\boldsymbol{D}) = -\sum_{i=1}^{n}\sum_{j=1}^{n} \boldsymbol{D}_{ij}\log(\boldsymbol{D}_{ij}), \tag{13}$$

where the input $\boldsymbol{D}$ is a $n \times n$ doubly stochastic matrix. This approach, partly inspired by Dong et al. (2021a), directly encourages the soft matrices to be close to permutation matrices without the numerical issues associated with a small temperature τ. We also empirically validate this point by our ablation study in Section 5.3.

3.4 Final Objective

Thus, by adding the entropy term to the optimization objective, combining the differentiable loss and the hybrid permutation strategy, our final optimization objective is to find the parameters $(\boldsymbol{\theta}_r^*, \boldsymbol{\theta}_c^*)$

Algorithm 1 Adversarial Table Permutation (ATP) Attack

Input: P_{model}, $\mathbf{T}, \mathbf{Q}, \mathbf{A}$, N_{attack}, λ_1, λ_2;
Output: Worst case row permutation $\boldsymbol{P}_r^*$ and column permutation $\boldsymbol{P}_c^*$;
Initialize $\boldsymbol{\theta}_r \in \mathbb{R}^{n\times n}$ and $\boldsymbol{\theta}_c \in \mathbb{R}^{m\times m}$ and get $\mathbf{T}^{\text{emb}}, \mathbf{T}^{\text{pos}}, \mathbf{T}^{\text{att}}$;
for $i = 1$ to N_{attack} **do**
  Calculate $\boldsymbol{S}(\boldsymbol{\theta}_r)$ and $\boldsymbol{S}(\boldsymbol{\theta}_c)$ by Equation (8);
  Calculate $\boldsymbol{D}_r$ and $\boldsymbol{D}_c$ by Equation (10) and $\mathcal{H}(\boldsymbol{D}_r)$ and $\mathcal{H}(\boldsymbol{D}_c)$ by Equation (13) ;
  Calculate $\text{Proj}(\boldsymbol{D}_r)$ and $\text{Proj}(\boldsymbol{D}_c)$ by the Hungarian algorithm (Kuhn, 1955);
  Calculate $\mathcal{L}_{\text{CE}}(P_{\text{model}},\ \boldsymbol{D}_r\mathbf{T}^{\text{emb}}\boldsymbol{D}_c,\ \text{Proj}(\boldsymbol{D}_r)\mathbf{T}^{\text{pos}}\text{Proj}(\boldsymbol{D}_c), \text{Proj}(\boldsymbol{D}_r)\mathbf{T}^{\text{att}}\text{Proj}(\boldsymbol{D}_c), \mathbf{Q}, \boldsymbol{A}) + \lambda_1\mathcal{H}(\boldsymbol{D}_r) + \lambda_2\mathcal{H}(\boldsymbol{D}_c)$;
  Calculate the gradient of $\mathcal{L}_{\text{CE}}$ with respect to $\boldsymbol{\theta}_r$ and $\boldsymbol{\theta}_c$;
  Update $\boldsymbol{\theta}_r$ and $\boldsymbol{\theta}_c$ by Adam (Kingma, 2014);
Let $\boldsymbol{\theta}_r^*, \boldsymbol{\theta}_c^* = \boldsymbol{\theta}_r, \boldsymbol{\theta}_c$, calculate $\boldsymbol{D}_r^*, \boldsymbol{D}_c^*$ given $\boldsymbol{\theta}_r^*, \boldsymbol{\theta}_c^*$ by Equation (10);
Calculate $\boldsymbol{P}_r^* = \text{Proj}(\boldsymbol{D}_r^*)$ and $\boldsymbol{P}_c^* = \text{Proj}(\boldsymbol{D}_c^*)$
return $\boldsymbol{P}_r^*$, $\boldsymbol{P}_c^*$

that maximizes the weighted sum of cross-entropy loss and entropy regularization terms:

$$(\boldsymbol{\theta}_r^*, \boldsymbol{\theta}_c^*) = \underset{\boldsymbol{\theta}_r\in\mathbb{R}^{n\times n}, \boldsymbol{\theta}_c\in\mathbb{R}^{m\times m}}{\arg\max} \mathcal{L}_{\text{CE}}(P_{\text{model}},\ \boldsymbol{D}_r\mathbf{T}^{\text{emb}}\boldsymbol{D}_c,\ \text{Proj}(\boldsymbol{D}_r)\mathbf{T}^{\text{pos}}\text{Proj}(\boldsymbol{D}_c), \tag{14}$$

$$\text{Proj}(\boldsymbol{D}_r)\mathbf{T}^{\text{att}}\text{Proj}(\boldsymbol{D}_c), \mathbf{Q}, \boldsymbol{A}) + \lambda_1\mathcal{H}(\boldsymbol{D}_r) + \lambda_2\mathcal{H}(\boldsymbol{D}_c), \tag{15}$$

where λ_1 and λ_2 are hyper-parameters controlling the weight of the entropy terms.

Once the solution $(\boldsymbol{\theta}_r^*, \boldsymbol{\theta}_c^*)$ are found, the resulting optimal soft matrices $(\boldsymbol{D}_r^*, \boldsymbol{D}_c^*)$ are projected via $\text{Proj}(\cdot)$ to obtain the final adversarial permutation matrices $(\boldsymbol{P}_r^*, \boldsymbol{P}_c^*)$ used to attack the LLM. In our implementation the optimization in Equation (14) is solved by Adam (Kingma, 2014), where the number of iterations N_{attack} serves as a hyper-parameter (ablation study on N_{attack} in Section 5.3). We also summarize the whole algorithm procedure of the proposed ATP attack in Algorithm 1.

4 Related Work

Adversarial Attacks and Robustness for LLM. LLMs remain vulnerable to adversarial manipulations that bypass predefined safety mechanisms and induce harmful or undesired outputs (He et al., 2024; Qi et al., 2024; Hsiung et al., 2025). Token- and sentence-level perturbations can reliably elicit such outputs (Dong et al., 2021a;b; Zhao et al., 2022; Ye et al., 2022; Huang & Chang, 2021), and audits of ChatGPT have revealed substantial susceptibility (Wang et al., 2023). Beyond input-level perturbations, black-box jailbreak frameworks have been proposed to automate the discovery of exploit templates (Yu et al., 2023). Moreover, even a small number of harmful instruction–response exemplars can serve as few-shot triggers that compromise model alignment (Qi et al., 2024). Zeng et al. (2024) further applies a persuasion taxonomy to generate adversarial prompts that effectively jailbreak LLMs. These findings underscore the need for principled robustness and defense analyses.

LLM for TQA. Table-based reasoning has advanced significantly with LLMs and their emergent reasoning capabilities (Deng et al., 2024; Liu et al., 2024; Su et al., 2024; Wang et al., 2024). To utilize their capability, table contents are typically linearized into a textual sequence and included in the prompt as part of the model input. Rajkumar et al. (2022) use in-context examples for SQL generation, while Cheng et al. (2023) prompt LLMs to produce executable programs via SQL APIs. Lin et al. (2023) extracts sub-tables containing the most relevant information, and Ye et al. (2023b) enhances end-to-end reasoning by decomposing table contexts and questions using few-shot prompting, while Nguyen et al. (2025) decomposes the query into atomic steps for interpretable answers.

Robustness Evaluation for TQA. Several studies have highlighted the robustness limitations of TQA systems. For instance, Chen et al. (2023) introduces permutation-invariant table representations, while Wang & Sun (2022) extends this approach to multi-table scenarios. Bhandari et al. (2025) investigates the robustness of LLMs for TQA under domain shift, and Yang et al. (2022) demonstrates that row and column order can significantly influence model predictions. In related work, Zong et al. (2023) prompt LLMs to generate adversarial examples to improve model robustness during training. However, these investigations are largely empirical—typically involving random permutations of table structures followed by performance observation—thus failing to systematically characterize the worst-case effects of structural perturbations under in-context LLM inference.

Table 1: Alignment scores by LLM-as-judge based on Gemini 2.5 between the responses of different LLMs and the ground truth response, on WTQ dataset, under (i) Vanilla: vanilla input, (ii) Rand Perm: randomly permuted table as input, and (iii) ATP Attack: table permuted by ATP attack as input. The higher score, the better alignment between the model response with the ground truth.

LLMs	WTQ Dataset					
	Training Set			Evaluation Set		
	Vanilla	Rand Perm	ATP Attack	Vanilla	Rand Perm	ATP Attack
LLAMA-3.1-8B	0.26	0.19	**0.13**	0.26	0.17	**0.13**
LLAMA-3.1-8B-INST	0.43	0.29	**0.21**	0.47	0.31	**0.22**
TABLELLM-8B	0.29	0.20	**0.14**	0.33	0.23	**0.16**
QWEN2.5-7B-INST	0.26	0.21	**0.14**	0.29	0.21	**0.12**
QWEN2.5-14B-INST	0.44	0.33	**0.24**	0.47	0.33	**0.26**
CODELLAMA-7B-INST	0.20	0.15	**0.08**	0.18	0.16	**0.09**
DS-R1-DIST-LLAMA-8B	0.26	0.18	**0.12**	0.23	0.15	**0.09**
DS-R1-DIST-QWEN-7B	0.16	0.11	**0.07**	0.14	0.12	**0.06**

Table 2: Alignment scores of the responses of different LLMs on TATQA dataset under attack.

LLMs	TATQA Dataset					
	Training Set			Evaluation Set		
	Vanilla	Rand Perm	ATP Attack	Vanilla	Rand Perm	ATP Attack
LLAMA-3.1-8B	0.23	0.15	**0.08**	0.28	0.15	**0.11**
LLAMA-3.1-8B-INST	0.50	0.28	**0.19**	0.49	0.28	**0.20**
TABLELLM-8B	0.30	0.17	**0.11**	0.25	0.18	**0.12**
QWEN2.5-7B-INST	0.28	0.15	**0.11**	0.28	0.20	**0.13**
QWEN2.5-14B-INST	0.45	0.24	**0.17**	0.47	0.30	**0.24**
CODELLAMA-7B-INST	0.08	0.06	**0.03**	0.07	0.05	**0.03**
DS-R1-DIST-LLAMA-8B	0.28	0.17	**0.12**	0.27	0.17	**0.12**
DS-R1-DIST-QWEN-7B	0.11	0.08	**0.04**	0.11	0.08	**0.05**

5 EXPERIMENTS

In this section we focus on answering the two previous mentioned research questions: to what extent are current LLMs robust to random table permutations and the worst case table permutations, respectively? In addition, we also conduct ablation study to investigate the effectiveness of each design of the proposed ATP attack. We begin with the experimental settings.

5.1 EXPERIMENTAL SETTINGS

Datasets. We focus on TQA tasks to evaluate the robustness of LLMs. Specifically, we use three famous document-embedded TQA datasets, WikiTQ (Pasupat & Liang, 2015), and TATQA (Zhu et al., 2021), and FeTaQA (Nan et al., 2022), following Zhang et al. (2024). For each dataset, 1000 random samples from the training set and all the samples from the evaluation set are used.

Evaluation Metric. In the literature of TQA, various evaluation metrics have been considered, such as exact match, BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004), and LLM-as-judge (Zheng et al., 2023). Similar to Zhang et al. (2024), we also found that the LLM-as-judge works the best in terms of evaluating whether the model response is aligned with the reference response. Therefore, we mainly focus on using LLM-as-judge based on Gemini-2.5 (Comanici et al., 2025). To further support our choice, we conduct human evaluation and calculate the rank correlation between the score by human and the score by each metric. We found that the LLM-as-judge achieves the highest rank correlation (around 0.85) with human rating, and it surpasses the runner-up with a clear margin. More detailed discussion about the metric and the rank correlation result can be found in Section A.2.

Victim LLMs. We consider several recent and popular open-source LLMs. These are Llama family (Dubey et al., 2024) including Llama-3.1-8B and Llama-3.1-8B-Instruct, Qwen family (Bai et al., 2023) including Qwen2.5-7B-Instruct and Qwen2.5-14B-Instruct, DeepSeek family (Guo et al., 2025) including DeepSeek-R1-Distill-Llama-8B and DeepSeek-R1-Distill-Qwen-7B (due to memory limit

Table 3: Alignment scores of the responses of different LLMs on FeTaQA dataset under attack.

LLMs	FeTaQA Dataset					
	Training Set			Evaluation Set		
	Vanilla	Rand Perm	ATP Attack	Vanilla	Rand Perm	ATP Attack
LLAMA-3.1-8B	0.38	0.28	**0.21**	0.36	0.25	**0.20**
LLAMA-3.1-8B-INST	0.50	0.41	**0.30**	0.48	0.37	**0.31**
TABLELLM-8B	0.29	0.24	**0.16**	0.34	0.27	**0.20**
QWEN2.5-7B-INST	0.30	0.23	**0.17**	0.30	0.25	**0.14**
QWEN2.5-14B-INST	0.48	0.39	**0.28**	0.47	0.40	**0.29**
CODELLAMA-7B-INST	0.20	0.15	**0.09**	0.23	0.19	**0.13**
DS-R1-DIST-LLAMA-8B	0.29	0.22	**0.13**	0.28	0.21	**0.13**
DS-R1-DIST-QWEN-7B	0.14	0.11	**0.05**	0.14	0.11	**0.06**

Table 4: Ablation study on the hyper-parameters λ_1 and λ_2 by the alignment score by LLM-as-judge. Generally $\lambda_1 = \lambda_2 = 10$ performs the best.

LLMs	WTQ Dataset				
	Evaluation Set Against ATP Attack with different values of λ_1, λ_2				
	$\lambda_1, \lambda_2 = 0.0$	$\lambda_1, \lambda_2 = 0.1$	$\lambda_1, \lambda_2 = 1$	$\lambda_1, \lambda_2 = 10$	$\lambda_1, \lambda_2 = 20$
LLAMA-3.1-8B-INST	0.24	0.24	0.23	**0.22**	0.24
QWEN2.5-7B-INST	0.15	0.14	0.13	**0.12**	0.14
LLMs	**TATQA Dataset**				
	Evaluation Set Against ATP Attack with different values of λ_1, λ_2				
	$\lambda_1, \lambda_2 = 0.0$	$\lambda_1, \lambda_2 = 0.1$	$\lambda_1, \lambda_2 = 1$	$\lambda_1, \lambda_2 = 10$	$\lambda_1, \lambda_2 = 20$
LLAMA-3.1-8B-INST	0.22	0.22	**0.19**	0.20	0.22
QWEN2.5-7B-INST	0.15	0.15	0.14	**0.13**	0.15
LLMs	**FeTaQA Dataset**				
	Evaluation Set Against ATP Attack with different values of λ_1, λ_2				
	$\lambda_1, \lambda_2 = 0.0$	$\lambda_1, \lambda_2 = 0.1$	$\lambda_1, \lambda_2 = 1$	$\lambda_1, \lambda_2 = 10$	$\lambda_1, \lambda_2 = 20$
LLAMA-3.1-8B-INST	0.33	0.33	0.32	**0.31**	0.33
QWEN2.5-7B-INST	0.18	0.16	0.15	**0.14**	0.17

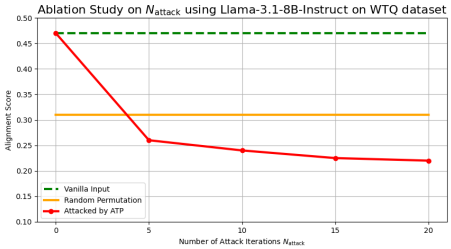
we consider distilled models for DeepSeek family), CodeLlama-7B-Instruct (Roziere et al., 2023), and TableLLM-8B that is specifically finetuned for TQA tasks Zhang et al. (2024).

5.2 MAIN RESULTS

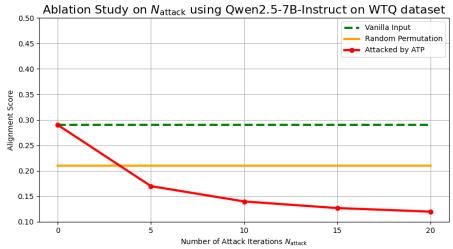
Robustness of Victim LLMs Against Random Permutations. The result for WTQ, TATQA, and FeTaQA datasets can be found in Table 1, Table 2, and Table 3 respectively. As we can see, none of the considered LLMs is robust to even random permutations. For example, on WTQ training set, the best performing LLM given vanilla input is Qwen2.5-14B-Instruct, which achieves an average of 0.44 alignment score. When attacked by random permutations, the performance of Qwen2.5-14B-Instruct drops from 0.44 to 0.33. As for TATQA training set, the best performing LLM given vanilla input is Llama-3.1-8B-Instruct, which achieves 0.50 alignment score. The performance decreases to 0.28 given random permutations. On FeTaQA training set, the best one given vanilla input is also Llama-3.1-8B-Instruct. Yet, this performance decreases from 0.50 to 0.41 given randomly permuted tables.

Robustness of Victim LLMs Against ATP Attack. Here we are interest in the robustness of modern LLMs against the worst case table permutations. As the exact worst-case solution requires solving a combinatorial optimization in Equation (4) that is NP-hard, we use our proposed ATP attack method (as in Equation (14)) to approximates the worst case scenario. The result for WTQ, TATQA, and FeTaQA datasets is shown in Table 1, Table 2, and Table 3 respectively. We found that, current LLMs are very vulnerable to these adversarially permuted input tables. For example, on WTQ evaluation set, the best performing LLM against ATP attack is Qwen2.5-14B-Instruct, which achieves an average of 0.26 alignment score under ATP attack. However, it still suffers from a significant drop of 0.21 as its performance given vanilla data is 0.47.

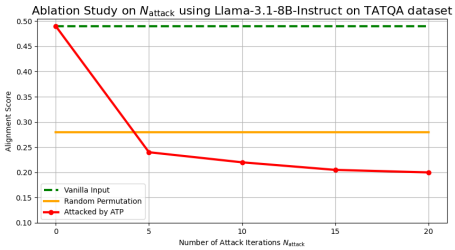
These two main empirical results reveal a fundamental defect in the current LLMs when handling tabular data: they are not robust given randomly permuted tables, and are very vulnerable against adversarially permuted input.



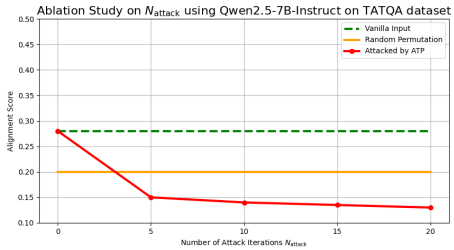
(a) Llama-3.1-8B-Inst as victim on WTQ dataset.



(b) Qwen2.5-7B-Inst as victim on WTQ dataset.



(c) Llama-3.1-8B-Inst as victim on TATQA dataset.



(d) Qwen2.5-7B-Inst as victim on TATQA dataset.

Figure 3: Influence of different values of N_{attack}, the number of attack iterations, on the power of ATP.

5.3 Ablation Study

Here we conduct ablation studies to investigate (i) the effectiveness of the entropy regularization term, and (ii) how the number of attack iterations N_{attack} influence the attack performance. The ablation study on the entropy term can be found in Table 4, where having $\lambda_1 = \lambda_2 = 10$ consistently achieves better attack performance than $\lambda_1 = \lambda_2 = 0$ (i.e., without the entropy regularization terms). For example, on FeTaQa evaluation set, without the entropy term the ATP attack can only decrease the performance of Qwen2.5-7B-Inst from 0.30 to 0.18, while with $\lambda_1 = \lambda_2 = 10$, ATP attack can further degrade the performance of Qwen2.5-7B-Inst to 0.14. At the same time, we notice that a too big λ_1, λ_2 cannot achieve the best attack power either. The reason lies in that with too big λ_1, λ_2, the optimization focuses too much on "hardening" the soft permutations, instead of fooling the victim model. As shown in the table, $\lambda_1 = \lambda_2 = 10$ generally works the best, and thus it is used for ATP attack in our main result in Section 5.2.

As for N_{attack}, the result showing the influence of N_{attack} on the power of the ATP attack is in Figure 3. We found that ATP with 5 iterations are enough to fool LLMs better than random permutations. For example, as shown in (a) Figure 3, ATP with $N_{\text{attack}} = 5$ degrades the performance of Llama-3.1-8B-Inst from 0.47 to 0.26 while random permutation can only decrease it to 0.31. At the same time, a bigger N_{attack} generally results in better attack power, which starts to converge around $N_{\text{attack}} = 20$. Thus $N_{\text{attack}} = 20$ is used in our main result in Section 5.2.

5.4 Runtime Analysis and Discussions on Attacking Closed Source LLMs

The computation complexity of ATP attack largely depends on the hyper-parameter N_{attack}. In our implementation with a single A100 GPU, it takes around 10 seconds for ATP with $N_{\text{attack}} = 20$ to attack a data point. We can also trade some attack power off by decreasing N_{attack} to 5, which can still fool modern LLMs to a considerable extent and takes only around 3 seconds per sample.

We note that our ATP attack is a gradient-based attack method and requires the access of gradients of a victim model. Thus, for those closed source LLMs such as Gemini and ChatGPT, ATP cannot be directly deployed. However, as long as we can query the model, methods such as zero-order optimization can be employed to enable ATP to work in such a black-box attack scenario. We leave this direction for future explorations.

6 Conclusion

In this work, we demonstrate the susceptibility of modern LLMs to row and column permutations and then formalize this permutation sensitivity for TQA. Building on this, we introduce Adversarial Table Permutation (ATP), attack that finds worst-case permutations to fool a victim model. We show that ATP consistently uncovers worst-case permutations that significantly degrade the performance of a various modern LLMs.

REFERENCES

Ryan Prescott Adams and Richard S Zemel. Ranking via sinkhorn propagation. *arXiv preprint arXiv:1106.1925*, 2011.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Kushal Raj Bhandari, Sixue Xing, Soham Dan, and Jianxi Gao. Exploring the robustness of language models for tabular question answering via attention analysis. *Trans. Mach. Learn. Res.*, 2025.

Garrett Birkhoff. Tres observaciones sobre el algebra lineal. *Univ. Nac. Tucuman, Ser. A*, 5:147–154, 1946.

Pei Chen, Soumajyoti Sarkar, Leonard Lausen, Balasubramaniam Srinivasan, Sheng Zha, Ruihong Huang, and George Karypis. Hytrel: Hypergraph-enhanced tabular data representation learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. Binding language models in symbolic languages. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Naihao Deng, Zhenjie Sun, Ruiqi He, Aman Sikka, Yulong Chen, Lin Ma, Yue Zhang, and Rada Mihalcea. Tables as texts or images: Evaluating the table reasoning ability of llms and mllms. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*. Association for Computational Linguistics, 2024.

Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. Towards robustness against natural language word substitutions. In *ICLR*, 2021a.

Xinshuai Dong, Anh Tuan Luu, Min Lin, Shuicheng Yan, and Hanwang Zhang. How should pre-trained language models be fine-tuned towards adversarial robustness? *Advances in Neural Information Processing Systems*, 34:4356–4369, 2021b.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Luxi He, Mengzhou Xia, and Peter Henderson. What is in your safe data? identifying benign data that breaks safety. *arXiv preprint arXiv:2404.01099*, 2024.

Lei Hsiung, Tianyu Pang, Yung-Chen Tang, Linyue Song, Tsung-Yi Ho, Pin-Yu Chen, and Yaoqing Yang. Why llm safety guardrails collapse after fine-tuning: A similarity analysis between alignment and fine-tuning datasets. *arXiv preprint arXiv:2506.05346*, 2025.

Kuan-Hao Huang and Kai-Wei Chang. Generating syntactically controlled paraphrases without using annotated parallel pairs. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pp. 1022–1033. Association for Computational Linguistics, 2021.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. Structgpt: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*. Association for Computational Linguistics, 2023.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Harold W Kuhn. Variants of the hungarian method for assignment problems. *Naval research logistics quarterly*, 3(4):253–258, 1956.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Weizhe Lin, Rexhina Blloshmi, Bill Byrne, Adria de Gispert, and Gonzalo Iglesias. An inner table retriever for robust table question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.

Tianyang Liu, Fei Wang, and Muhao Chen. Rethinking tabular data understanding with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pp. 450–482. Association for Computational Linguistics, 2024.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with gumbel-sinkhorn networks. *arXiv preprint arXiv:1802.08665*, 2018.

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

Giang Nguyen, Ivan Brugere, Shubham Sharma, Sanjay Kariyappa, Anh Totti Nguyen, and Freddy Lécué. Interpretable llm-based table question answering. *Trans. Mach. Learn. Res.*, 2025.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

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

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Nitarshan Rajkumar, Raymond Li, and Dzmitry Bahdanau. Evaluating the text-to-sql capabilities of large language models. *arXiv preprint arXiv:2204.00498*, 2022.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

Lin Shi, Chiyu Ma, Wenhua Liang, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic investigation of position bias in pairwise comparative assessments by LLMs, 2024. URL https://openreview.net/forum?id=y3jJmrKWQ4.

Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964.

Aofeng Su, Aowen Wang, Chao Ye, Chen Zhou, Ga Zhang, Gang Chen, Guangcheng Zhu, Haobo Wang, Haokai Xu, Hao Chen, et al. Tablegpt2: A large multimodal model with tabular data integration. *arXiv preprint arXiv:2411.02059*, 2024.

Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Haojun Huang, Wei Ye, Xiubo Geng, et al. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. *arXiv preprint arXiv:2302.12095*, 2023.

Zifeng Wang and Jimeng Sun. Transtab: Learning transferable tabular transformers across tables. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, et al. Chain-of-table: Evolving tables in the reasoning chain for table understanding. *arXiv preprint arXiv:2401.04398*, 2024.

Ziqi Wang, Hanlin Zhang, Xiner Li, Kuan-Hao Huang, Chi Han, Shuiwang Ji, Sham M. Kakade, Hao Peng, and Heng Ji. Eliminating position bias of language models: A mechanistic approach. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=fvkElsJOsN.

Jingfeng Yang, Aditya Gupta, Shyam Upadhyay, Luheng He, Rahul Goel, and Shachi Paul. Tableformer: Robust transformer modeling for table-text encoding. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, 2022.

Muchao Ye, Chenglin Miao, Ting Wang, and Fenglong Ma. Texthoaxer: Budgeted hard-label adversarial attacks on text. In *Thirty-Sixth AAAI Conference on Artificial Intelligence*. AAAI Press, 2022.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pp. 174–184, 2023a.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*. ACM, 2023b.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts. *arXiv preprint arXiv:2309.10253*, 2023.

Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge AI safety by humanizing llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, 2024.

Xiaokang Zhang, Sijia Luo, Bohan Zhang, Zeyao Ma, Jing Zhang, Li Yang, Guanlin Li, Zijun Yao, Kangli Xu, Jinchang Zhou, Daniel Zhang-Li, et al. Tablellm: Enabling tabular data manipulation by llms in real office usage scenarios. *arXiv preprint arXiv:2403.19318*, 2024.

Haiteng Zhao, Chang Ma, Xinshuai Dong, Anh Tuan Luu, Zhi-Hong Deng, and Hanwang Zhang. Certified robustness against natural language attacks by causal intervention. In *International Conference on Machine Learning*, pp. 26958–26970. PMLR, 2022.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. *arXiv preprint arXiv:2105.07624*, 2021.

Yongshuo Zong, Tingyang Yu, Ruchika Chavhan, Bingchen Zhao, and Timothy Hospedales. Fool your (vision and) language model with embarrassingly simple permutations. *arXiv preprint arXiv:2310.01651*, 2023.

A APPENDIX

A.1 USAGE OF LLM

We employed LLMs as an auxiliary tool to polish the grammar and clarity of our manuscript. This usage was limited to improving readability and presentation; all conceptual contributions, analyses, and interpretations were developed by the authors.

A.2 METRIC USED FOR ALIGNMENT SCORES

We further illustrate the effect of the ATP attack by contrasting model outputs generated from the original table input with those obtained from its adversarially permuted counterpart, thereby validating the effectiveness of our evaluation metrics. For each instance, we generate outputs under both conditions and compute example-level alignment scores with respect to the reference answer. We report three metrics to assess the similarity between generated and reference outputs:

- **ROUGE-L (Lin, 2004)**: lexical similarity via longest common subsequence between generated and reference answers.
- **LLM-as-judge** (Zheng et al., 2023): a held-out LLM (Gemini 2.5 (Comanici et al., 2025)) scores the semantic similarity between the model's output and the ground truth.
- **Human annotation**: human raters assess semantic similarity relative to the ground truth answer.

Method. We uniformly sample TQA instances in the dataset. For each instance, we construct two prompts (original vs. ATP-permuted table), prompt the model to generate outputs, and score them with ROUGE-L, the LLM-as-judge, and human raters. To assess agreement among metrics, we compute Spearman's rank correlation over example-level scores for each metric pair. The resulting correlations are summarized in Figure 4.

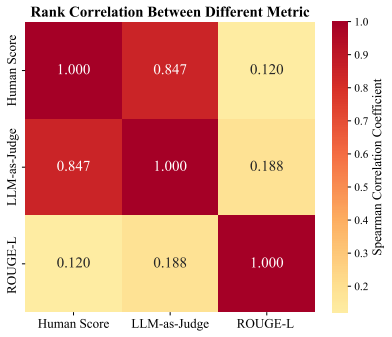


Figure 4: Spearman's Rank Correlation between different metrics. The LLM-as-judge method showed strong alignment with the human rater.

Based on these results, we observe that the LLM-as-judge metric exhibits strong alignment with human judgments, whereas ROUGE-L fails to capture such alignment. This finding supports the effectiveness of our LLM-as-judge as an evaluation metric for measuring alignment score between LLM-generated responses and ground-truth answers. In contrast, metrics like ROUGE-L, which rely on lexical overlap, are limited to token-level similarity and thus fall short in evaluating semantic equivalence, especially in tasks involving long-form reasoning in TQA task. We further illustrate this limitation with an example in which the adversarially perturbed (ATP-attacked) table input flips the model's originally correct prediction(see Section A.3). Although the perturbed output achieves a higher ROUGE-L score due to surface-level overlap, it no longer preserves the correct semantics. Conversely, the LLM-as-judge metric successfully detects this change in prediction, validating its suitability for robust evaluation in such scenarios.

We also provide an illustrative example below to show how ATP permutes the input table to fool a victim LLM. In the example we can also see that the LLM-as-judge works better than Rouge-L. More specifically, given original input, the model gives a correct response of "Desmond Trufant" while with permuted input, the model outputs an incorrect answer of "Robert Alford", Note that the ground truth is "Desmond Trufant". Rouge-L gives a score of 0.37 for the correct response but gives 0.39 to the incorrect response, which is not ideal. As a contrast, LLM-as-judge gives 0.8 to the correct one and gives 0.1 to the incorrect response, which aligns well with human rating.

A.3 An Illustrative Example for ATP Attacked Input

Example of ATP Attack Result (by Qwen2.5-14B-Instruct)

Instruction. Offer a thorough and accurate solution that directly addresses the Question outlined in the [Question].

[Table]

```
Round|Selection|Player|Position|College|
1|22|Desmond Trufant|CB|Washington|
2|60|Robert Alford|CB|Southeastern Louisiana|
4|127|Malliciah Goodman|DE|Clemson|
4|133|Levine Toilolo|TE|Stanford|
5|153|Stansly Maponga|DE|TCU|
7|243|Kemal Ishmael|S|Central Florida|
7|244|Zeke Motta|S|Notre Dame|
7|249|Sean Renfree|QB|Duke|
```

[Table Description]
This table summarizes an NFL draft class drawn from college programs. For each pick, it lists the round, overall selection number, player name, position, and college.
[Question]
Who is the top listed player?

Original Answer (Rouge-L: 0.37, LLM-as-judge Score: 0.8)
[Solution]Among the athletes listed in the draft class table, the first corresponds to **Desmond Trufant**, who was selected in the opening round of the draft, holding the 22nd overall position.

Attacked Instruction. Offer a thorough and accurate solution that directly addresses the Question outlined in the [Question].

[Table] (Permuted by ATP Attack)

```
College|Round|Player|Selection|Position|
Clemson|4|Malliciah Goodman|127|DE|
Stanford|4|Levine Toilolo|133|TE|
Duke|7|Sean Renfree|249|QB|
Central Florida|7|Kemal Ishmael|243|S|
Washington|1|Desmond Trufant|22|CB|
Notre Dame|7|Zeke Motta|244|S|
TCU|5|Stansly Maponga|153|DE|
Southeastern Louisiana|2|Robert Alford|60|CB|
```

[Table Description]
This table summarizes an NFL draft class drawn from college programs. For each pick, it lists the round, overall selection number, player name, position, and college.
[Question]
Who is the top listed player?

Attacked Answer (Rouge-L: 0.39, LLM-as-judge Score: 0.1)
[Solution]The top-listed player in your table is **Robert Alford** from Southeastern Louisiana, where he played as a Cornerback. He was selected the 2nd pick 60th in Round 2.

[Ground Truth]
The top listed player is **Desmond Trufant**, who was selected in Round 1 with the 22nd pick. He played as a cornerback and attended the University of Washington.