How Do Language Models Understand Tables? A Mechanistic Analysis of Cell Location

Xuanliang Zhang[1], Dingzirui Wang[1], Keyan Xu[1], Qingfu Zhu[1], Wanxiang Che[1]*
[1]Harbin Institute of Technology
{xuanliangzhang, dzrwang, kyxu, qfzhu, car}@ir.hit.edu.cn

Abstract

While Large Language Models (LLMs) are increasingly deployed for table-related tasks, the internal mechanisms enabling them to process linearized two-dimensional structured tables remain opaque. In this work, we investigate the process of table understanding by dissecting the atomic task of cell location. Through activation patching and complementary interpretability techniques, we delineate the table understanding mechanism into a sequential three-stage pipeline: Semantic Binding, Coordinate Localization, and Information Propagation. We demonstrate that models locate the target cell via an ordinal mechanism that counts discrete delimiters to resolve coordinates, offering mechanistic evidence across diverse table formats. Furthermore, column indices are encoded within a linear subspace that allows for precise steering of model focus through vector arithmetic. Finally, we extend our analysis to the real-world HiTab dataset and show that the same three-stage mechanism persists in hierarchical real-world tables and more complex tasks.

1 Introduction

Tables serve as a ubiquitous format for organizing two-dimensional structured data across real-world applications [1]. Given the exceptional capabilities of Large Language Models (LLMs) in natural language comprehension and reasoning, researchers have increasingly applied these models to table-related tasks [4, 49]. However, Transformer models necessitates the linearization of two-dimensional structures into one-dimensional sequences, which differs from linearized natural language and could limit their understanding. While numerous methods have been proposed to enhance performance through prompt optimization, reasoning decomposition, or specialized training [50, 42, 20], the internal mechanisms governing how models understand tabular data remain largely opaque. This lack of transparency regarding the underlying dynamics significantly impedes the development of mechanistic interpretability and reasoning reliability.

Prior works in mechanistic interpretability primarily address tasks such as factual recall or indirect object identification [32, 43, 13, 6]. However, **no existing study investigates the internal mechanics of table understanding**. This gap is critical because table understanding differs fundamentally from natural language understanding: it demands that the model adhere to rigid spatial coordinates within a linearized input. Deciphering the mechanisms is therefore a prerequisite for understanding how LLMs bridge the gap between sequential processing and two-dimensional structural logic.

We address this gap by discovering the atomic task of *cell location*. We select this task as our focal point because it demands the synergistic execution of both semantic alignment (matching headers) and structural reasoning (navigating coordinates), thereby serving as a minimal yet comprehensive proxy for broader table understanding capabilities. We analyze the internal states of state-of-the-art open-weights models including the Qwen and Llama series to discover the information flow from input headers to output cells under controlled mechanistic intervention and probing settings.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

To unravel the internal mechanics of this process, we employ activation patching analysis, cross-validated via linear probing and attention ablation. We delineate the table understanding mechanism into a sequential three-stage pipeline: **Semantic Binding**, **Coordinate Localization**, and **Information Propagation**. The process commences with Semantic Binding in early layers, where the model aligns query constraints with table headers. Then in Coordinate Localization, the model abstracts away from semantic content to navigate structural indices. The final stage involves Information Propagation in late layers, where the targeted value is propagated to the output distribution.

To further explore the nature of the Coordinate Localization phase, we examine the internal representations using linear probes and noise intervention experiments. We demonstrate that the model locates the target cell via an ordinal, distance-invariant mechanism that **counts discrete delimiters** to resolve coordinate states before propagating them to cell content. Furthermore, to investigate the geometric properties of this representation, we inject directional shift vectors into the residual stream. We find that scaling the unit vector successfully steers the model to focus on specific relative column positions, and that vectors representing different offsets exhibit additive compositionality, confirming that column indices are encoded within **a linear subspace navigated via vector arithmetic**.

To assess whether these mechanisms generalize beyond controlled synthetic tables, we extend our analysis to HiTab [3], a real-world hierarchical table reasoning dataset that requires models to identify supporting cells and perform more complex reasoning operations such as aggregation. We find that the same three-stage pipeline persists: row alignment heads remain causally necessary for Semantic Binding, coordinate-encoding heads exhibit a diagonal geometric interaction pattern over HiTab matrix indices, and late-layer information mover heads propagate the located answer cells for decoding. These results suggest that LLMs rely on a shared structural mechanism for table understanding across both synthetic and real-world settings.

We summarize our primary contributions as follows.

- We utilize activation patching to formalize the table understanding process as three stages: Semantic Binding, Coordinate Localization, and Information Propagation.
- We explore the Coordinate Localization stage using linear probes and steering vectors. We discover that the models maintain an implicit coordinate system by counting delimiters and represent column indices in a linear subspace.
- We verify the generalization of these mechanisms on the real-world HiTab benchmark, showing that the same three-stage pipeline persists in hierarchical tables.

2 Preliminaries

2.1 Task Formulation: Cell Location

Cell Location Task. We study a fundamental operation in table understanding: locating a cell value from its row and column headers. In the synthetic setting,, a table T consists of column headers $\mathcal{C} = \{c_1, \ldots, c_C\}$ and row headers $\mathcal{R} = \{r_1, \ldots, r_R\}$, where row headers are unique identifiers in the primary key column. The cell value at the intersection of row header r_i and column header c_j is denoted as v_{r_i,c_j}. We serialize T into x_{table} using Markdown format with pipe delimiters ('|') and newlines.

Input Construction. The model receives a prompt $x = [D, x_{table}, x_{query}]$, where D is a one-shot demonstration. The query x_{query} is a natural language question that explicitly specifies a target row header constraint $r \in \mathcal{R}$ and a column header constraint $c \in \mathcal{C}$, and the model is expected to generate $y = v_{r,c}$. To isolate structural reasoning from parametric memorization, we use synthetic tables whose cell values are randomly sampled from an entity pool, forcing the model to rely on the provided context rather than memorized facts [2, 36]. We construct 500 samples for atomic cell location (Appendix D); variability across independently generated datasets is reported in Appendix D.3.

Design Rationale and Generalization Analysis. We start from this controlled synthetic setting for two reasons. First, cell location is an *atomic yet representative operation*: it requires the model to not only perform semantic alignment but also to execute structural reasoning by navigating the coordinates [26, 34]. Second, the setting yields *high model accuracy*, which enables more reliable causal interventions and reduces confounds from ambiguous queries, noisy content, or task-specific

shortcuts [24, 43, 21]. We then test whether the resulting mechanistic findings generalize to the zero-shot setting, larger tables, more complex tasks, and real-world data in §6 and Appendix G.

2.2 Models

To ensure the generality of our findings, we conduct our analysis on 4 LLM families, ranging from 0.6B to 32B. Specifically, we utilize the **Qwen** series (including Qwen3-0.6B, Qwen3-4B, Qwen2.5-32B) [40], **Llama** series (including Llama-3.2-3B, Llama-3.1-8B) [8], **Mistral** series (Mistral-7B-v0.1) [14], and **Gemma** series (Gemma-3-4b-it) [30]. These models employ Rotary Positional Embeddings (RoPE) [28] and represent state-of-the-art open-weights architectures. We primarily report the results of Qwen3-4B (thinking mode) as it achieves a high accuracy of 90% on our benchmark, with error analysis in Appendix F. Our core findings consistently generalize across evaluated models and non-thinking mode, with detailed results and analysis in Appendix G.3 and H.

2.3 Analytic Techniques

To unravel the mechanism of table understanding, we employ a suite of mechanistic interpretability techniques. We mainly introduce the three analysis tools that recur throughout our analysis.

Activation Patching We employ Activation Patching to localize task-relevant information flow within the model. Let $\mathbf{h}_t^{(l)} \in \mathbb{R}^d$ denote the pre-attention residual stream at layer l and token position t. To estimate the causal contribution of a component, we replace its activation in a corrupted run, given x_{corrupt}, with the corresponding activation from a clean run, given x_{clean} and target y.

Following previous works [33, 43, 17], we consider two patching granularities:

- **Layer-level Patching:** We patch the residual stream $\mathbf{h}_t^{(l)}$ over a token span $t \in \mathcal{T}$ to identify layers and positions that mediate the location process.
- **Head-level Patching:** We patch the output of an attention head $H^{(l,a)}$, where $a \in \{1, \ldots, A\}$, at the final input token or target-cell positions to measure head-level causal contributions.

We quantify the importance of each patched component using the *Effect Score*, based on Logit Difference (LD) [12]. We define $\text{LD}(x) = \text{logit}(x)_y - \text{logit}(x)_{y'}$ as the confidence margin between the target and the foil, and compute

$$\text{Effect} = \frac{\text{LD}_{\text{patch}} - \text{LD}_{\text{corrupt}}}{\text{LD}_{\text{clean}} - \text{LD}_{\text{corrupt}}}, \tag{1}$$

where LD_{patch} is the logit difference after intervention. Effect measures relative recovery of the clean–corrupt gap. Denominator and raw-recovery controls appear in Appendix E.2.

Linear Probing To test whether geometric coordinates are linearly encoded in the residual stream, we train a linear regressor P on activations $\mathbf{h}_t^{(l)}$ to predict the token's ground-truth row or column index $z \in \{r_{idx}, c_{idx}\}$:

$$\hat{z} = P(\mathbf{h}_t^{(l)}) = \mathbf{W}_p \mathbf{h}_t^{(l)} + \mathbf{b}_p, \tag{2}$$

where $\mathbf{W}_p \in \mathbb{R}^{1 \times d}$ and $\mathbf{b}_p \in \mathbb{R}$. We evaluate probe performance using R^2. A higher R^2 indicates stronger linear encoding of table coordinates. Training and evaluation details are provided in Appendix E.4.

Ablation Studies To verify the causal necessity of specific attention heads, we employ ablation techniques. The total output of the multi-head attention (MHA) block at layer l and position t is the aggregation of independent head outputs: $\text{MHA}(\mathbf{h}_t^{(l)}) = \sum_{a=1}^{A} H_t^{(l,a)}$. We define a target set of heads $\mathcal{H}_{abl}$ to intervene upon. **Zero Ablation** explicitly removes the information flow from the target heads by forcing their output activations to zero [7]: $\hat{H}_t^{(l,a)} = \mathbf{0}, \quad \forall (l,a) \in \mathcal{H}_{abl}$. By measuring the change of outputs, we observe whether the targeted heads are necessary components for cell location.

3 The Three-Stage Information Flow of Table Understanding

3.1 Overview

Experimental Setup To map the information flow of table understanding, we employ activation patching on semantically segmented input sequences. Let $\mathcal{I}_q(r)$ and $\mathcal{I}_q(c)$ denote the query token spans for row and column constraints. Within the fixed table x_{table}, we define $\mathcal{I}_t(c)$ and $\mathcal{I}_t(r)$ as the token spans for the column header c and row headers r, and $\mathcal{I}_t(v_{r,c})$ as the span for the cell value located by r and c. We construct corrupted inputs x' by altering the query constraints to a counterfactual column or row header, thereby shifting the target value location. We define two settings:

- **Column Patching:** We replace the query column constraint c with a counterfactual c'. This shifts the target value indices from the correct span $\mathcal{I}_t(v_{r,c})$ to the counterfactual span $\mathcal{I}_t(v_{r,c'})$.
- **Row Patching:** We replace the query row constraint r with r'. This shifts the target indices to $\mathcal{I}_t(v_{r',c})$.

Figure 1 presents the Column Patching results; detailed configurations, coordinate definitions, and the symmetric Row Patching results are provided in Appendix E.2.

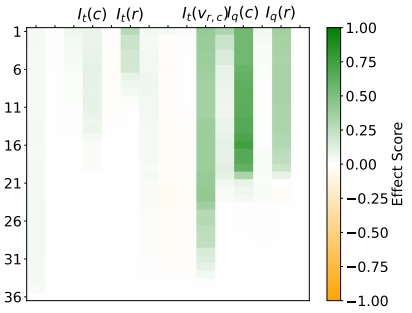


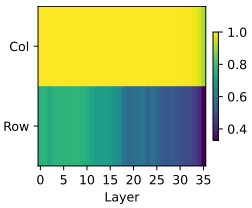
Figure 1: Average column patching effect score across layers and token positions by corrupting the queried column. The heatmap shows a three-stage causal flow: early-layer dependence on query constraints (I_q), mid-layer dependence on the target cell ($I_t(v_{r,c})$), and near-zero late-layer sensitivity.

Analysis of Causal Dynamics he Effect Score distribution in Figure 1 reveals a layer-wise shift in causal mediation. Together with the complementary analyses below, they support three functional stages:

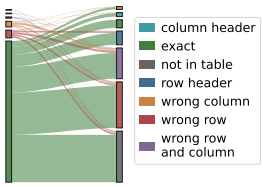
1. **Stage I: Semantic Binding (Early Layers 1-16).** Significant causal mass is concentrated on query constraints ($\mathcal{I}_q(r), \mathcal{I}_q(c)$), indicating that the model focuses on the constraints in the question and mainly binds the natural-language constraints to the corresponding headers.
2. **Stage II: Coordinate Localization (Middle Layers 17-23).** The causal importance shift decisively from the headers to the target cell region ($\mathcal{I}_t(v_{r,c})$), and patching the query constraints no longer matters. This indicates that the model has already identified the queried row and column and now needs to locate their intersection in the serialized table.
3. **Stage III: Information Propagation (Late Layers 24+).** Interventions on all token positions yield negligible causal effects in this stage. This diminished sensitivity indicates that the answer $v_{r,c}$ has been encoded within the internal states. Consequently, the primary mechanism shifts from location to the propagation to the final token position for decoding.

3.2 Stage I: Semantic Binding

To investigate the semantic binding of query constraints to table headers, we first analyze the geometric alignment of their internal representations. At each layer, we mean-pool post-layer residual states over the full token span of each query constraint and candidate header, and measure the top-1 accuracy, defined as the top-1 header retrieval by cosine similarity between the pooled representations. As illustrated in Figure 2a, the accuracy for both dimensions converges close to 1.0 in early layers, indicating that semantic mapping occurs well. Notably, column headers exhibit tighter alignment than rows, suggesting easier binding, whereas row constraint resolution is strictly confined to the initial layers.



(a) Representation alignment across layers.



(b) Error transitions after row-head ablation.

Figure 2: Semantic Binding analysis. **(a)** We measure top-1 header retrieval by cosine similarity between query constraints and candidate table headers; both row and column constraints align in early layers. **(b)** Zero-ablating the top-20 *row alignment heads* collapses exact matches and increases wrong-row/header errors, providing behavioral evidence for their necessity.

We further examine the causal role of semantic attention through attention ablation. We identify specialized alignment heads that preferentially

attend from query constraints to target table headers, with details in Appendix E.3. In Figure 2b, only the `exact` flow denotes correct predictions, while all other flows correspond to error types. Zero-ablating the top-20 *row alignment heads* (95% in layers 1-16) reduces accuracy from 94.4% to 5.2%. By contrast, ablating the top-20 column alignment heads or random heads causes only smaller drops of 6.6% and 13.8%. The `wrong row` rate also rises to 29.4% under row-head ablation, compared with only 11.0% for random ablation, confirming that these heads bind row-header semantics. Thus, row grounding requires explicit semantic attention, whereas column localization appears to rely on signals beyond semantic alignment.

3.3 Stage II: Coordinate Localization

Following semantic alignment, the model transitions to locating the geometric coordinates of the target cell. Specifically, when the row header is identified, the model should locate the target cell according to the column header.

We use a Virtual-RoPE coordinate diagnostic on column-averaged Q/K states. Let $\bar{q}_j$ and $\bar{k}_l$ denote query and key vectors averaged by target and table column index, respectively. We compute $S_{j,l} = \mathrm{RoPE}(\bar{q}_j, j)^\top \mathrm{RoPE}(\bar{k}_l, l)$, using logical column indices rather than the actual sequence positions of native attention. Thus, $S_{j,l}$ is a counterfactual geometric diagnostic, not a realized logit. We select the top 20 heads that maximize the contrast between diagonal ($j = l$) and off-diagonal ($j \neq l$) scores, and term them *coordinate-encoding heads*. Notably, 90% of these heads are located in layers 17–23.

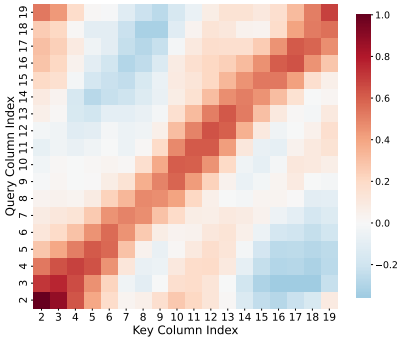


Figure 3: Virtual-RoPE coordinate diagnostic for *coordinate-encoding heads*. Logical column indices are used as virtual rotary positions. The diagonal pattern shows that target column indices are aligned with table column positions.

Figure 3 shows a clear diagonal pattern: $S_{j,l}$ is maximized when the target column index matches the table column index. Using the maximal column in each row as the predicted position yields a Pearson correlation of 0.997 with the ideal diagonal. This geometric alignment indicates that middle-layer heads localize target cells by column coordinates, effectively treating the serialized table as a coordinate system. Held-out baselines and head-count sensitivity appear in Appendices E.4 and E.4.

3.4 Stage III: Information Propagation

We analyze the final stage in which the model transfers the localized table value to the final token for generation. We perform activation patching at the final input token while keeping the table context x_{table} fixed and modifying the query x_{query} to target a different row header r' and column header c'. For each attention head, we compute the Effect Score to measure its causal contribution to predicting the original target y over the counterfactual target y'.

Figure 4 shows the head-level Effect Scores across layers at the final input token. Positive-effect heads become densely distributed from Layer 24 onward. By inspecting their attention patterns, we find that the three highlighted heads attend to the target cell token $v_{r,c}$ in over 70% of samples, substantially more often than to other context tokens. This identifies them as *information mover heads*, which propagate the value at the resolved coordinates to the final-token residual stream for decoding.

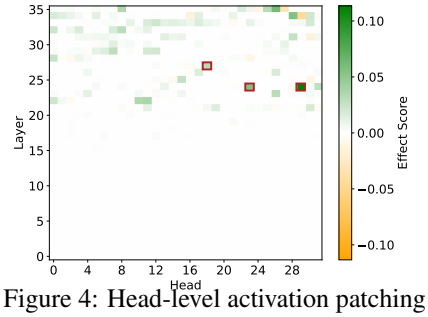
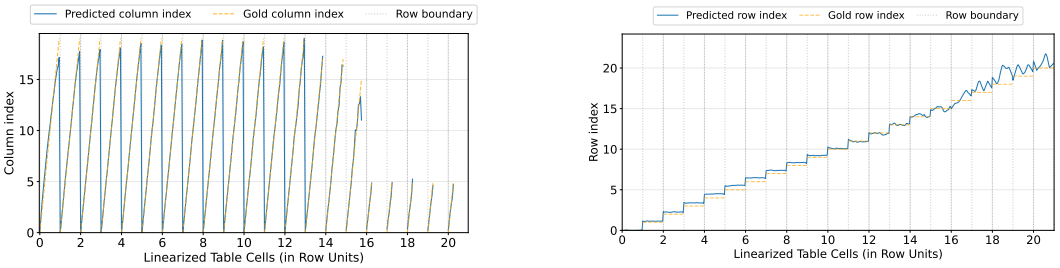


Figure 4: Head-level activation patching at the final input token. Late-layer heads with positive Effect Scores, highlighted by red boxes, attend strongly to the target cell and act as *information mover heads* for decoding.



(a) Column index probe.

(b) Row index probe.

Figure 6: Layer 17 probe predictions (blue) versus gold column/row index (orange) across linearized table cells. The column probe forms a sawtooth pattern that resets at row boundaries, and the row probe forms a stepwise function that increments across newlines, supporting a delimiter-driven ordinal coordinate system for tracking table cells during location.

4 Implicit Coordinate System via Delimiter Counting

Having established that the model performs coordinate localization in the middle layers, we now investigate the precise mechanism by which these coordinates are constructed.

4.1 Representations of Column and Row Indices

To verify the existence of internal coordinate representations, we train linear probes on residual stream activations at each layer to predict the column index c_{idx} and row index r_{idx} of each cell token in the table sequence x_{table}. Figure 5 reports the average R^2 across tokens. Both probes stabilize between layers 17 and 23, with peak R^2 values of 0.954 for columns and 0.843 for rows at layer 17. The stronger column probe suggests that the model relies primarily on column indices for navigation, while row positions are treated as secondary features accumulated from row boundaries.

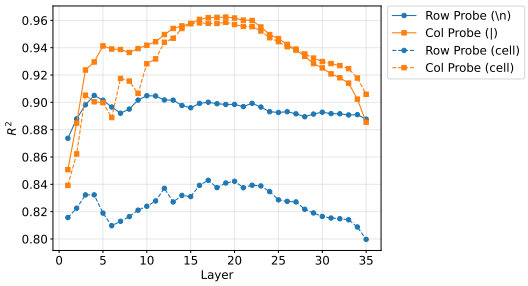


Figure 5: Layer-wise R^2 of linear probes for row and column indices on cell and delimiter tokens. Delimiter probes reach higher peak R^2 earlier than cell probes, showing earlier coordinate decodability at structural delimiters.

Figure 6 shows the predicted column and row indices. The column index follows a clear **sawtooth pattern**: it increases within each row and resets at row boundaries, suggesting a delimiter-triggered column counter. In contrast, the row index forms a **stepwise function**, remaining stable within a row and incrementing after each newline. Prediction errors grow with larger row and column counts, indicating degraded table understanding as table dimensions increase; see Appendix G.2.

4.2 Role of Delimiters

The sawtooth pattern observed above implies that the model relies on specific tokens to govern its counting mechanism. We posit that the pipe delimiter ('|') serves as the fundamental trigger for this incremental counting process, marking transitions between adjacent table cells.

Probe Comparison To disentangle whether coordinate information is intrinsic to cell content or anchored to structural delimiters, we analyze the layer-wise evolution of probe performance. While both delimiter and cell-based probes eventually reach comparable peak R^2 values in deeper layers, their trajectory differs significantly (Figure 5). Probes trained on delimiter tokens exhibit robust linear separability ($R^2 > 0.95$) immediately in the early layers. In contrast, cell content probes show a marked latency, stabilizing at high accuracy only after Layer 15. This performance gap in the initial 23 layers suggests a hierarchical information flow: the model first resolves coordinate states at the

delimiter positions and subsequently propagates this positional information to the constituent cell tokens. Thus, the delimiter acts not merely as a separator, but as the primary synchronization signal for the model's internal coordinate system. We also present the results of using other tabular formats with different delimiters in Appendix G.4.

Activation Patching To test whether coordinate states originate at delimiters and then propagate to cells, we perform a targeted causal intervention. We select the top-20 attention heads by the R^2 of row- and column-index probes at either cell or delimiter positions. We construct counterfactual inputs by swapping rows or columns, which changes the target cell coordinates while keeping the query fixed, and patch the selected head activations from the original sequence to the swapped sequence at the target cell position. Table 1 shows a clear causal hierarchy consistent with the probing results. Heads selected from cell positions have negligible Effect Scores (0.006 for rows and 0.025 for columns), suggesting that cell tokens receive coordinate information rather than drive the counting process. By contrast, delimiter-selected heads exert strong causal control. These interventions support a causal role for delimiter-associated coordinate information in localization, consistent with delimiter-driven updates and propagation to cell content.

Table 1: Causal contribution of coordinate-encoding heads selected from cell- versus delimiter-token probes.

Head Function	**Probing Position**	
	Cell	**Delimiter**
Row Index Heads	0.006	0.527
Column Index Heads	0.025	0.958

We also show the experiments of noise intervention in Appendix E.5. Synthesizing findings from above, we conclude that the model locates cells by simulating a discrete ordinal coordinate system. This system is strictly ordinal: the model resolves column indices through the sequential accumulation of delimiter synchronization signals. Similarly, row indices are updated via discrete state transitions at row boundaries. Then the model propagates the tabular indices to the constituent cell tokens.

5 The Linear Geometry of Tabular Indices

In this section, we investigate the geometric properties of column index representation. We hypothesize that the model represents column indices within a structured linear subspace with approximate translation consistency and additivity in the tested regime We show the detailed experiment setup in Appendix E.6.

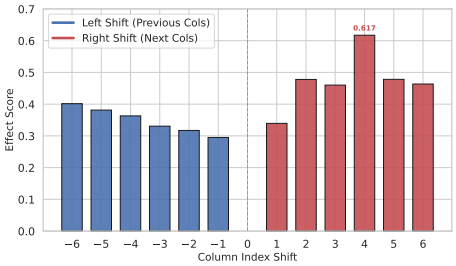
5.1 Steering with Unit Shift Vectors

To characterize the geometric regularity and **translation invariance** of the column representation, we derive a global unit shift vector $\vec{v}_{unit}$ by averaging the mean activation differences between adjacent column header tokens in x_{table} across layers 17-23. We then perform steering interventions by modifying the target activation $\mathbf{h}$ as $\tilde{\mathbf{h}} = \mathbf{h} + k \cdot \alpha \cdot \vec{v}_{unit}$ where $k \in \mathbb{Z}$ is an integer representing the desired column shift and α is a steering coefficient.

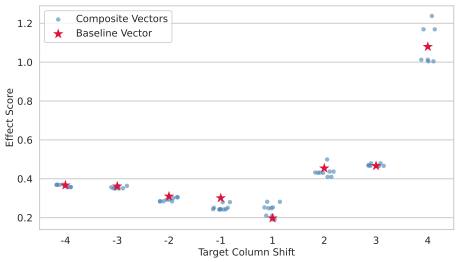
Figure 7a illustrates the Effect Score across different integer shifts k, which provides strong evidence for the linearity of the column representation. We observe that multiplying the unit shift vector by k consistently steers the model to focus on the column located k steps away. This suggests that the model locates the cell by aligning its column position with that of the target column header, rather than relying solely on header semantics, and encodes column positions as approximately equidistant states over the tested offsets along a direction. The ability to traverse multiple columns by scaling a single vector confirms that the internal coordinate system possesses a robust linear structure. Steering is local and serialization-dependent: it weakens at large offsets and boundaries, while HTML benefits from format-specific extraction (Appendices E.7 and E.7).

5.2 Compositionality of Column Vectors

We further investigate the **additive compositionality** of the representation space. If the subspace is truly linear, the vector required to shift attention by k columns should be decomposable: the sum of two vectors shifting by a and b columns should be functionally equivalent to a single shift of $a + b = k$ columns. We define the composite vector $\vec{v}_{comp} = \vec{v}_a + \vec{v}_b$ and compare its steering effect



(a) Unit-vector steering across column offsets.



(b) Compositionality of column shift vectors.

Figure 7: Interventions in the column-index subspace. **(a)** Injecting $k\vec{v}_{unit}$ steers the model toward columns shifted by k, supporting approximate translation consistency over the tested offsets. **(b)** Composite vectors $\vec{v}_a + \vec{v}_b$ match directly extracted baseline vectors when $a + b = k$, showing additive linear geometry.

against the baseline vector $\vec{v}_k$, which is directly extracted from columns separated by distance k. We conduct experiments for various target offsets k using multiple combinations of constituent shifts a and b drawn from the range $[-6, 6]$. We evaluate the interventions using the Effect Score.

Figure 7b compares baseline shift vectors with composite vectors, with details in Appendix E.6. The close alignment between the red baseline stars and blue composite dots indicates strong linearity. For example, the $k = 4$ baseline vector achieves an Effect Score of 1.079, and the composite shift $1 + 3$ achieves 1.169. Similarly, the $k = -4$ baseline vectors score of 0.367 is closely matched by $-2 + (-2)$ with 0.357. These results support approximate additivity over the tested offsets. We further validate this interpretation through robustness analyses over instance-level directions, table serializations, prompt templates, and alternative shift-vector constructions in Appendix E.7.

6 Generalization to Real-World Hierarchical Tables

6.1 Experimental Setup

We now examine whether the same three-stage mechanism persists in realistic table understanding scenarios. To this end, we conduct experiments on HiTab [3], a hierarchical table reasoning dataset involving nested headers, multi-level table structures, and complex operations. We retain the original synthetic multi-cell retrieval experiments in Appendix G.5, where the same heads are reused to multiplex atomic location mechanisms by employing parallel semantic binding and multi-threaded coordinate localization across multiple row or column constraints.

We first evaluate Qwen3-4B on HiTab by prompting the model to output both the final answer and the evidence cells supporting its prediction. We categorize each prediction according to answer correctness, evidence faithfulness, and localization behavior. Appendix E.8 summarizes the distribution of prediction types. To ensure that we analyze the mechanism of locating the correct cells, we *exclude the corresponding error cases at each stage*, following previous works [16, 24].

6.2 Stage I: Semantic Binding in Hierarchical Tables

We first examine whether the semantic binding mechanism identified in Stage I generalizes to HiTab. On the row-correct subset, we identify *row alignment heads* using the same criterion as in §3.2. We zero-ablate the identified heads and measure the transition of prediction types. As shown in Figure 8, ablating these heads does not merely perturb the final answer; it causes 57.4% of outputs to collapse into empty answers. The strong failure mode on HiTab suggests that hierarchical row grounding is brittle: once the model loses access to the aligned row semantics, it frequently becomes unable to produce any supported

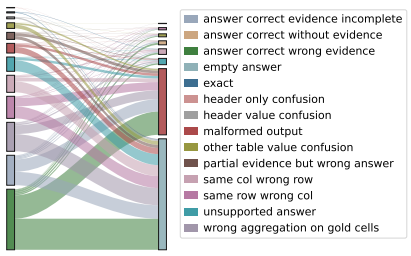


Figure 8: Error-type transitions after zero-ablating HiTab *row alignment heads*.

answer , confirming that Semantic Binding is still the first stage of table understanding in realistic hierarchical tables.

6.3 Stage II: Coordinate Localization over HiTab Matrix Indices

We next investigate whether the model still performs structural coordinate localization after semantic binding. Since HiTab contains hierarchical and merged headers, we map each table cell to its matrix column index and apply the same Virtual-RoPE coordinate diagnostic as in §3.3. Figure 9 shows the average geometric interaction scores of the top-20 *coordinate-encoding heads*, exhibiting a clear diagonal structure: the interaction score is maximized when the query column index matches the key column index. We take the column with the maximum score in each row as the predicted position and compare it with the ideal diagonal position. The resulting Pearson correlation coefficient is 0.988, demonstrating that the coordinate-encoding heads discovered in controlled tables have a direct analogue in HiTab. This result suggests that the model navigates the serialized table using an internal coordinate system under realistic nested table structures.

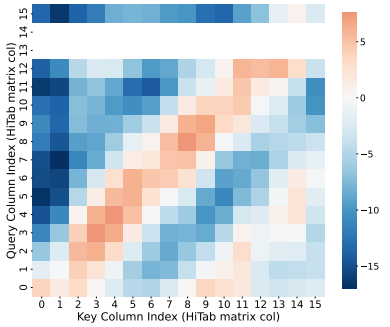


Figure 9: Geometric interaction scores of HiTab *coordinate-encoding heads*. Rows denote query-side target column indices and columns denote table matrix column indices.

6.4 Stage III: Information Propagation from Located Cells

Finally, we analyze whether the located cell value is propagated to the final token position by information mover heads. We focus on examples where the answer is a table cell and the model correctly localizes the cell, thereby ensuring that the analyzed behavior reflects successful cell location. We compute the target max-attention win rate on each head, which measures the fraction of samples in which an attention head assigns its largest attention, from the final input token, to a target-cell token rather than to any other token. As shown in Figure 10, we again observe a few late-layer heads that consistently focus on the target answer token. The top-10 highlighted heads have high target max-attention win rates on these cell-answer examples, consistent with the behavior of the *information mover heads* identified in §3.4. This provides evidence that, once the model locates the correct cell in a realistic table, it uses the same late-stage propagation mechanism to transfer the cell content to the final residual stream for decoding. The HiTab analysis demonstrates that the three-stage pipeline discovered in atomic cell location is not an artifact of synthetic tables. Semantic Binding, Coordinate Localization, and Information Propagation remain observable in hierarchical tables with nested headers and more complex reasoning requirements. This supports the view that LLMs reuse a general structural mechanism for table understanding across both controlled and realistic settings, despite substantial variation in table schema complexity. Hierarchy-stratified and failure analyses show that row binding can remain detectable despite column mislocalization, with hierarchy-to-coordinate mapping emerging as a major bottleneck (Appendices E.8 and F).

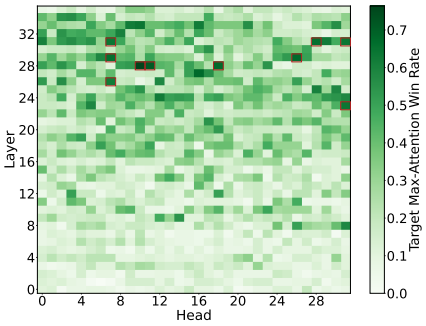


Figure 10: Target max-attention win rate of attention heads on HiTab cell-answer examples. Red boxes highlight the top-10 heads, acting as *information mover heads* for decoding.

7 Conclusion

In this work, we provide a mechanistic dissection of table understanding in LLMs via the cell location task. Through activation patching, linear probing, and ablation study, we formalize this process into three stages: Semantic Binding, Coordinate Localization, and Information Propagation. We reveal that models navigate structures by maintaining an implicit coordinate system driven by delimiter

counting, with locally steerable column representations. Finally, we show that the same mechanism generalizes to HiTab, suggesting that LLMs rely on a shared structural pipeline for both synthetic and real-world table understanding. Our findings bridge the gap between sequential processing and structural logic, offering a robust framework for improving the reliability and interpretability of LLMs in structured data applications.

Acknowledge

We gratefully acknowledge the support of the National Natural Science Foundation of China (NSFC) via grant 62236004 and 62476073.

References

[1] Gilbert Badaro, Mohammed Saeed, and Paolo Papotti. Transformers for tabular data representation: A survey of models and applications. *Transactions of the Association for Computational Linguistics*, 11:227–249, 2023.

[2] Yuxing Cheng, Yi Chang, and Yuan Wu. A survey on data contamination for large language models, 2025.

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

[4] Xi Fang, Weijie Xu, Fiona Anting Tan, Ziqing Hu, Jiani Zhang, Yanjun Qi, Srinivasan H. Sengamedu, and Christos Faloutsos. Large language models (LLMs) on tabular data: Prediction, generation, and understanding - a survey. *Transactions on Machine Learning Research*, 2024.

[5] Jiahai Feng and Jacob Steinhardt. How do language models bind entities in context? In *The Twelfth International Conference on Learning Representations*, 2024.

[6] Zeyu Gan, Ruifeng Ren, Wei Yao, Xiaolin Hu, Gengze Xu, Chen Qian, Huayi Tang, Zixuan Gong, Xinhao Yao, Pengwei Tang, Zhenxing Dou, and Yong Liu. Beyond the black box: Theory and mechanism of large language models, 2026.

[7] Rhys Gould, Euan Ong, George Ogden, and Arthur Conmy. Successor heads: Recurring, interpretable attention heads in the wild. In *The Twelfth International Conference on Learning Representations*, 2024.

[8] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and et al. The llama 3 herd of models, 2024.

[9] Abir Harrasse, Philip Quirke, Clement Neo, Dhruv Nathawani, Luke Marks, and Amir Abdullah. TinySQL: A progressive text-to-SQL dataset for mechanistic interpretability research. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29256–29284, Suzhou, China, November 2025. Association for Computational Linguistics.

[10] Xinyi He, Yihao Liu, Mengyu Zhou, Yeye He, Haoyu Dong, Shi Han, Zejian Yuan, and Dongmei Zhang. TableLoRA: Low-rank adaptation on table structure understanding for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22376–22391, Vienna, Austria, July 2025. Association for Computational Linguistics.

[11] Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. Linearity of relation decoding in transformer language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[12] Guan Zhe Hong, Nishanth Dikkala, Enming Luo, Cyrus Rashtchian, Xin Wang, and Rina Panigrahy. A implies b: Circuit analysis in LLMs for propositional logical reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[13] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[14] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[15] Rihui Jin, Yu Li, Guilin Qi, Nan Hu, Yuan-Fang Li, Jiaoyan Chen, Jianan Wang, Yongrui Chen, Dehai Min, and Sheng Bi. Hegta: Leveraging heterogeneous graph-enhanced large language models for few-shot complex table understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23):24294–24302, Apr. 2025.

[16] Michael Lan, Philip Torr, and Fazl Barez. Towards interpretable sequence continuation: Analyzing shared circuits in large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12576–12601, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[17] Belinda Z. Li, Zifan Carl Guo, and Jacob Andreas. (how) do language models track state? In *Forty-second International Conference on Machine Learning*, 2025.

[18] Jia-Nan Li, Jian Guan, Wei Wu, Zhengtao Yu, and Rui Yan. 2d-TPE: Two-dimensional positional encoding enhances table understanding for large language models. In *THE WEB CONFERENCE 2025*, 2025.

[19] Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*, 2023.

[20] Liyao Li, Chao Ye, Wentao Ye, Yifei Sun, Zhe Jiang, Haobo Wang, Jiaming Tian, Yiming Zhang, NINGTAO WANG, Xing Fu, Gang Chen, and Junbo Zhao. Table as a modality for large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[21] Tianhe Lin, Jian Xie, Siyu Yuan, and Deqing Yang. Implicit reasoning in transformers is reasoning through shortcuts. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9470–9487, Vienna, Austria, July 2025. Association for Computational Linguistics.

[22] Tianyang Liu, Fei Wang, and Muhao Chen. Rethinking tabular data understanding with large language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 450–482, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[23] Zhenghao Liu, Haolan Wang, Xinze Li, Qiushi Xiong, Xiaocui Yang, Yu Gu, Yukun Yan, Qi Shi, Fangfang Li, Ge Yu, and Maosong Sun. Hippo: Enhancing the table understanding capability of large language models through hybrid-modal preference optimization, 2025.

[24] Aaron Mueller, Atticus Geiger, Sarah Wiegreffe, Dana Arad, Iván Arcuschin, Adam Belfki, Yik Siu Chan, Jaden Fried Fiotto-Kaufman, Tal Haklay, Michael Hanna, Jing Huang, Rohan Gupta, Yaniv Nikankin, Hadas Orgad, Nikhil Prakash, Anja Reusch, Aruna Sankaranarayanan, Shun Shao, Alessandro Stolfo, Martin Tutek, Amir Zur, David Bau, and Yonatan Belinkov. MIB: A mechanistic interpretability benchmark. In *Forty-second International Conference on Machine Learning*, 2025.

[25] Yonatan Oren, Nicole Meister, Niladri S. Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. Proving test set contamination in black-box language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[26] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Proc. of ACL*, 2015.

[27] Ananya Singha, José Cambronero, Sumit Gulwani, Vu Le, and Chris Parnin. Tabular representation, noisy operators, and impacts on table structure understanding tasks in LLMs. In *NeurIPS 2023 Second Table Representation Learning Workshop*, 2023.

[28] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[29] Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, page 645–654, New York, NY, USA, 2024. Association for Computing Machinery.

[30] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025.

[31] Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023.

[32] Boshi Wang, Xiang Yue, Yu Su, and Huan Sun. Grokking of implicit reasoning in transformers: A mechanistic journey to the edge of generalization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[33] Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023.

[34] Lanrui Wang, Mingyu Zheng, Hongyin Tang, Zheng Lin, Yanan Cao, Jingang Wang, Xunliang Cai, and Weiping Wang. NeedleinATable: Exploring long-context capability of large language models towards long-structured tables. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[35] Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister. Chain-of-table: Evolving tables in the reasoning chain for table understanding. In *The Twelfth International Conference on Learning Representations*, 2024.

[36] Mingqi Wu, Zhihao Zhang, Qiaole Dong, Zhiheng Xi, Jun Zhao, Senjie Jin, Xiaoran Fan, Yuhao Zhou, Huijie Lv, Ming Zhang, Yanwei Fu, Qin Liu, Songyang Zhang, and Qi Zhang. Reasoning or memorization? unreliable results of reinforcement learning due to data contamination, 2025.

[37] Xianjie Wu, Jian Yang, Linzheng Chai, Ge Zhang, Jiaheng Liu, Xeron Du, Di Liang, Daixin Shu, Xianfu Cheng, Tianzhen Sun, Tongliang Li, Zhoujun Li, and Guanglin Niu. Tablebench: a comprehensive and complex benchmark for table question answering. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025.

[38] Zhenhe Wu, Jian Yang, Jiaheng Liu, Xianjie Wu, Changzai Pan, Jie Zhang, Yu Zhao, Shuangyong Song, Yongxiang Li, and Zhoujun Li. Table-r1: Region-based reinforcement learning for table understanding, 2025.

[39] Junjie Xing, Yeye He, Mengyu Zhou, Haoyu Dong, Shi Han, Lingjiao Chen, Dongmei Zhang, Surajit Chaudhuri, and H. V. Jagadish. MMTU: A massive multi-task table understanding and reasoning benchmark. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

[40] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and et al. Qwen3 technical report, 2025.

[41] Yukang Yang, Declan Iain Campbell, Kaixuan Huang, Mengdi Wang, Jonathan D. Cohen, and Taylor Whittington Webb. Emergent symbolic mechanisms support abstract reasoning in large language models. In *Forty-second International Conference on Machine Learning*, 2025.

[42] Zhen Yang, Ziwei Du, Minghan Zhang, Wei Du, Jie Chen, Zhen Duan, and Shu Zhao. Triples as the key: Structuring makes decomposition and verification easier in LLM-based tableQA. In *The Thirteenth International Conference on Learning Representations*, 2025.

[43] Jiaran Ye, Zijun Yao, Zhidian Huang, Liangming Pan, Jinxin Liu, Yushi Bai, Amy Xin, Liu Weichuan, Xiaoyin Che, Lei Hou, and Juanzi Li. How do transformers learn implicit reasoning? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[44] 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 '23, page 174–184, New York, NY, USA, 2023. Association for Computing Machinery.

[45] Elias Zavitsanos, Dimitris Mavroeidis, Eirini Spyropoulou, Manos Fergadiotis, and Georgios Paliouras. Entrant: A large financial dataset for table understanding. *Scientific Data*, 11(1):876, 2024.

[46] Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*, 2024.

[47] Hugh Zhang and David Parkes. Chain-of-thought reasoning is a policy improvement operator. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023.

[48] Xuanliang Zhang, Dingzirui Wang, Longxu Dou, Baoxin Wang, Dayong Wu, Qingfu Zhu, and Wanxiang Che. Flextaf: Enhancing table reasoning with flexible tabular formats, 2024.

[49] Xuanliang Zhang, Dingzirui Wang, Longxu Dou, Qingfu Zhu, and Wanxiang Che. A survey of table reasoning with large language models. *Frontiers of Computer Science*, 19(9):199348, 2025.

[50] Xuanliang Zhang, Dingzirui Wang, Keyan Xu, Qingfu Zhu, and Wanxiang Che. RoT: Enhancing table reasoning with iterative row-wise traversals. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 559–579, Suzhou, China, November 2025. Association for Computational Linguistics.

A Limitations

Our analysis is limited to a selected set of table-related datasets and language models, and thus the identified mechanisms may not fully cover all table reasoning scenarios. Future work should evaluate the proposed interpretation across more diverse datasets, model families, and model scales to test its generality. In addition, our experiments focus on English tables and questions; extending the analysis to other languages and multilingual table settings would further clarify whether the observed semantic binding, coordinate localization, and information propagation mechanisms are language-independent.

Residual-stream patching identifies where task-relevant information is causally mediated, not every computation that produces it. Our functional interpretation therefore relies on its convergence with probing, head ablation, attention analysis, and steering rather than on patching alone. The Virtual-RoPE score is a counterfactual geometric diagnostic, not a reconstruction of native attention logits. Steering establishes locally usable coordinate directions in the tested regimes: effects weaken at large offsets and boundaries, and intervention directions depend on serialization. HiTab stage-conditioned analyses reveal executed computations in eligible subsets but do not establish robust merged-cell or hierarchical reasoning; hierarchy-stratified and failure diagnostics document these limits. The pre-output diagnostic is moderate and variable, rather than a deployment-ready detector. Our experiments also cover a selected set of models, datasets, and English-language tables and questions.

B Social Impact

This paper presents the work whose goal is to advance the field of machine learning through a mechanistic interpretability analysis of table understanding in Large Language Models. By identifying the specific attention heads and internal representations responsible for cell location, we contribute to the broader effort of making AI systems more transparent and predictable. Such insights are foundational for building trustworthy AI that handles structured information across various domains. Our code is available at github.

C Related Works

C.1 Mechanistic Interpretability of Reasoning

Recent advancements in mechanistic interpretability have extensively mapped the internal circuits responsible for logical and linguistic reasoning [6]. Researchers reverse-engineer the algorithmic components of specific tasks such as indirect object identification where models employ specialized attention heads to move information between token positions [33, 46]. Subsequent studies extend this analysis to propositional logic demonstrating that models utilize a modular pipeline of locating moving and processing heads to resolve implications [12]. Furthermore investigations into implicit multi-hop reasoning reveal that Transformers can learn to separate memorization from generalization often exhibiting distinct developmental stages [32, 43]. Also, research on Chain-of-Thought processes demonstrates that models utilize intermediate tokens as a computational scratchpad to decompose global logic into iterative state updates across layers [31, 47]. Most recently, analysis reveals that Text-to-SQL generation operates via a strictly staged mechanism: the model first determines the abstract syntactic structure and subsequently fills in specific schema entities [9].

However, prior works primarily investigate tasks governed by explicit logical rules or generative syntax. In contrast our work investigates the mechanism of table understanding where the model must resolve spatial constraints rather than purely semantic dependencies. We show that table understanding necessitates a distinct Coordinate Localization stage, where the model navigates the serialized grid structure via delimiter counting instead of semantic signals.

C.2 Internal Representations of Structure

A parallel line of research explores how LLMs encode geometric or structural knowledge within their internal representations. The existing work on synthetic tasks such as Othello proves that sequence models can spontaneously emerge internal world representations allowing them to track complex game states without explicit supervision [19]. Regarding sequential structure researchers identify successor heads that utilize linear subspaces to represent ordered sequences confirming that models

Table 2: Statistics of the synthetic datasets for Table Understanding tasks. The table details the range of table dimensions and the scope of the query targets (number of rows or columns requested) for each experimental setting.

Task Type	**Table Rows (R)**	**Table Cols (C)**	**Query Rows**	**Query Cols**
Atomic Cell Location	$[4, 20]$	$[4, 20]$	1	1
Multi-Row Location	$[4, 20]$	$[4, 20]$	$\lvert\mathcal{R}_{sub}\rvert \in [2, 11]$	1
Multi-Column Location	$[4, 20]$	$[4, 20]$	1	$\lvert\mathcal{C}_{sub}\rvert \in [2, 11]$

possess linear arithmetic capabilities for progression [7]. Additional studies on entity binding propose that models assign abstract binding identifiers to link entities with their attributes independent of their absolute positions [5, 41]. Furthermore, recent findings demonstrate that for many factual and linguistic relations, transformer language models implement a linear decoding strategy where an affine transformation on the intermediate subject representation directly maps to the corresponding object [11].

While these works establish that models can internalize static world models or sequential progressions, our research uncovers a dynamic mechanism for understanding tables. Unlike fixed game boards or sequential entities, tables require an internal coordinate system. We distinguish our contribution by proving that this structural navigation is governed by an implicit linear geometry constructed via delimiter counting, which allows the model to generalize coordinate localization to multi-target queries through vector arithmetic in the residual stream.

C.3 LLMs for Table Understanding

Since the integration of LLMs into table-related tasks, a burgeoning body of research has emerged to evaluate and enhance the tabular understanding capabilities of these models [4]. Regarding evaluation, datasets such as MMTU [39] and ENTRANT [45] introduce multi-task and domain-specific datasets in the financial sector, revealing that current models still exhibit significant deficiencies in comprehensive table comprehension. Furthermore, existing studies indicate that LLMs lack robustness in this area, as modifications to the organizational order of contents or structural formats frequently change performance [27, 29, 22]. Efforts to improve tabular understanding are generally categorized into two paradigms. The first encompasses training-free methods which optimize prompts by decomposing complex questions and tables [44], refining data serialization formats [42, 48], or integrating tools to facilitate reasoning [35]. The second paradigm focuses on enhancing intrinsic capabilities through training, which involves designing specialized model architectures [10, 15] or conducting supervised fine-tuning and reinforcement learning on large-scale tabular corpora [23, 38].

Despite these advancements in evaluating and enhancing performance, the underlying mechanisms governing how LLMs understand tables remain a black box. Our work fills this gap by shifting the focus from external performance metrics to internal mechanism analysis. We focus on the atomic *cell location* task and provide the first formal characterization of its internal stages: Semantic Binding, Coordinate Localization, and Information Propagation. By uncovering that models navigate linearized tables via delimiter-counting mechanisms and encode coordinates within a linear subspace, we offer a fundamental shift in perspective—from observing that models understand tables to explaining *how* they bridge the gap between sequential input and two-dimensional logic.

D Data Details

To rigorously investigate the internal mechanics of table understanding, we synthesize datasets specifically designed for the cell location task. We select cell location as the primary focus because it constitutes an atomic operation requiring the model to not only perform semantic alignment—matching query terms to row and column headers—but also to execute structural reasoning by navigating the two-dimensional coordinates of the table. Also, it is a common and fundamental subproblem of more complex table-related reasoning problems in the wild, such as the Table Question-Answering task [26, 37].

We deliberately avoid utilizing existing datasets to preclude the risk of data contamination [2, 25]. By constructing synthetic tables with randomized values, we ensure that the model cannot rely on

Example D.1: An input sample for the Single-Cell Location task.

Question:
In the table, what is the value of category for Nike?

Table:
brand | category | price
Samsung | electronics | high
Nike | clothing | medium
Ikea | furniture | low

Answer:
clothing

Question:
In the table, what is the value of element for director?

Table:
role | genre | element | fabric | gemstone
coordinator | horror | Earth | Fleece | Opal
scientist | documentary | Light | Denim | Quartz
technician | history | Void | Nylon | Sapphire
writer | sport | Thunder | Suede | Pearl
consultant | thriller | Metal | Lace | Onyx
planner | animation | Steam | Cotton | Emerald
director | epic | Dust | Velvet | Topaz
marketer | biopic | Wood | Silk | Agate
trainer | adventure | Shadow | Cashmere | Jade
leader | noir | Air | Satin | Ruby
operator | drama | Fire | Rayon | Coral
researcher | action | Ice | Leather | Amethyst

Answer:

parametric knowledge or memorization to answer queries [36]. Instead, the model is forced to extract answers solely via in-context reasoning over the provided x_{table}. We describe the data generation process below, with dataset statistics summarized in Table 2.

D.1 Atomic Cell Location

Table Generation We first construct a comprehensive entity pool comprising 83 distinct semantic categories. To ensure value uniqueness, the sets of values across different categories are disjoint. For the table structure, the primary key (the header of the first column c_1) is restricted to generic identifiers including *name*, *role*, and *id*. The remaining column headers $\mathcal{C} \setminus \{c_1\}$ are randomly sampled from the entity pool categories. The dimensions of the tables are randomized, with the number of rows R and columns C uniformly sampled from the interval $[4, 20]$. This upper bound 20 is selected to ensure that the linearized table sequences remain within the model's context window while maintaining a challenging complexity profile, following the principles of the previous work [12]. Critically, this scale requires the model to perform robust two-dimensional coordinate mapping within a linear sequence without inducing performance collapse, thereby facilitating a granular investigation into the internal mechanisms of cell location. We generate a total of 500 independent tables.

Query Formulation For each synthesized table, we configure a corresponding query x_{query}. We randomly sample a target row header $r \in \mathcal{R}$ and a target column header $c \in \mathcal{C}$. The natural language question is generated using one of several templates, such as *What is the c for r?* or *According to*

Table 3: Sampling variability across ten independently generated Qwen3-4B datasets. SD denotes the sample standard deviation across *datasets*, not the standard error or a confidence interval. Percentage-based SDs are in percentage points (pp).

Metric	Mean	Sample SD
Clean EM	90.4%	5.48 pp
Row-ablation EM	3.8%	2.74 pp
Row-head accuracy drop	86.6 pp	4.43 pp
Coordinate top-1	78.2%	5.20 pp
Target/predicted-coordinate correlation	0.684	0.105
Markdown steering Effect Score	0.261	0.032

the table, what is the value of c for r?. To clearly define the task and the expected output format, a one-shot demonstration D is prepended to every input. Example D.1 illustrates a representative input sample.

D.2 Multi-Cell Location

To probe the model's ability to generalize to more complex structural constraints, we extend the benchmark while maintaining the same base table corpus (500 samples) used in the atomic cell location task. The queries are categorized into two distinct types:

- **Multi-Row Location:** For a fixed column header c, we randomly select a subset of row headers $\mathcal{R}_{sub} \subset \mathcal{R}$ to form the query, requiring the model to locate multiple values in a vertical structure.
- **Multi-Column Location:** For a fixed row header r, we randomly select a subset of column headers $\mathcal{C}_{sub} \subset \mathcal{C}$, requiring the location of values across a horizontal structure.

Consistent with the atomic-cell setting, specific one-shot demonstrations are provided for each task type to guide the generation of list-formatted answers.

D.3 Variability across Independently Generated Datasets

Deterministic decoding does not eliminate variability from random table and query generation. We evaluate Qwen3-4B on ten independently generated synthetic datasets and report the mean and sample standard deviation across datasets in Table 3.

Every dataset preserves the large row-head ablation effect, above-chance localization, and positive steering, although the point estimates vary. These results quantify sampling variability rather than variation from repeatedly decoding identical inputs.

E Experiment Details and Supplement

E.1 Experiments compute resources

Our main experiments are adapted on a single A100-80B, which costs one hour for each experiment.

E.2 Details of Activation Patching

In this subsection, we provide the specific implementation details of the activation patching experiments and further analyze the results of Row Patching to corroborate the findings presented in §3.1.

Experimental Setup To achieve high-resolution localization of information flow, we employ a fine-grained tokenization strategy rather than averaging over broad token types. The input sequence is partitioned into 15 distinct regions. The underlying segmentation principle is to isolate the semantic focus tokens, specifically the query constraints, their corresponding table headers, and the target cell

values, while aggregating the surrounding non-focal text into context segments. This approach allows us to distinguish the causal role of structural coordinates from general context.

The coordinates on the x-axis (Indices 1-15) in Figure 1 correspond to the following token spans:

- **Index 1:** The demonstration and initial table content preceding the relevant column headers.
- **Index 2:** The corrupt column header $\mathcal{I}_t(h')$ in the table schema definition.
- **Index 3:** The tokens between $\mathcal{I}_t(c')$ and $\mathcal{I}_t(c)$.
- **Index 4:** The relevant column header in the table schema, corresponding to the target column $\mathcal{I}_t(c)$.
- **Index 5:** The intermediate table content, including data rows positioned between the header definition and the target row.
- **Index 6:** The row header of the target row $\mathcal{I}_t(r)$, which serves as the primary vertical anchor.
- **Index 7:** The delimiter token separating the row header from cell values.
- **Index 8:** The corrupt cell value $\mathcal{I}_t(v_{r,c'})$ within the target row.
- **Index 9:** The tokens between $\mathcal{I}_t(v_{r,c'})$ and $\mathcal{I}_t(v_{r,c})$.
- **Index 10:** The target cell value $\mathcal{I}_t(v_{r,c})$ situated at the intersection of the specified row and column.
- **Index 11:** The remaining table content following the target row and the preamble query.
- **Index 12:** The column constraint token within the natural language query $\mathcal{I}_q(c)$.
- **Index 13:** The syntactic bridge tokens (e.g., prepositions) within the query between $\mathcal{I}_q(c)$ and $\mathcal{I}_q(r)$.
- **Index 14:** The row constraint token within the natural language query $\mathcal{I}_q(r)$.
- **Index 15:** The query suffix and answer prefix tokens.

We calculate the Effect Score at each layer and token span index. An Effect Score close or beyond 1.00 indicates that the patch fully recovers the correct logit difference.

Row Patching Analysis In the main text, we discuss the Three-Stage mechanism primarily through the lens of Column Patching. Here, we demonstrate that Row Patching (corrupting the query row constraint to shift the target to a different row) exhibits an identical mechanistic pattern, as shown in Figure 11.

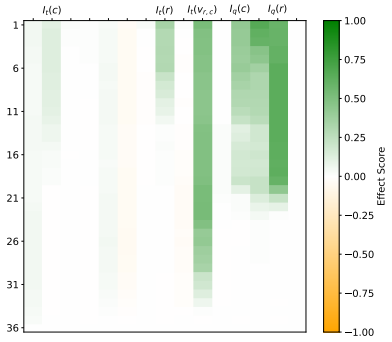


Figure 11: Average Row Patching Effect Score across layers and token positions. The heatmap visualizes the causal importance of specific token regions when the row constraint in the query is corrupted.

1. **Stage 1: Semantic Binding (Layers 1-16).** Consistent with column patching, the early layers exhibit high Effect Scores concentrated on the query constraints and their corresponding schema matches. Specifically, we observe strong activation on the Query Row $\mathcal{I}_q(r)$ (Index 14) and the corresponding Table Row Header $\mathcal{I}_t(r)$ (Index 6). This confirms that the model initiates the reasoning process by binding the semantic constraint from the question to the specific row identifier in the table structure.
2. **Stage 2: Coordinate Localization (Layers 17-23).** In the middle layers, the causal mass shifts decisively rightward from the row header to the cell values. Notably, Index 10, which corresponds to the target value $\mathcal{I}_t(v_{r,c})$, displays a distinct band of high importance. This validates the Coordinate Localization stage, where the model, having anchored the correct row r and column c, converges on the specific intersection coordinate.

Table 4: Clean–corrupt denominator statistics for the Qwen3-4B control.

Statistic	Value
Number of examples	500
Mean denominator	44.79
Median absolute denominator	45.13
Fifth percentile	32.52
Examples with $\lvert D\rvert < 0.5$	1 (0.2%)

3. **Stage 3: Information Propagation (Layers 24+).** In the final layers, the heatmap fades to neutral values across all input tokens. This with final-token head patching indicates that the extraction is complete; the target information $v_{r,c}$ has been successfully encoded into the residual stream of the final token, and the generation process no longer relies on direct attention to the input sequence.

These results reinforce our conclusion that table understanding in LLMs operates via a structured, sequential process of constraint matching followed by intersection navigation, independent of whether the location requires row-centric or column-centric reasoning.

Robustness of the Effect Score For each example, let

$$D = \mathrm{LD}_{\mathrm{clean}} - \mathrm{LD}_{\mathrm{corrupt}}, \qquad R = \mathrm{LD}_{\mathrm{patch}} - \mathrm{LD}_{\mathrm{corrupt}}.$$

The normalized Effect is R/D, whereas R measures raw logit recovery without a denominator. We examine both on 500 Qwen3-4B examples. As shown in Table 4, the single small-denominator example has a zero denominator and is omitted only from normalized Effect calculations; it remains in the raw-recovery analysis. Raw recovery and normalized Effect identify the same peak layer at every analyzed site. Across layer–site entries, their Pearson and Spearman correlations are 0.99998 and 0.99955. Removing the lowest-margin 5% leaves the layer–site ranking unchanged. For per-head patching, the normalized and raw rankings share $18/20$ top heads before filtering and $20/20$ afterward; the all-head Pearson and Spearman correlations are 0.9984 and 0.9845.

Thus, normalization does not determine the localization patterns in this control. Effect measures recovery relative to the chosen clean–corrupt counterfactual.

E.3 Ablation Study in Stage I

In this subsection, we detail the experimental setup for identifying alignment heads and provide a granular error analysis of the ablation studies discussed in §3.2.

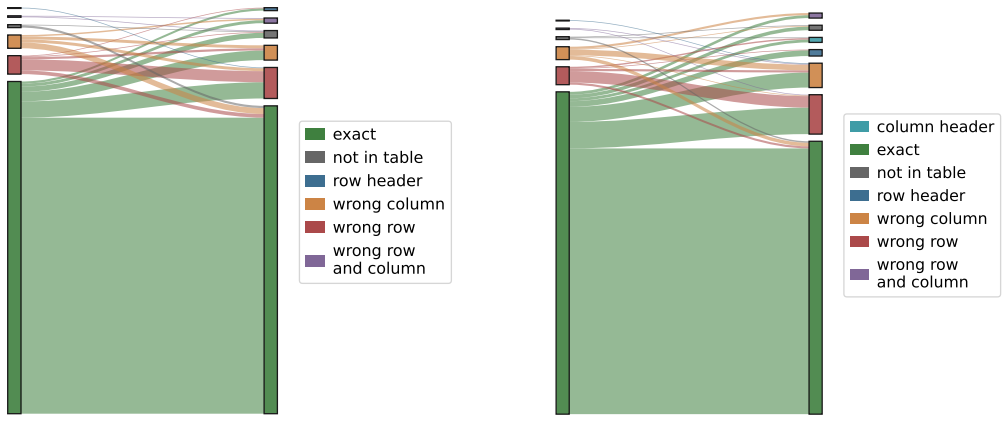
Identification of Alignment Heads To pinpoint the attention heads responsible for Semantic Binding, we define an *Alignment Score* that quantifies the preferential attention flow from query constraints to their corresponding table headers. Let x be the input sequence. We denote the set of token indices corresponding to the row or column constraint in the query as $\mathcal{I}_q \in \{\mathcal{I}_q(r), \mathcal{I}_q(c)\}$. Similarly, let $\mathcal{I}_{tgt} \in \{\mathcal{I}_t(r), \mathcal{I}_t(c)\}$ denote the token indices of the ground-truth header in the table x_{table}, and $\mathcal{I}_{dist}$ denote the indices of all other same-type headers (distractors). For a given attention head h at layer l, we calculate the Alignment Score $S^{(l,h)}$ as the difference between the attention mass focused on the target and the average attention mass focused on distractors:

$$S^{(l,h)} = \frac{1}{|\mathcal{I}_q|} \sum_{i \in \mathcal{I}_q} \left(\sum_{j \in \mathcal{I}_{tgt}} Att_{i,j}^{(l,h)} - \frac{1}{|\mathcal{I}_{dist}|} \sum_{k \in \mathcal{I}_{dist}} Att_{i,k}^{(l,h)} \right) \tag{3}$$

where $Att_{i,j}^{(l,h)}$ represents the attention weight from token i to token j. We compute the global score $\bar{S}^{(l,h)}$ by averaging over $N = 500$ samples. We identify the top-20 heads with the highest $\bar{S}^{(l,h)}$ for rows and columns respectively as *row alignment heads* and *column alignment heads*.

Error Categorization To rigorously assess the impact of ablation, we classify the model's output y_{pred} into seven distinct categories relative to the ground truth value $v_{r,c}$:

- **Exact:** The model correctly generates the target cell value ($y_{pred} = v_{r,c}$).
- **Wrong Row:** The output corresponds to a cell in the correct column but a different row ($y_{pred} = v_{r',c}$ where $r' \neq r$). This error indicates a specific failure in Row Semantic Binding.
- **Wrong Column:** The output corresponds to a cell in the correct row but a different column ($y_{pred} = v_{r,c'}$ where $c' \neq c$). This indicates a failure in Column Localization.
- **Wrong Row and Column:** The output is a value from the table but matches neither the target row nor column ($y_{pred} = v_{r',c'}$).
- **Row Header:** The model hallucinates the row header constraint and output it directly instead of a value ($y_{pred} = r$).
- **Column Header:** The model repeats the column header constraint ($y_{pred} = c$).
- **Not in Table:** The output is not found within the table context x_{table}.



(a) Column alignment heads.

(b) Random heads.

Figure 12: Sankey diagrams illustrating the flow of error types under different ablation conditions. Compared with Figure 2b shows catastrophic failure for row ablation, **(a)** demonstrates that ablating column alignment heads maintains high performance, similar to the **(b)** random baseline.

Result Analysis We compare the ablation of row alignment with column alignment heads and random heads (Figure 12). As established in §3.2, zero-ablating row alignment heads caused a catastrophic drop in performance, with the vast majority of errors shifting to the `wrong row` category. This confirms that specific attention heads are causally necessary for binding row constraints. In contrast, Figure 12a reveals that ablating the top column alignment heads does not lead to a significant performance degradation. The dominant flow remains directed toward the `exact` category, comparable to the random baseline shown in Figure 12b. Specifically, the incidence of `wrong column` errors does not increase disproportionately. This disparity highlights a fundamental mechanistic asymmetry in table understanding. *Row Binding is Explicit:* The model relies on sparse, dedicated attention heads in early layers to point to the correct row header based on semantic similarity. Removing these pointers breaks the location mechanism. *Column Binding is Implicit or Distributed:* The model does not rely on strong semantic attention weights to bind column headers. The resilience of column location to attention ablation suggests that column information is likely processed via the geometric mechanisms described in Stage II (Coordinate Localization), which provides coordinate-structured information beyond semantic alignment alone.

A Direct Causal Link to the Binding Representation We repeat the representation analysis in Figure 2a after ablating the top-20 row-alignment heads selected by Alignment Score. Each

Table 5: Representation-level consequences of alignment-head ablation. Header retrieval is distinct from the task-level answer EM.

Intervention	Header retrieval	Gold–distractor gap
Clean	99.1%	0.42
Row alignment-head ablation	31.7%	0.08
Random-head ablation	92.8%	0.36
Column alignment-head ablation	96.4%	0.39

Table 6: Row-design controls on Qwen3-4B. Ablation columns report task EM.

Setting	Clean EM	Row-head ablation	Column-head ablation
Semantic rows: unique, same entity pool as columns	92.8	7.6	86.9
Non-unique rows: unique composite condition	84.6	8.8	78.9
Transposed axes	52.5	2.5	50.0

multi-token query constraint or candidate header is represented by mean-pooling its post-layer residual states over the full token span. We evaluate query-to-row-header top-1 retrieval and the cosine-similarity gap between the gold header and the strongest distractor.

Row-head ablation disrupts both representation-level metrics, whereas random-head and column-head controls largely preserve them. This intervention directly links the identified heads to the row-binding representation, complementing the original behavioral evidence that the heads are necessary for task accuracy.

Controls for Row–Column Ablation Asymmetry We test whether the row–column ablation asymmetry depends on the unique-primary-key design. The controls use semantic row headers sampled from the same entity pool as columns, non-unique row headers with a unique composite query condition, and transposed axes.

The asymmetry remains in these controls: row-head ablation is strongly damaging while column-head ablation has a smaller effect. This supports row grounding rather than primary-key uniqueness alone as an explanation. The lower clean accuracy after transposition also means that these settings should not be treated as equally difficult.

E.4 Details of Linear Probing and Coordinate Diagnostics

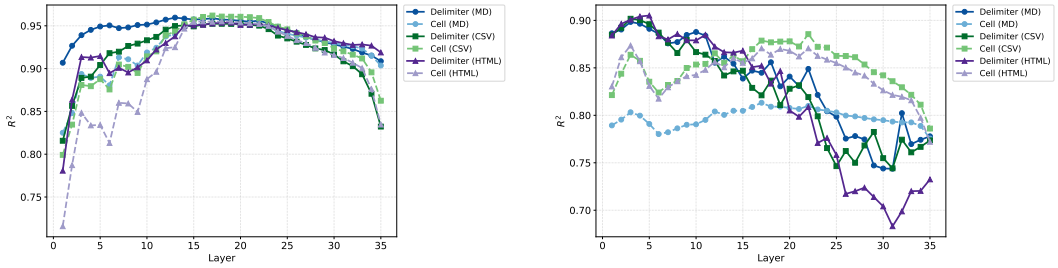
In this subsection, we provide further details regarding the optimization objective and training setup for the linear probes described in §4.

Optimization Objective To mitigate the effects of multicollinearity inherent in high-dimensional neural representations and to improve generalization, we employ Ridge Regression (linear least squares with L_2 regularization). For a specific layer l, the parameters $\mathbf{W}_p$ and $\mathbf{b}_p$ are obtained by minimizing the following regularized loss function:

$$\mathcal{L}(\mathbf{W}_p, \mathbf{b}_p) = \sum_{i=1}^{N} \left(z_i - (\mathbf{W}_p \mathbf{h}_i^{(l)} + \mathbf{b}_p) \right)^2 + \lambda \|\mathbf{W}_p\|_2^2 \tag{4}$$

where:

- N is the total number of tokens used for training the probe.
- $\mathbf{h}_i^{(l)} \in \mathbb{R}^d$ is the activation vector corresponding to the i-th token in the training set.
- $z_i \in \mathbb{R}$ is the ground-truth coordinate (normalized) for the i-th token.
- λ is the regularization strength hyperparameter (set to $\lambda = 1.0$).



(a) Column Probe Performance (R^2)

(b) Row Probe Performance (R^2)

Figure 13: Coefficient of determination (R^2) for linear probes trained on varying tabular formats (MD, CSV, HTML). Solid lines represent probes trained on delimiter tokens corresponding to formats, while dashed lines represent probes trained on cell content.

Coordinate Normalization Since tables in our dataset vary significantly in size, using absolute indices would introduce scale variance. We normalize the geometric coordinates to the unit interval $[0, 1]$. Let R_{max} and C_{max} denote the total number of rows and columns in the table containing the i-th token. The target variable z is defined as:

$$z_{row}^{(i)} = \frac{r_{idx}^{(i)}}{R_{max} - 1}, \quad z_{col}^{(i)} = \frac{c_{idx}^{(i)}}{C_{max} - 1} \tag{5}$$

This normalization changes only the numerical scale of the probe target. It neither assumes continuous coordinate updates nor conflicts with discrete delimiter-triggered transitions.

Training Setup We construct the training dataset by extracting internal representations of specific tokens that correspond to structural markers across the corpus. For each layer l, this yields a dataset of N token-coordinate pairs $\mathcal{D}_l = \{(\mathbf{h}_i^{(l)}, z_i)\}_{i=1}^N$. The dataset is randomly partitioned into a training set (80%) and a held-out evaluation set (20%). We evaluate probe performance using the coefficient of determination:

$$R^2 = 1 - \frac{\sum_{i=1}^N (z_i - \hat{z}_i)^2}{\sum_{i=1}^N (z_i - \bar{z})^2}. \tag{6}$$

The R^2 scores are reported on the evaluation set.

Per-head probes used for causal selection. The residual-stream probes above differ from the head-selection probes used in Table 1. For that table, we fit the row- and column-coordinate targets separately to each attention-head output. We construct distinct probe datasets from cell-content tokens and structural-delimiter tokens, rank heads by held-out R^2 within each token set, and activation-patch the top-20 selected heads at the target-cell positions. A full-residual probe score is therefore not reused as a per-head score.

Virtual-RoPE Coordinate Diagnostic and Controls A native attention head uses rotary positions determined by the token sequence, with the Q/K interaction

$$\mathrm{RoPE}(q_t, t)^\top \mathrm{RoPE}(k_s, s).$$

In contrast, our counterfactual diagnostic assigns logical column indices to column-averaged Q/K states:

$$S_{j,l}^{(\alpha)} = \mathrm{RoPE}(\bar{q}_j, \alpha j)^\top \mathrm{RoPE}(\bar{k}_l, \alpha l).$$

Here, j is the query's target-column index and l is the table-column index. The diagnostic tests coordinate-structured Q/K geometry. The original diagnostic in Figure 3 uses $\alpha = 1$. In the additional held-out evaluation, one global α is selected on calibration tables and frozen for evaluation on three non-overlapping, target-balanced held-out folds. The scale is not fitted separately for individual examples, heads, columns, or targets. With 16 candidate columns, chance gold-column top-1 is approximately 6.25%.

The column-index diagnostic achieves high gold-column top-1, whereas the actual-position, permutation, and content-only controls are near chance, and the random heads do not recover the gold column

Table 7: Held-out controls for the Virtual-RoPE diagnostic. Pearson, gold-column top-1 and gold–best-off-target margin are distinct metrics.

Method	**Pearson**	**Gold top-1**	**Gold–best margin**
Column-index Virtual RoPE	$.878 \pm .144$	$93.8\% \pm 6.3\%$	$.231 \pm .081$
Actual token-position RoPE	$.297 \pm .199$	$6.3\% \pm 6.3\%$	$-.170 \pm .035$
Independent Q/K permutation	$.061 \pm .192$	$6.5\% \pm 3.2\%$	$-.583 \pm .013$
Key-only permutation	$.099 \pm .065$	$6.7\% \pm 1.0\%$	$-.583 \pm .010$
Content-only $\bar{q}_j^\top \bar{k}_l$	$.034 \pm .370$	$8.3\% \pm 3.6\%$	$-.242 \pm .094$
Layer-matched random heads	$.334 \pm .330$	$0.0\% \pm 0.0\%$	$-.348 \pm .007$

Table 8: Head-count sensitivity. Strict top-1 is shown as a fraction; EM and wrong-row rates are percentages.

Metric	$\boldsymbol{k = 5}$	$\boldsymbol{k = 10}$	$\boldsymbol{k = 20}$	$\boldsymbol{k = 40}$
Row-head ablation EM	95.0%	45.0%	2.5%	2.5%
Wrong-row rate	5.0%	32.5%	25.0%	25.0%
Column-head ablation EM	95.0%	92.5%	90.0%	82.5%
Canonical QK Pearson	.070	.026	.997	.997
Strict top-1	.056	.611	.722	.778

in this evaluation. These comparisons support coordinate-specific geometry rather than a diagonal induced generically by applying RoPE. With fixed $\alpha = 1$, an additional control gives Pearson 0.996 and top-1 55.6%, versus $9.4\% \pm 6.0\%$ and $7.8\% \pm 6.8\%$ top-1 for the two permutation controls. The original aggregate (0.997 Pearson), fixed-scale control, and calibrated held-out evaluation are reported separately rather than treated as interchangeable measurements.

Sensitivity to the Number of Selected Heads We vary the number of selected heads, $k \in \{5, 10, 20, 40\}$, for semantic-head ablation and the canonical coordinate diagnostic. Table 8 distinguishes task-level ablation metrics from coordinate-geometry metrics. Row-head ablation saturates by $k = 20$, while column-head ablation changes more gradually. The canonical coordinate aggregate stabilizes at $k = 20$–40; its small-k behavior should not be interpreted as monotonic evidence in the number of heads.

To distinguish subset size from ranking variability, we also evaluate nested prefixes of a fixed cross-validated head ranking. At $k = 5$ and $k = 10$, Pearson is 0.960 ± 0.085 and 0.9966 ± 0.0018, respectively, with strict top-1 of 86.9% and 80.7%. Thus, smaller fixed-ranking subsets can already recover the coordinate pattern, while $k = 20$ provides a stable operating point for the reported analyses. These sweeps concern semantic-head ablation and the coordinate diagnostic. The cell- versus delimiter-probe-selected causal patching in Table 1 is a separate comparison reported at $k = 20$; the present sweeps do not establish its effect-versus-k curve.

E.5 Noise Intervention

To rigorously test the structural dependencies identified in our probing and patching experiments, we introduce targeted noise intervention to the table context x_{table}. We design two intervention conditions to decouple token distance from structural count: (*i*) **Structural Injection:** Inserting additional pipe delimiters ('|') to simulate false coordinate boundaries. (*ii*) **Length Injection:** Inserting random alphanumeric characters of equivalent token length to test resilience against context expansion.

Table 9 presents the performance impact under these noise settings. Consistent with our hypothesis that coordinates are synchronized solely by delimiters, the model exhibits remarkable robustness to Length Injection. The negligible performance drop indicates that the internal counting mechanism is invariant to absolute token distance, suggesting a reliance on discrete delimiter counting rather than token length proximity. Conversely, Structural Injection preceding the target column induces a catastrophic failure. Crucially, delimiter insertions succeeding the target column exert minimal influence, confirming that the counting mechanism is strictly sequential.

Table 9: Accuracy comparison of Qwen3-4B under different noise settings. "Baseline" denotes performance on clean inputs without noise injection. "Noise +" and "+ Noise" indicate noise insertion preceding and succeeding the target column within the target row, respectively. The highest scores are highlighted in **bold**.

Noise Type	Baseline	Noise +	+ Noise
Structural Injection	**90.0**	32.8	88.8
Length Injection		86.5	88.0

Table 10: Detailed results of the Vector Composition Experiment. **Target** denotes the desired column shift k. **Baseline** indicates the vector was extracted directly from column pairs with distance k; **Composition** describes how the vector was constructed. **Effect** reports the Effect Score (higher is better).

Composition ($a + b$)	Effect
Target -1 Baseline: 0.301	
$-6 + 5$	0.244
$-5 + 4$	0.243
$-4 + 3$	0.242
$-3 + 2$	0.251
$-2 + 1$	0.280
$1 + (-2)$	0.280
$2 + (-3)$	0.251
$3 + (-4)$	0.242
$4 + (-5)$	0.243
$5 + (-6)$	0.244
Target 1 Baseline: 0.197	
$-5 + 6$	0.191
$-4 + 5$	0.253
$-3 + 4$	0.210
$-2 + 3$	0.249
$-1 + 2$	0.282
$2 + (-1)$	0.282
$3 + (-2)$	0.249
$4 + (-3)$	0.210
$5 + (-4)$	0.253
$6 + (-5)$	0.191
Target 2 Baseline: 0.454	
$-4 + 6$	0.432
$-3 + 5$	0.437
$-2 + 4$	0.430
$-1 + 3$	0.410
$1 + 1$	0.499
$3 + (-1)$	0.410
$4 + (-2)$	0.430
$5 + (-3)$	0.437
$6 + (-4)$	0.432
Target -2 Baseline: 0.310	
$-6 + 4$	0.284
$-5 + 3$	0.284
$-4 + 2$	0.293
$-3 + 1$	0.305
$-1 + (-1)$	0.301
$1 + (-3)$	0.305
$2 + (-4)$	0.293
$3 + (-5)$	0.284
$4 + (-6)$	0.284

Composition ($a + b$)	Effect
Target 3 Baseline: 0.467	
$-3 + 6$	0.469
$-2 + 5$	0.466
$-1 + 4$	0.467
$1 + 2$	0.479
$2 + 1$	0.479
$4 + (-1)$	0.467
$5 + (-2)$	0.466
$6 + (-3)$	0.469
Target -3 Baseline: 0.361	
$-6 + 3$	0.351
$-5 + 2$	0.356
$-4 + 1$	0.364
$-2 + (-1)$	0.350
$-1 + (-2)$	0.350
$1 + (-4)$	0.364
$2 + (-5)$	0.356
$3 + (-6)$	0.351
Target 4 Baseline: 1.079	
$-2 + 6$	1.004
$-1 + 5$	1.012
$1 + 3$	1.169
$2 + 2$	1.238
$3 + 1$	1.169
$5 + (-1)$	1.012
$6 + (-2)$	1.004
Target -4 Baseline: 0.367	
$-6 + 2$	0.369
$-5 + 1$	0.370
$-3 + (-1)$	0.357
$-2 + (-2)$	0.357
$-1 + (-3)$	0.357
$1 + (-5)$	0.370
$2 + (-6)$	0.369

E.6 Details on the Linear Geometry of Tabular Indices

In this subsection, we provide further implementation details and comprehensive results for the vector composition experiments discussed in §5.

Vector Extraction and Intervention Setup Based on preliminary probing results indicating that column information is most localized in the middle layers, we restrict our extraction and intervention to layers $L \in \{17, \ldots, 23\}$. For a given column header token at position t in x_{table}, we extract the residual stream activations at positions $[t-1, t]$ (i.e., the header token and the immediate previous token) to capture the representations of column positions. The unit shift vectors and composite vectors

are computed by averaging differences across $N = 100$ samples from the held-out benchmark. During the steering experiments, we inject the computed vectors into the residual stream with a steering coefficient of $\alpha = 8.0$. The intervention is applied simultaneously to the same token positions $[t-1, t]$ used during extraction.

Vector Composition Analysis We validate the linearity of the representation space by comparing the causal effect of a "true" shift vector (Baseline) against a "composite" vector formed by summing two distinct shift vectors (e.g., aiming for a shift of $+4$ by summing vectors for $+2$ and $+2$). Table 10 details the Effect scores for various target shifts $k \in \{-4, \ldots, 4\}$. The results strongly support the hypothesis of a linear subspace. For instance, for a target rightward shift of $k = 4$, the baseline vector yields an Effect score of 1.079. The composite vector formed by $\vec{v}_{+2} + \vec{v}_{+2}$ yields an Effect score of 1.238, actually surpassing the baseline, suggesting a strong constructive interference of the positional signal. Similarly, for $k = 2$, the additive combination $\vec{v}_{+1} + \vec{v}_{+1}$ achieves an Effect score of 0.499, compared to the baseline of 0.454. Even for subtractive compositions (e.g., reaching a target of -1 via $\vec{v}_{+1} + \vec{v}_{-2}$), the model maintains robust performance (Effect Score 0.280 vs. Baseline 0.301), confirming that the geometric representation supports standard vector arithmetic operations.

E.7 Robustness of the Linear-Subspace Interpretation

To rule out alternative explanations for the steering results, we conduct a set of targeted robustness analyses on the column-index subspace. Specifically, we test whether the observed direction is an averaging artifact, whether it transfers across different table serializations and prompt templates, and whether it depends on a particular construction of the shift vector. These experiments provide consistent evidence that the steering effect reflects a stable coordinate direction in the residual stream, while also clarifying the boundary of this interpretation.

Instance-level consistency and averaging stability. A possible concern is that the global shift vector $\vec{v}_{unit}$ is created only after averaging many unrelated instance-level directions. To test this, we perform bootstrap resampling over the extraction set and compare each bootstrap vector with the full-data vector. The cosine similarity reaches 0.9956, indicating that the extracted direction is highly stable under resampling. We further compute single-instance directional alignment before averaging; the cosine similarity is 0.7668, showing that individual examples already contain a reasonably aligned column-shift direction. Thus, the direction is not merely an averaging artifact, but is already present at the instance level.

Table 11: Cross-format transfer under the original header-neighborhood extraction protocol. HTML-specific boundary extraction is evaluated separately in Appendix E.7.

Extract Format	Apply Format	Effect Score
Markdown	Markdown	0.259
Markdown	CSV	0.229
Markdown	HTML	0.000
CSV	Markdown	0.261
CSV	CSV	0.236
CSV	HTML	0.000
HTML	Markdown	0.000
HTML	CSV	0.000
HTML	HTML	0.000

Different serializations. We next test whether the linear-subspace interpretation depends on the specific Markdown serialization used in the main experiments. We extract the column shift vector from one table format and apply it to another format while keeping the underlying cell location task fixed. As shown in Table 11, the effect transfers between Markdown and CSV: for example, extracting and applying on Markdown gives an Effect Score of 0.259, while extracting from Markdown and applying to CSV still yields 0.229. The reverse direction is similar, with CSV-extracted vectors reaching 0.261 on Markdown and 0.236 on CSV. Under this original protocol, the effect largely disappears for HTML. HTML-boundary-specific extraction yields a held-out Effect Score of 0.154 (Appendix E.7), supporting serialization-dependent steering directions rather than the absence of an HTML coordinate representation.

Prompt templates. We also vary the question wording and prompt template while preserving the same underlying table lookup task. Across these prompt variants, the Effect Score remains within a similar range, from 0.161 to 0.256. This indicates that the intervention does not depend on a single lexical pattern in the query, but instead acts on a more abstract representation of the target column coordinate.

Table 12: Held-out HTML steering with format-specific extraction. This evaluation is distinct from the original protocol in Table 11.

Vector extraction	Application format	Effect Score
Markdown	HTML	0.021
CSV	HTML	0.005
HTML-specific boundaries	HTML	0.154

Table 13: Cosine similarity of corresponding 32-/64-column directions. Results are shown at selected offsets.

Offset	Cross-width direction cosine
-1	0.790
$+1$	0.843
-4	0.773
$+4$	0.780
-16	0.685
$+16$	0.715

Alternative shift-vector constructions. Finally, we reconstruct the shift vector using multiple extraction strategies, including header-based, cell-based, and all-pairs constructions, and compare them with a shuffled control. Semantically aligned constructions consistently outperform the control: representative Effect Scores reach 0.1899 and 0.1778, whereas the shuffled control reaches only 0.0660. Therefore, the steering effect is robust to the specific construction method as long as the extracted direction preserves the semantic correspondence between source and target column offsets. Together, these controls support approximately additive column representations in the tested regime, while the boundary and format-specific tests below delimit the steering claim.

HTML-Specific Boundary Extraction The original cross-format results in Table 11 use the header-neighborhood extraction protocol. HTML distributes structural boundaries across `<td>`/`</td>` and `<tr>`/`</tr>` tags, so that protocol need not expose an effective HTML steering direction. We therefore extract a direction from HTML structural-boundary states and evaluate it on held-out HTML tables, fixing intervention settings before evaluation. As shown in Table 12, HTML-specific extraction recovers a nonzero causal steering effect, while cross-format directions remain weak. Together with the HTML probes in Appendix G.4, this supports linearly accessible coordinate states without requiring an unchanged steering direction or token-boundary intervention across serializations. The effect does not establish Markdown-level steering performance on HTML.

Transfer across Widths and Limits at Large Offsets We evaluate same-width and cross-width steering on 32- and 64-column tables with offsets $k \in \{\pm 1, \pm 4, \pm 8, \pm 16, \pm 24\}$. Table 13 reports cosine similarities between corresponding directions at selected offsets. Directional agreement declines further at offsets ± 24. Cross-width vectors retain positive causal effects and recover 62.3% of same-width steering in the reported aggregate, but weaken at large offsets and near table boundaries. These results support transferable local coordinate directions, not indefinite global linearity or an absence of boundary effects.

E.8 HiTab Evaluation: Prediction Types, Coverage, and Hierarchy

The results reveal both the potential and the limitation of current LLMs on realistic tables. On the one hand, the model often recovers the correct answer even when the reported evidence is incomplete or inaccurate, suggesting that it can exploit table structure to extract useful information in practical settings. On the other hand, the low ratio of fully exact predictions and the large fraction of evidence-related errors indicate a substantial gap between answer correctness and faithful table grounding. In addition, row and column localization errors remain frequent, reflecting the difficulty of resolving coordinates in hierarchical tables where headers span multiple rows or columns. These observations motivate a mechanistic analysis of whether the internal pipeline discovered in synthetic tables still governs successful HiTab reasoning.

Table 14: Prediction type distribution of Qwen3-4B on HiTab. Percentages are computed over all evaluated samples. The taxonomy separates answer correctness, evidence faithfulness, aggregation errors, and row/column localization failures.

Prediction type	Ratio
answer correct wrong evidence	30.49%
answer correct evidence incomplete	15.09%
wrong aggregation on gold cells	14.58%
same row wrong col	11.11%
same col wrong row	8.59%
unsupported answer	7.20%
malformed output	4.86%
partial evidence but wrong answer	3.54%
other table value confusion	2.84%
header value confusion	0.95%
exact	0.57%
header only confusion	0.19%

Table 15: HiTab stage-specific coverage out of 1,584 evaluated examples. Counts describe the separate analysis subsets, not a joint-exact set.

Stage	Retained examples
I: Semantic Binding	655
II: Coordinate Localization	479
III: Information Propagation	712

Stage-Specific Coverage and Hierarchy-Stratified Diagnostics We evaluate 1,584 HiTab examples. The three stage analyses use stage-specific eligible subsets, as shown in Table 15. We also partition the evaluated examples by header hierarchy. The corresponding diagnostic rates are summarized in Table 16. Both reported signals decline with hierarchical complexity. The stage-conditioned evidence therefore identifies observable functional signatures, not robust solving of hierarchical tables or resolved merged-cell reasoning.

F Error Analysis

To understand how different failure modes arise within the proposed three-stage pipeline, we separately analyze Qwen3-4B examples with row-location and column-location errors.

Row-location errors. For examples with incorrect row localization, we examine the behavior of *row alignment heads* in Stage I: Semantic Binding. After zero-ablating these heads, 44.0% of examples produce outputs that either fall outside the table context or directly repeat a row header. We further compute the Alignment Score between the query constraint and the target row header using the definition in Appendix E.3. The average score is 0.209 on correct examples, but drops to 0.146 on row-error examples. This indicates that row-location errors arise when the model fails to bind the query-side row constraint to the correct table-side row header in Stage I.

Column-location errors. For examples with incorrect column localization, we analyze the *coordinate-encoding heads* in Stage II: Coordinate Localization. Unlike correct examples, the geometric interaction score $S_{j,l}$ no longer exhibits a clear diagonal pattern. Only 9.09% of examples attain the maximum interaction score at the correct column, whereas 77.8% attain the maximum score at the incorrectly predicted column. This confirms that column-location errors mainly result from failures in structural coordinate localization rather than semantic binding.

Similarity of predicted and correct locations. We also compare the predicted incorrect row/column with the correct row/column using three metrics: normalized string similarity based on edit distance, cosine similarity, and structural proximity. For structural proximity, we mark a prediction as similar if the predicted row or column lies within ± 2 indices of the correct one. All metrics are normalized so that larger values indicate higher similarity.

Table 16: Hierarchy-stratified HiTab diagnostics. The second column records each stratum's total size; the last two columns report its binding and coordinate diagnostic rates.

HiTab stratum	Total N	Row binding	Gold-column QK peak
No merged headers	84	83.3%	36.7%
Single merged level	269	80.0%	33.3%
Multi-level nested	1,231	63.3%	23.3%

Table 17: HiTab failure diagnostics. A dash means that a value is not reported for that error group. Support-cell attention is a fraction of table-directed attention.

Diagnostic	Same row, wrong column	Aggregation errors
Row heads prefer the gold row	93.0%	–
Row-head gold-row gap	+0.0342	–
Evidence contains the gold row	73.3%	–
Evidence contains the gold column	20.0%	–
Coordinate heads peak at the gold column	16.7%	33.3%
Coordinate gold–best margin	−0.0238	−0.0112
Required-support attention / table-directed attention	–	9.44%

The predicted incorrect row header is more similar to the correct row header than the predicted incorrect column is to the correct column, both lexically and semantically. Specifically, row errors obtain string and cosine similarities of 0.158 and 0.013, respectively, compared with 0.083 and 0.005 for column errors. This suggests that Stage I relies more on semantic representations when identifying row headers, making semantically similar row headers a common source of row-location errors.

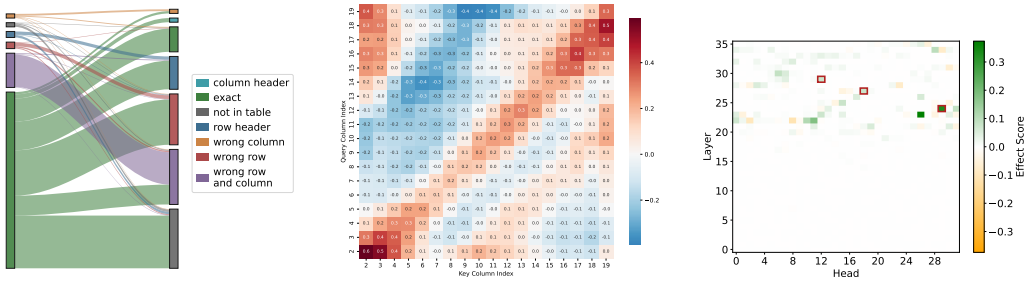
In contrast, column errors exhibit stronger structural proximity. The predicted incorrect column lies within ± 2 columns of the correct column in 70.6% of cases, whereas the corresponding proximity for row errors is only 47.4%. This supports our interpretation that Stage II relies more on coordinate indexing for column localization, and that errors in delimiter counting or coordinate tracking can shift the model to a nearby but incorrect column.

Failure Diagnostics on HiTab We complement the stage-conditioned analyses with diagnostics on HiTab failures, focusing on same-row, wrong-column predictions and aggregation errors. These are distinct from the synthetic error analysis above and from the joint-exact evaluation category. As shown in Table 17, in same-row, wrong-column errors, row grounding often remains detectable while the coordinate diagnostic mislocalizes the column. In aggregation errors, the gold column is selected inconsistently and required support cells receive only a small fraction of table-directed attention, indicating weak support-cell propagation as well as localization instability. Together with the hierarchy-stratified results in Appendix E.8, these observations identify hierarchy-to-coordinate mapping as a major bottleneck without claiming that every HiTab failure follows an identical causal pathway.

Pre-Output Diagnostics of Column Errors We evaluate a lightweight classifier using only Stage-II features: QK diagonal correlation, the target-versus-best-off-target column margin, or both. Features are computed before answer decoding and do not include the generated answer. In this cell-location setting, the query specifies the target column, making the target-column margin available from the input and internal states before decoding. As shown in Table 18, the coordinate features outperform the random-head control in the reported estimates, but performance is moderate and variable. This is an above-control diagnostic signal, not a high-accuracy standalone or deployment-ready correctness detector. The input-resolvable target column in this experiment should not be assumed available for every hierarchical or implicit real-world query.

Table 18: Pre-output Stage-II diagnostics. The classifier uses no features from the generated answer.

Signal	Accuracy	Balanced accuracy	AUROC
QK diagonal	$.685 \pm .073$	$.553 \pm .062$	$.538 \pm .069$
Target-column margin	$.619 \pm .094$	$.598 \pm .118$	$.690 \pm .139$
Both	$.633 \pm .074$	$.576 \pm .052$	$.676 \pm .127$
Random-head control	$.408 \pm .113$	$.433 \pm .137$	$.367 \pm .115$



(a) Semantic Binding (b) Coordinate Localization (c) Information Propagation

Figure 14: Mechanistic analysis of Qwen3-4B in the zero-shot setting. **(a)** Zero-ablating row alignment heads causes 69.2% of originally correct predictions to become `not in table` errors. **(b)** Coordinate-encoding heads still exhibit a clear diagonal pattern in the geometric interaction score $S_{j,l}$, with Pearson correlation 0.979. **(c)** Late-layer information mover heads are again identified on correctly answered examples, with average attention to the answer token exceeding 30%.

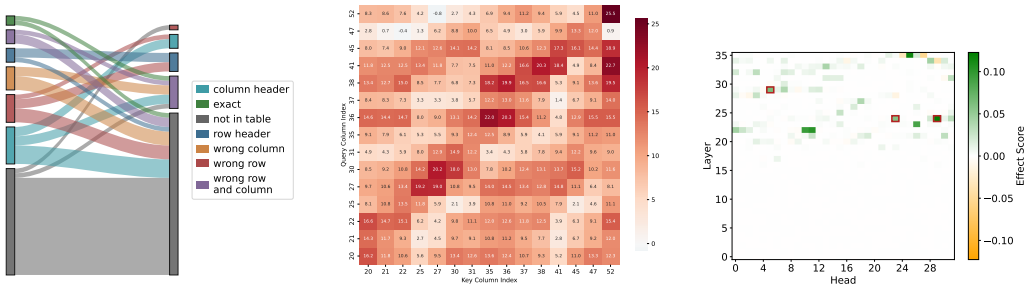
G Generalization Experiments

Before presenting the individual generalization experiments, we emphasize that these settings are not exact replications of the main experimental setup, but deliberate stress tests along different axes, including prompting format, table scale, inference mode, serialization format, and task complexity. Such distributional changes can reduce the model's end-to-end task performance, which in turn may make the corresponding mechanistic signatures less visually salient or quantitatively pronounced than in the high-performance setting analyzed in the main text [24, 16]. This attenuation is expected: when the model succeeds less reliably, the internal computation contains more failed or noisy trajectories, and causal effects, geometric interaction patterns, or information-moving behavior may therefore appear weaker. Consequently, we interpret the following results by focusing on whether the same structural patterns persist under each shift, rather than requiring their magnitudes to match those observed under the original controlled setting.

G.1 Generalization to Zero-shot Setting

We further evaluate Qwen3-4B under the zero-shot setting, where the one-shot demonstration D is removed from the input prompt. This setting tests whether the three-stage mechanism depends on the demonstration format or instead reflects a more intrinsic table-location circuit. As shown in Figure 14, the same mechanistic pattern remains observable.

For Stage I, ablating the identified *row alignment heads* sharply disrupts row grounding: 69.2% of predictions that were originally correct are redirected to answers not present in the table. This confirms that explicit row-level Semantic Binding is still causally necessary without an in-context demonstration. For Stage II, the *coordinate-encoding heads* preserve a strong diagonal structure in $S_{j,l}$, indicating that the model continues to align query-side column indices with table-side column indices through the same geometric localization mechanism. Finally, after retaining only examples with correct answers, we again identify late-layer *information mover heads*; their attention mass on the answer token exceeds 30% on average. These results suggest that the three-stage pipeline is not induced by the one-shot prompt format, but generalizes to zero-shot table localization.



(a) Semantic Binding (b) Coordinate Localization (c) Information Propagation

Figure 15: Mechanistic analysis of Qwen3-4B on larger tables with 20–55 rows and columns. **(a)** After ablating row alignment heads, 27.0% of predictions shift from other categories to `not in table`, making it the dominant error type at 70.0%. **(b)** On examples with correct column localization, coordinate-encoding heads retain a diagonal geometric interaction pattern, with Pearson correlation 0.917. **(c)** On correctly answered examples, late-layer information mover heads again attend strongly to the answer token, with average attention exceeding 30%.

G.2 Generalization to Larger Tables

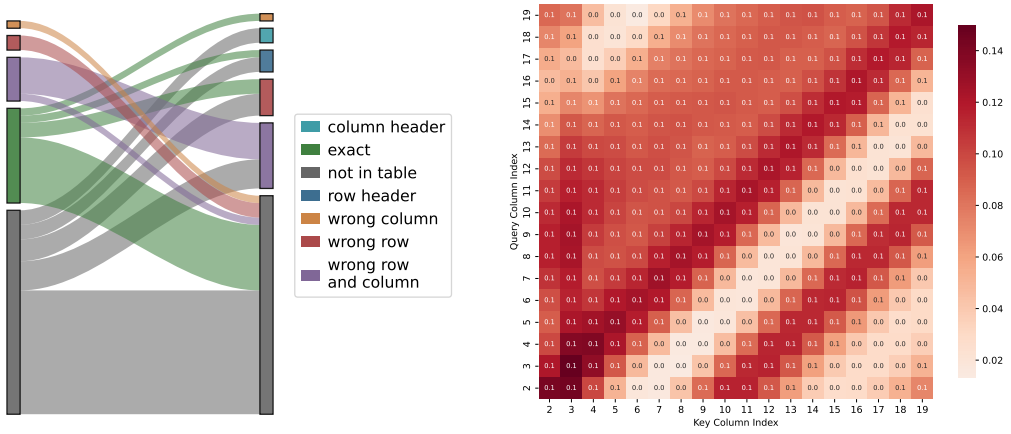
We next test whether the mechanism scales to larger tables. Specifically, we conduct the same mechanistic analysis with Qwen3-4B on tables containing 20–55 rows and columns. This setting substantially increases the coordinate range and therefore provides a stricter test of whether the model's internal table-location mechanism can operate beyond the table sizes used in the main experiments. The results are summarized in Figure 15.

In Stage I, zero-ablating the *row alignment heads* causes a substantial increase in unsupported generations: 27.0% of predictions move from other output categories to `not in table`, and `not in table` becomes the dominant error type, accounting for 70.0% of outputs after ablation. This indicates that, as the table grows, losing row grounding no longer merely causes confusion among nearby table cells; instead, it often prevents the model from producing any table-supported value. In Stage II, we restrict the analysis to the subset where column localization is correct and identify *coordinate-encoding heads* using the same criterion as in the main text. Despite the expanded coordinate range, the geometric interaction score still exhibits a clear diagonal structure, with Pearson correlation 0.917. In Stage III, among correctly answered examples, we again identify *information mover heads* whose average attention to the answer token exceeds 30%. Together, these findings show that the three-stage mechanism extends to substantially larger tables, although the stronger `not in table` failure mode suggests that larger tables place higher pressure on stable Semantic Binding.

G.3 Generalization to Non-Thinking Mode

We also evaluate Qwen3-4B in Non-Thinking Mode to test whether the identified mechanism depends on explicit thinking behavior. As shown in Figure 16, the results remain consistent with those observed in Thinking Mode.

For Stage I, the same *row alignment heads* remain causally important: after ablation, 69.2% of originally correct predictions become incorrect answers that are not present in x_{table}. This mirrors the failure mode observed in Thinking Mode and shows that row Semantic Binding is still a prerequisite for grounded table extraction. For Stage II, we again identify *coordinate-encoding heads*; their geometric interaction score $S_{j,l}$ forms a sharply diagonal pattern, with Pearson correlation 0.999. This near-perfect alignment indicates that Non-Thinking Mode still relies on a RoPE-mediated coordinate localization mechanism to map query-side column indices to table-side positions. For Stage III, applying the same final-token analysis identifies late-layer *information mover heads*, showing that the located value is again propagated to the final residual stream for decoding. Therefore, the three-stage pipeline is not specific to Thinking Mode; rather, it persists when the model directly produces the answer without explicit reasoning traces.



(a) Semantic Binding

(b) Coordinate Localization

Figure 16: Mechanistic analysis of Qwen3-4B in Non-Thinking Mode. **(a)** Ablating row alignment heads turns 69.2% of originally correct predictions into `not in table` errors. **(b)** Coordinate-encoding heads preserve an almost perfect diagonal pattern in $S_{j,l}$, with Pearson correlation 0.999.

G.4 Generalization across Tabular Formats

To ensure that the implicit coordinate system observed in §4 is not an artifact specific to the Markdown format, we extend our probing analysis to two additional common tabular serialization formats: Comma-Separated Values (CSV) and HyperText Markup Language (HTML).

Experimental Setup We adhere to the same data generation protocol described in §4, varying only the serialization function $\phi(T)$. For a table T with columns $\mathcal{C}$ and rows $\mathcal{R}$:

- **Markdown (MD):** Uses the pipe character ('|') as the cell delimiter and newlines for row boundaries.
- **CSV:** Uses the comma (',') as the cell delimiter and newlines for row boundaries.
- **HTML:** Uses hierarchical tags, where `<td>` and `</td>` delineate cells, and `<tr>` and `</tr>` delineate rows.

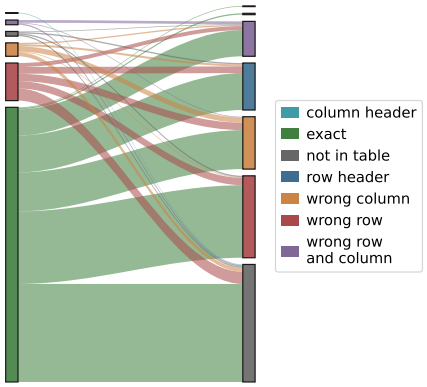
We train linear probes on the intermediate activations of these sequences to predict column indices (c_{idx}) and row indices (r_{idx}).

Results and Analysis Figure 13 presents the layer-wise R^2 scores for column and row probes across the three formats. As shown in Figure 13a, the model successfully constructs linear representations of column indices across all formats, achieving peak $R^2 > 0.90$ in the middle layers (Layers 17-23). Crucially, the hierarchical lag between delimiter probes (solid lines) and cell probes (dashed lines) is preserved regardless of the format. For instance, in HTML, the probe on the `</td>` tag consistently achieves high accuracy several layers before the probe on the enclosed cell content. These probe results support coordinate representations and earlier delimiter-associated decodability across the tested formats.

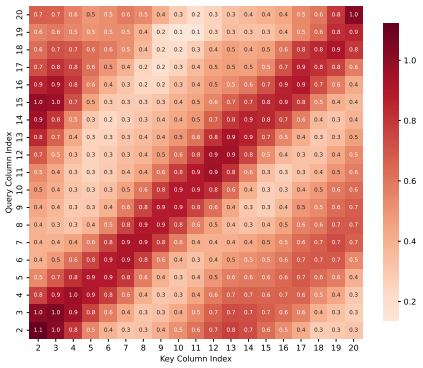
G.5 Generalization to Multi-Cell Location

G.5.1 Experimental Setup

To evaluate the generalization of the internal pipeline, we extend the atomic cell location task to multi-cell location scenarios. We construct two distinct benchmarks each consisting of 500 samples (details in Appendix D). These benchmarks retain the same tables with the benchmark of the atomic cell location task, with the modifications applied solely to the natural language questions. In the *Multi-Row* setting the query x_{query} explicitly references a subset of target row headers $\mathcal{R}_{sub} \subset \mathcal{R}$ and a single target column header $c \in \mathcal{C}$. The objective is to locate the set of values $\{v_{r,c} \mid r \in \mathcal{R}_{sub}\}$ corresponding to the intersection of the specified rows and the target column. In the *Multi-Column*



(a) Error distribution post-ablation.



(b) Multi-threaded coordinate localization.

Figure 17: Mechanistic analysis on the multi-cell location task. **(a)** illustrates the impact of ablating the row alignment heads identified in Stage I on multi-row queries. **(b)** depicts the Query-Key interaction scores for the top-20 coordinate-encoding heads from Stage II on multi-column queries.

setting the query specifies a single row header $r \in \mathcal{R}$ and a subset of target column headers $\mathcal{C}_{sub} \subset \mathcal{C}$ requiring the location of values $\{v_{r,c} \mid c \in \mathcal{C}_{sub}\}$. This design allows us to investigate whether the model resolves multiple constraints through sequential processing or by parallelizing the semantic binding and coordinate localization mechanisms.

G.5.2 Parallel Semantic Binding

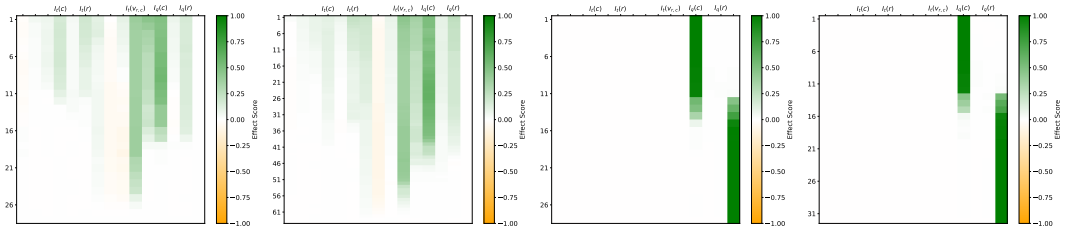
We investigate whether the semantic binding mechanism identified in the atomic cell location task generalizes to multi-row scenarios. We hypothesize that the model reuses the specifically identified *row alignment heads* from Stage I (§3.2) to process multiple constraints in parallel rather than instantiating distinct mechanisms. Specifically, this parallelism is inherent to the self-attention mechanism, where these heads compute binding scores for all query constraints concurrently across the sequence dimension. To verify this we perform an ablation study where we zero-ablate the top-20 row alignment heads, which are discovered in the atomic cell location task, during the processing of multi-row queries and observe changes of model outputs.

Figure 17a visualizes the distribution of error types following the ablation. The Sankey diagram reveals a catastrophic collapse in performance ($81.8\% \rightarrow 0.4\%$). Specifically, 38.4% of samples shifting from `exact` matches to the `wrong row` or `row header` (directly outputting the row headers in the query) error categories. This specific failure mode mirrors the degradation observed in the atomic cell location setting, confirming that the model relies on the identical set of functional components to resolve multiple semantic targets. This demonstrates that the model does not resort to an iterative or multi-step reasoning strategy; instead, it possesses the capacity for parallel execution, where the row alignment operator is instantiated across representation sites within the latency of a single forward pass.

G.5.3 Multi-Threaded Coordinate Localization

We subsequently examine the coordinate localization mechanism for multi-column queries. Crucially, we utilize the top-20 *coordinate-encoding heads* previously identified in the atomic cell location analysis (Stage II in §3.3) to explore the model's behavior in multi-column settings. We apply the RoPE-based interaction metric to the heads to compute the geometric interaction scores between the query column representations and the key vectors of the table columns.

Figure 17b displays the average interaction scores for these specific top-20 coordinate-encoding heads. The heatmap reveals that the attention patterns of these heads exhibit strong positive interaction scores on the indices that correspond to all queried columns simultaneously. The preservation of the diagonal alignment pattern across multiple targets confirms that the coordinate-encoding heads are not limited to single-state tracking. This implies that the linear coordinate system operates as



(a) Qwen3-0.6B (b) Qwen2.5-32B (c) Llama-3.2-3B (d) Llama-3.1-8B

Figure 18: Average column patching effect score across layers and token positions on various LLMs.

a superposition-ready vector space, where the same set of attention heads can calculate geometric offsets for distinct targets in parallel and locate the corresponding values in a single forward pass.

H Experiments of LLMs

In this section, we show the experiments and analysis of other evaluated models (including Qwen, Llama, Mistral, and Gemma models). We conclude that the findings in the main text can generalize to the models with different scales successfully.

H.1 Evaluation of LLMs

In this subsection, we present the Exact Match (EM) scores of various models on both the atomic and multi-cell location benchmarks. The EM metric requires the model's response to exactly match the gold answer. All results are obtained using greedy decoding with temperature set to 0. As shown in Table 19, Qwen3-4B and Qwen2.5-32B achieve the strongest performance on both settings, whereas the smaller or less table-specialized models exhibit a larger drop on the multi-cell task. This pattern suggests that scaling and model family both affect the robustness of table localization under increased compositional complexity. In the following subsections, we analyze whether the underlying mechanism remains consistent across models, and use the observed differences to better understand the source of the performance gap.

Table 19: EM performance (%) of evaluated models on the atomic and multi-cell location benchmarks.

Model	Atomic	Multi
Qwen3-0.6B	38.8	8.0
Qwen3-4B	90.0	88.4
Qwen2.5-32B	**98.0**	**95.0**
Llama-3.2-3B	52.8	26.0
Llama-3.1-8B	70.0	38.0
Mistral-7B-v0.1	89.8	81.2
Gemma-3-4b-it	45.6	6.0

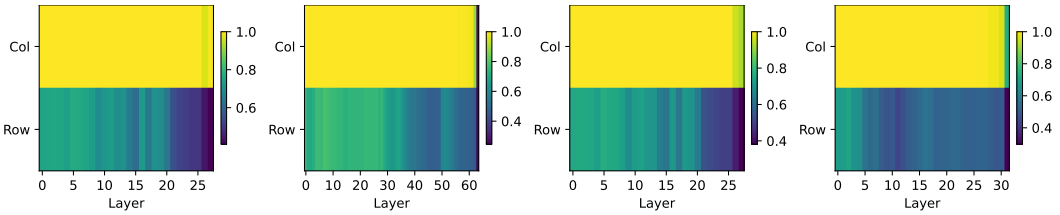
H.2 The Three-Stage Information Flow of Table Understanding

H.2.1 Overview

To evaluate the universality of the three-stage information flow, we extend our analysis to Qwen3-0.6B, Qwen2.5-32B, Llama-3.2-3B, Llama-3.1-8B, Mistral-7B-v0.1 (Mistral), and Gemma-3-4b-it (Gemma). While all models transition through semantic grounding and localization, the heatmaps in Figure 18 reveal two distinct mechanistic paradigms in how the internal representations utilize the input sequence. The Qwen models (Figures 18a and 18b) align closely with the mechanism described in the main text. We summarize the layer-wise distribution of the three-stage information flow in Table 20 and analyze them in detail.

Table 20: Approximate layer-wise distribution of the three-stage information flow across different LLMs.

Model	Stage I	Stage II	Stage III
Qwen3-0.6B	Layers 1-12	Layers 13-18	Layers 19-28
Qwen2.5-32B	Layers 1-26	Layers 27-46	Layers 47-64
Llama-3.2-3B	Layers 1-11	Layers 12-14	Layers 15-28
Llama-3.1-8B	Layers 1-12	Layers 13-15	Layers 16-32



(a) Qwen3-0.6B (b) Qwen2.5-32B (c) Llama-3.2-3B (d) Llama-3.1-8B

Figure 19: Analysis of semantic binding mechanisms: the top-1 accuracy evaluated with cosine similarity between query constraints and all table headers across layers of LLMs.

- **Qwen3-0.6B (28 Layers):** Stage I (Layers 1-12) focuses on query constraints. Stage II (Layers 13-18) shows a distinct causal bridge where interventions on the table tokens ($\mathcal{I}_t$) significantly affect the output. Stage III (Layers 19-28) shifts the mass to the final generation position.
- **Qwen2.5-32B (64 Layers):** The Stage II localization (Layers 27-46) involves nearly the entire vertical span of the table sequence, confirming that the model actively reads the serialized coordinates from the table tokens to resolve the target value.

In contrast, the Llama series (Figures 18c and 18d) displays a sparse causal pattern. Notably, the table token positions remain almost entirely white (effect score ≈ 0), suggesting a more direct mapping:

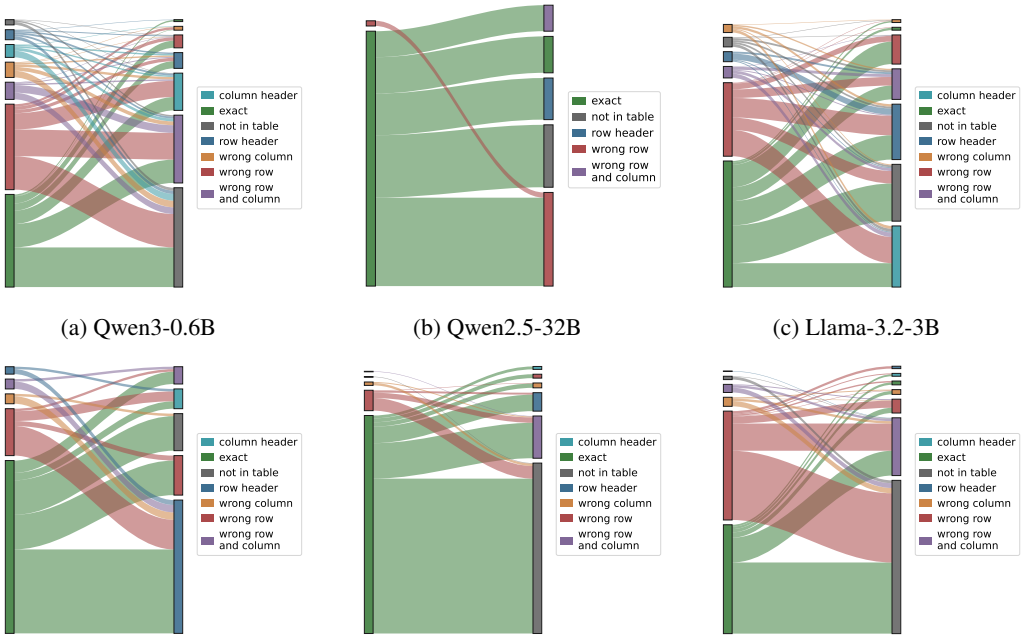
1. **Stage I:** For Llama-3.2-3B (Layers 1-10) and Llama-3.1-8B (Layers 1-11), significant patching effects are concentrated strictly on the query constraints ($\mathcal{I}_q$). This indicates that the initial semantic binding is purely focused on identifying the constraints within the question.
2. **Stage II:** Between the early and late layers, there is a narrow transitional window (e.g., Layers 12-14 in Llama-3.2) where the causal mass begins to shift. However, unlike Qwen, this transition bypasses the table tokens in the residual stream.
3. **Stage III:** High effect scores (approaching 1.0) appear abruptly at the position of the final token (e.g., Layer 15+ for 3B, Layer 16+ for 8B).

The absence of causal effects in the table region suggests that Llama models do not necessarily move information through the table tokens' hidden states. Instead, the late-layer attention heads at the final token position likely perform long-range retrieval directly from the table's KV-cache based on the semantic signals processed in the early-layer query tokens. The weaker table-token mediation and stronger output-position dependence in Llama are consistent with less reliance on table-token residual states than in Qwen, potentially involving more direct retrieval from the table KV-cache. The interventions do not isolate a unique storage or retrieval implementation. Cross-model generalization therefore concerns a shared functional sequence with model-specific routing.

H.2.2 Stage I: Semantic Binding

To validate the universality of the semantic binding stage, we extend our analysis to the broader model suite. As illustrated in Figures 19 and 20, while the specific layer boundaries vary according to total depth, the fundamental mechanism of binding query constraints ($\mathcal{I}_q$) to table headers ($\mathcal{I}_t$) remains consistent across different models and scales. The layer-wise cosine similarity heatmaps (Figure 19) demonstrate that all models achieve near-perfect retrieval accuracy (1.0) for both row and column headers in their respective early layers.

- **Qwen Series:** For Qwen3-0.6B and Qwen2.5-32B, the alignment converges rapidly. In Qwen3-0.6B, binding is stabilized by Layer 10, while in the larger Qwen2.5-32B, this process is completed within the first 15 layers. Consistent with the main text, column alignment consistently reaches a high-similarity plateau earlier than row alignment, indicating a lower inherent difficulty in binding column-wise semantic information.
- **Llama Series:** Llama-3.2-3B and Llama-3.1-8B exhibit a similar convergent pattern within Layers 1-12. However, a notable observation in Figures 19c and 19d is the slight noise or



(a) Qwen3-0.6B (b) Qwen2.5-32B (c) Llama-3.2-3B

(d) Llama-3.1-8B (e) Mistral-7B-v0.1 (f) Gemma-3-4b-it

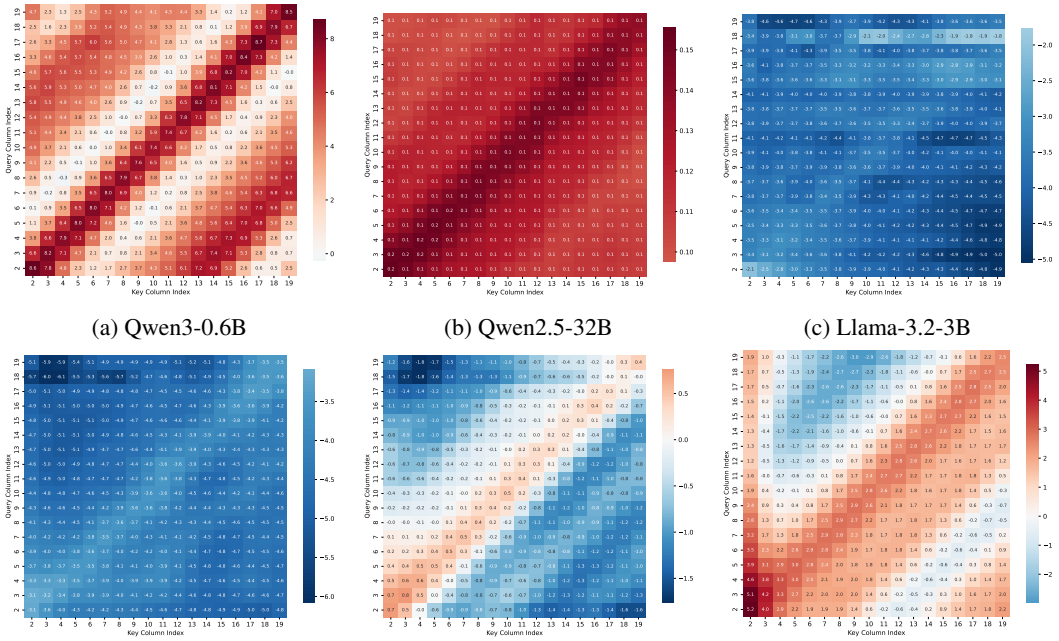
Figure 20: Analysis of semantic binding mechanisms: the impact of ablating row alignment heads on various LLMs.

delayed convergence in row retrieval compared to the Qwen series. This suggests that Llama models may require more transformation steps in the residual stream to align row semantics before moving to the localization stage.

The Sankey diagrams (Figure 20) visualize the performance collapse following the zero-ablation of identified *row alignment heads*.

- **Qwen3-0.6B & Qwen2.5-32B:** These models show a highly concentrated causal dependency. In Qwen2.5-32B, ablating these heads leads to a near-total elimination of `exact` matches, with the majority of errors flowing into `not in table` and `wrong row`. This confirms that the identified heads are the primary functional units responsible for grounding the query row constraint to the table's structure.
- **Llama-3.2-3B & Llama-3.1-8B:** The Llama series exhibits a more complex error distribution. While `exact` performance drops significantly, a non-negligible portion of the flow transitions to `row header` and `column header` errors. These specific error types represent cases where the model directly outputs the query constraints rather than the cell value. Combined with the paradigm observed in Figure 18, this suggests that when row binding heads are ablated, Llama models default to a copy-mechanism from the query tokens in the KV-cache, failing to initiate the retrieval process from the table.
- **Mistral and Gemma:** We further validate Stage I on Mistral and Gemma in Figure 20e and 20f. After ablating the identified *row-alignment heads*, 78.4% of originally correct predictions on Mistral and 70.2% on Gemma become incorrect. The resulting outputs are dominated by answers not present in the table or by answers localized to an incorrect row, which is fully consistent with the role of Stage I established in the main text. These results show that the causal necessity of semantic row binding extends to additional model families beyond Qwen and Llama.

In summary, across all tested LLMs, Stage I is characterized by a rapid geometric alignment in the early residual stream and is causally driven by a sparse set of specialized attention heads. The experimental results for these models align with the layer-wise distributions proposed in Table 20, confirming that semantic binding is a prerequisite for subsequent coordinate localization.



(a) Qwen3-0.6B (b) Qwen2.5-32B (c) Llama-3.2-3B

(d) Llama-3.1-8B (e) Mistral-7B-v0.1 (f) Gemma-3-4b-it

Figure 21: Visualization of the geometric interaction score $S_{j,l}$ in coordinate-encoding heads of various LLMs. The heatmaps illustrate the dot product between column-averaged query vectors ($\bar{q}_j$) and key vectors ($\bar{k}_l$) after injecting RoPE corresponding to their column indices.

Table 21: Pearson correlation of the diagonal pattern in the geometric interaction score $S_{j,l}$ for coordinate-encoding heads across models.

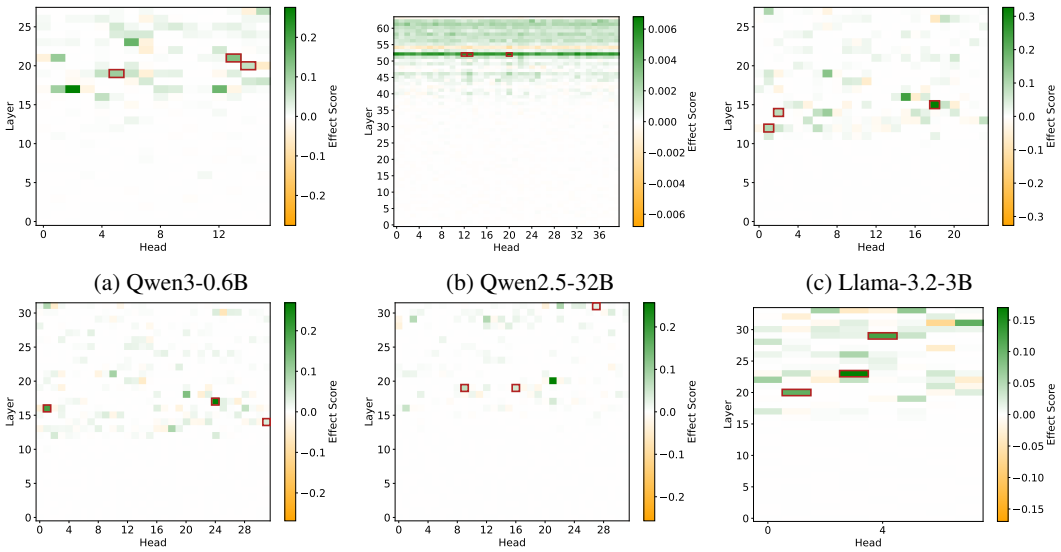
Model	Atomic	Multi
Qwen3-4B	0.997	0.908
Qwen3-0.6B	1.000	0.535
Qwen2.5-32B	0.999	0.993
Llama-3.2-3B	0.908	0.635
Llama-3.1-8B	0.768	0.454
Mistral-7B-v0.1	0.999	0.996
Gemma-3-4b-it	0.998	0.999

H.2.3 Stage II: Coordinate Localization

Following §3.3, we apply the Virtual-RoPE coordinate diagnostic to additional models by assigning logical column positions to column-averaged Q/K states (Figure 21). These are counterfactual geometric scores, not native attention logits. For Qwen3-0.6B and Qwen2.5-32B, the Virtual-RoPE diagnostic reveals a highly distinct diagonal pattern (Figures 21a and 21b), consistent with the observations for Qwen3-4B in the main text.

- **Qwen3-0.6B:** The prominent diagonal $j = l$ indicates that even at a smaller scale, the model identifies specialized heads in middle layers (Layers 13-18) that prioritize tokens based on their relative column indices.
- **Qwen2.5-32B:** The larger model exhibits a cleaner diagonal with minimal off-diagonal noise, indicating clearer coordinate structure under this diagnostic.

These results confirm that the Qwen series consistently transitions from semantic binding to a table-mediated localization stage, where the residual stream explicitly utilizes the table's internal geometry to resolve target coordinates [18].



(a) Qwen3-0.6B (b) Qwen2.5-32B (c) Llama-3.2-3B

(d) Llama-3.1-8B (e) Mistral-7B-v0.1 (f) Gemma-3-4b-it

Figure 22: Effect Scores of attention heads at the final token position on various LLMs. The red boxes highlight three critical heads that exhibit positive Effect Scores and act as the primary information movers.

The Llama series (Llama-3.2-3B and Llama-3.1-8B) presents a more nuanced geometric interaction. As shown in Figures 21c and 21d, while a diagonal pattern exists, it is characterized by significantly *negative* interaction scores (darker blue regions). The negative diagonal is a property of the Virtual-RoPE diagnostic, not a measurement of native attention logits. Its sign therefore does not establish inhibitory attention, masking of non-target columns, or a specific KV-cache retrieval mechanism. We interpret the column-aligned pattern as geometric evidence and analyze model-specific routing separately in Appendix H.2.

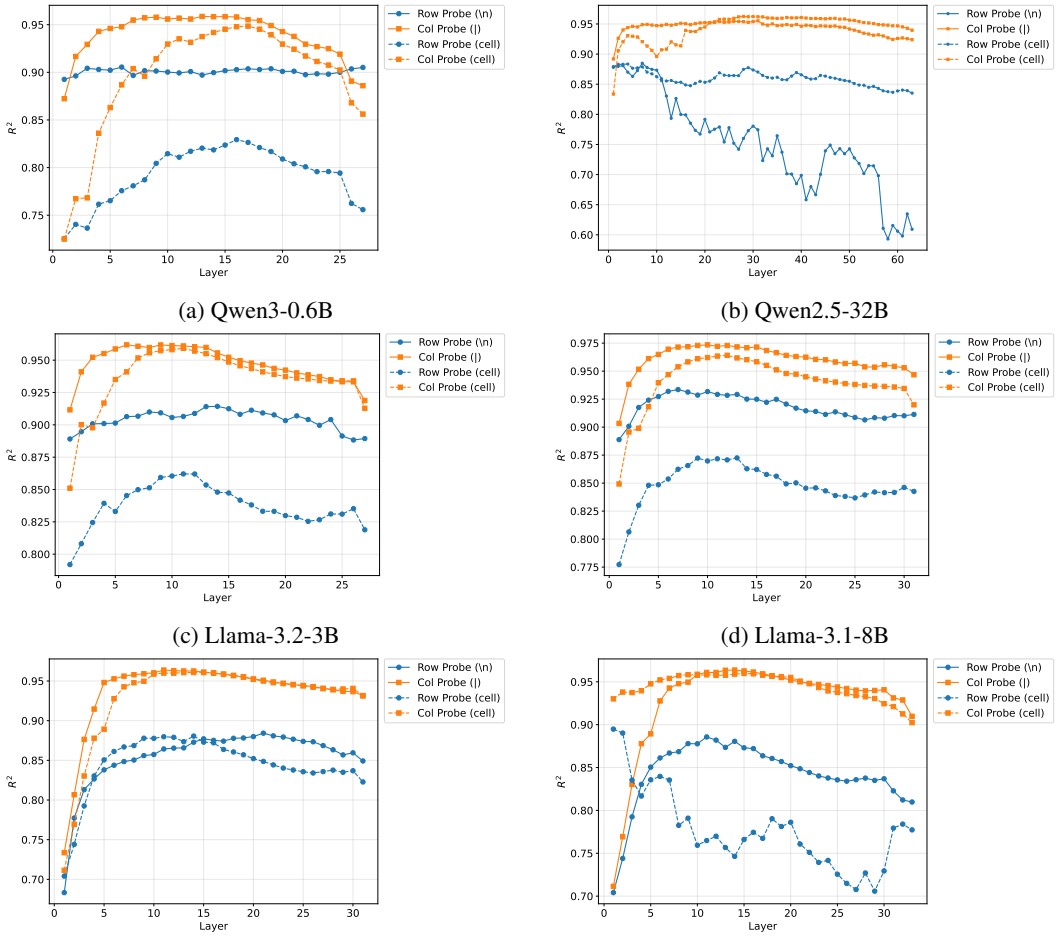
Additional evidence for Stage II is shown in Figure 21e and 21f. We identify *coordinate-encoding heads* in both Mistral and Gemma, and the resulting geometric interaction scores exhibit a clear diagonal structure. The corresponding Pearson correlations are 0.999 for Mistral and 0.998 for Gemma, supporting coordinate-structured geometry under the diagnostic across the evaluated architectures.

In conclusion, despite the divergence in the sign of the interaction scores, all tested models exhibit a clear reliance on RoPE-based structural signals during Stage II. Quantitatively, the diagonal pattern is strong across models, with Pearson correlations of 0.997 on Qwen3-4B, 1.000 on Qwen3-0.6B, 0.999 on Qwen2.5-32B, 0.908 on Llama-3.2-3B, and 0.768 on Llama-3.1-8B (Table 21).

H.2.4 Stage III: Information Propagation

To verify the generalizability of the information propagation mechanism, we replicate the final-token activation patching experiment on other models. Following the setup in §3.4, we intervene on the query constraints to target a counterfactual cell and compute the Effect Scores for all attention heads at the final token position. The results are visualized in Figure 22. For Qwen3-0.6B (Figure 22a), Llama-3.2-3B (Figure 22c), and Llama-3.1-8B (Figure 22d), we observe a clear concentration of high Effect Scores in the late layers (Stage III), consistent with the findings for Qwen3-4B.

- **Qwen Series:** In Qwen3-0.6B and Qwen2.5-32B, causal mass emerges prominently starting from Layer 19 and Layer 47, respectively. The identified *information mover heads* (highlighted in red) exhibit high precision in reading the resolved value from the localized table coordinates.
- **Llama Series:** In Llama-3.2-3B and Llama-3.1-8B, the propagation stage is also localized, appearing abruptly in the late layers. This supports our previous hypothesis: since Llama bypasses intermediate table-mediated storage, its late-layer heads must perform a direct, high-impact retrieval from the KV-cache to move information to the final residual stream.



(a) Qwen3-0.6B

(b) Qwen2.5-32B

(c) Llama-3.2-3B

(d) Llama-3.1-8B

(e) Mistral-7B-v0.1

(f) Gemma-3-4b-it

Figure 23: R^2 of column probes and row probes trained for cells and delimiters across layers on various LLMs.

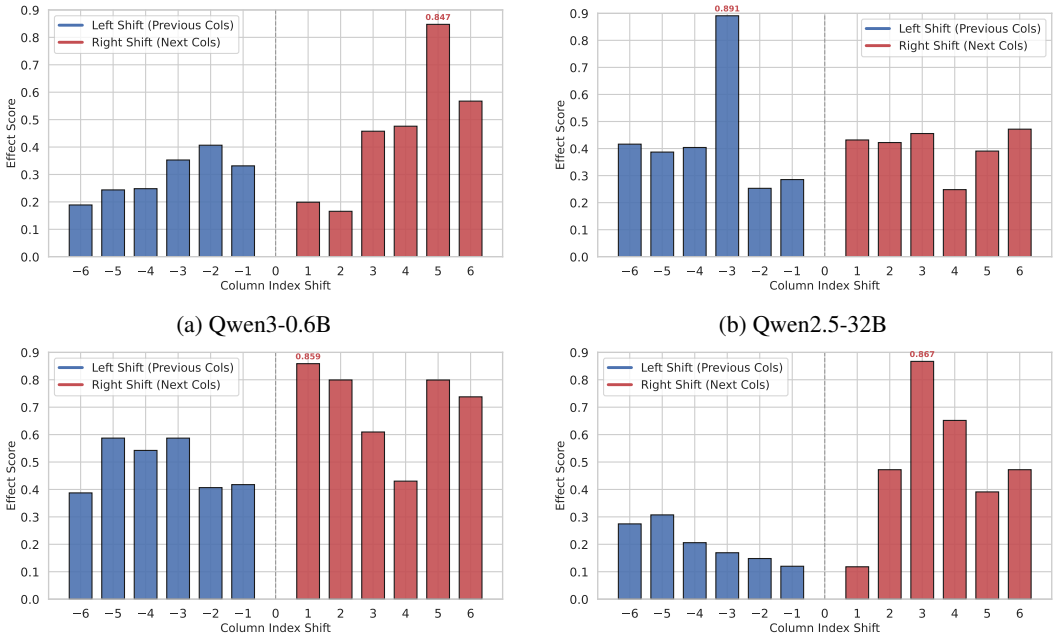
- **Mistral and Gemma:** In both models, we again identify late-layer *information mover heads* whose average attention to the answer token exceeds 30%. This shows that, once the target cell has been localized, the final decoding step is consistently implemented by a sparse set of late-layer heads that move the resolved value to the output position.

Across all models, the emergence of positive Effect Scores at the final token position serves as a definitive marker for Stage III. The temporal alignment of these heads with the final layers (as specified in Table 20) confirms that the three-stage pipeline is a robust mechanistic property of transformer-based table understanding.

H.3 Implicit Coordinate System via Delimiter Counting

In this subsection, we extend the investigation of the implicit coordinate system to the broader model suite to evaluate whether the delimiter-driven counting mechanism is a universal feature of transformer-based table understanding.

We visualize the R^2 of linear probes for the supplementary models in Figure 5. Similar to the findings for Qwen3-4B, all models exhibit a distinct performance gap between delimiter tokens and cell tokens in the early-to-middle layers. All models achieve near-perfect column R^2 (> 0.95) on delimiters extremely early (by Layer 5-10). The cell probes lag behind, converging to high accuracy only in the middle layers. This reinforces the hypothesis that delimiters act as the primary anchors for structural states, which are subsequently synchronized to cell hidden states. Also, the column probes of all



(a) Qwen3-0.6B

(b) Qwen2.5-32B

(c) Llama-3.2-3B

(d) Llama-3.1-8B

Figure 24: Effect Scores resulting from injecting the unit shift vector scaled by integer factors k on various LLMs. The positive integers denote rightward shifts and negative integers denote leftward shifts.

models exhibit consistently higher performance than row probes, confirming that the models utilize column indices as primary location mechanism, with treating row positions as secondary features derived from row boundaries. Using trained probes from various LLMs that exhibit high R^2 scores, we successfully predict column and row indices. Our results reveal distinct sawtooth patterns and stepwise functions, which are consistent with the findings presented in §4.1.

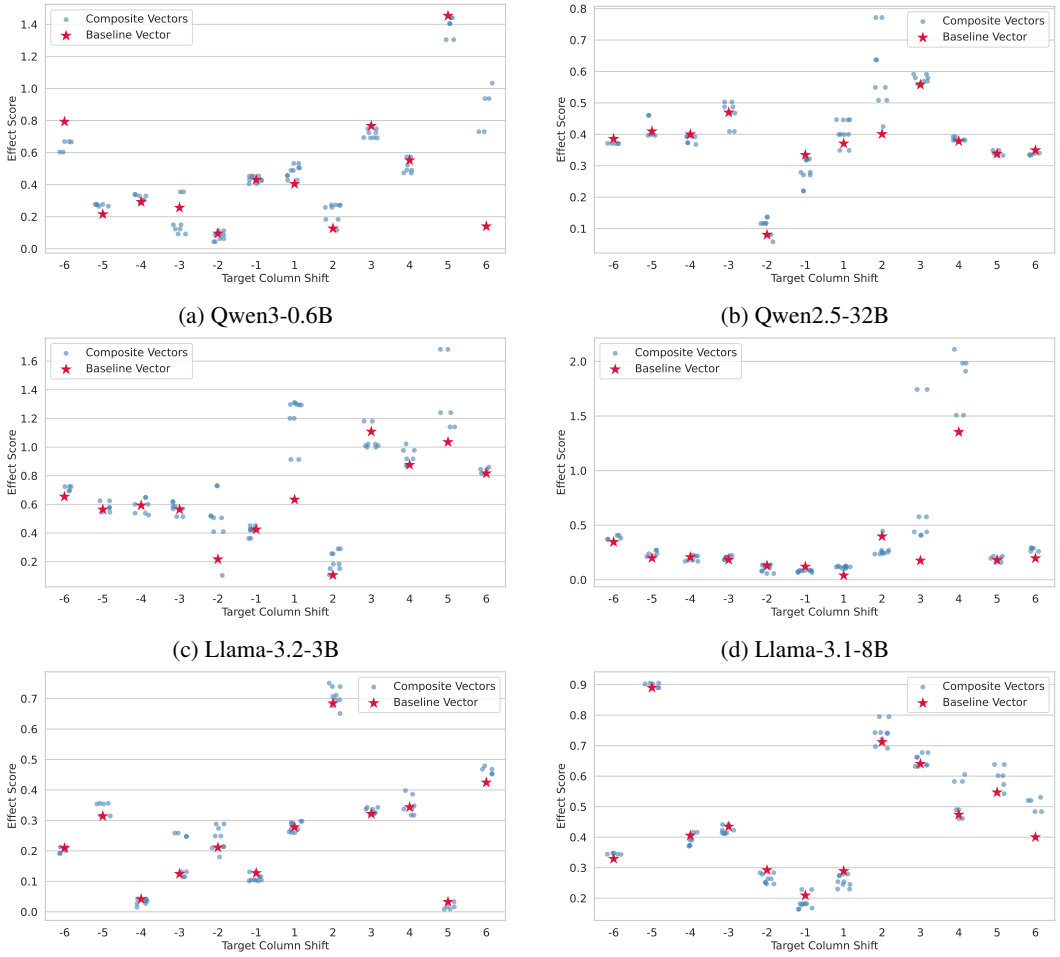
H.4 The Linear Geometry of Tabular Indices

To assess the cross-model consistency of the linear geometry hypothesis, we replicate the vector steering and compositionality experiments (§5) on more models. The goal is to determine if the representation of column indices as a linear subspace is a universal property of transformer-based table understanding.

H.4.1 Steering with Unit Shift Vectors

Following the setup in §5.1, we inject the scaled unit shift vector $\vec{v}_{unit}$ into the residual stream of the column header tokens. Figure 24 illustrates the resulting Effect Scores across varying integer factors k.

- **Qwen Series:** Both Qwen3-0.6B and Qwen2.5-32B exhibit a robust linear response to the steering vector. In Qwen3-0.6B (Figure 24a), the Effect Score peaks at $k = 5$ with a value of 0.847, demonstrating that even at a sub-billion parameter scale, the model maintains a structured linear progression for column navigation. Qwen2.5-32B (Figure 24b) displays a similar trend, particularly for negative shifts, with $k = -3$ achieving a peak score of 0.891.
- **Llama Series:** The Llama models also demonstrate successful steering, with Llama-3.2-3B (Figure 24c) and Llama-3.1-8B (Figure 24d) reaching peak Effect Scores of 0.859 and 0.867 respectively. This success confirms that despite the sparser causal mass observed in earlier patching experiments, the Llama residual stream indeed maintains a functional coordinate system. The injected shift vector effectively alters the latent geometric pointers, which



(a) Qwen3-0.6B

(b) Qwen2.5-32B

(c) Llama-3.2-3B

(d) Llama-3.1-8B

(e) Mistral-7B-v0.1

(f) Gemma-3-4b-it

Figure 25: Comparison of Effect Scores between Baseline Vectors (red stars) and Composite Vectors (blue dots) on various LLMs. The Baseline Vector represents the direct extraction of the shift vector for target offset k. The Composite Vectors are formed by summing two distinct vectors $\vec{v}_a + \vec{v}_b$ such that $a + b = k$.

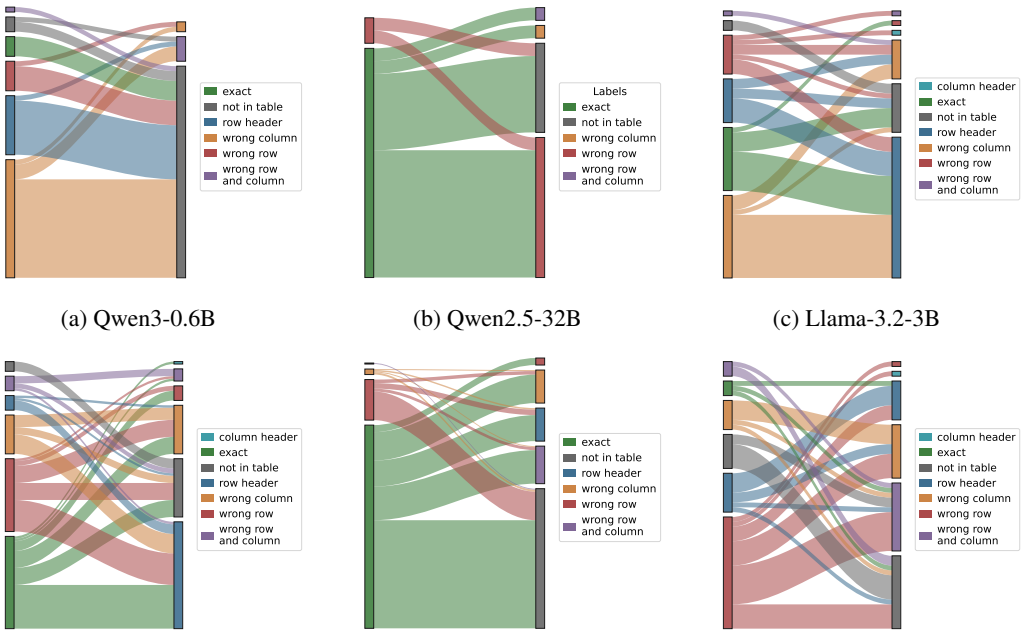
subsequently guide the late-layer attention heads to retrieve values from the steered column indices in the KV-cache.

H.4.2 Compositionality of Column Vectors

We evaluate the additivity of the shift vectors by comparing composite vectors $\vec{v}_{comp} = \vec{v}_a + \vec{v}_b$ against the baseline vectors $\vec{v}_k$ ($a + b = k$). The results are presented in Figure 25.

The visual alignment between the red baseline stars and the blue composite dots across all four models supports approximate additivity of the tabular index representation over the tested offsets.

- In the **Qwen series** (Figures 25a and 25b), the composite dots tightly cluster around the baseline stars. For Qwen2.5-32B, at target offsets like $k = 2$ and $k = 3$, the composite vectors yield performance comparable to the directly extracted baseline, supporting approximate additivity over these offsets.
- In the **Llama series** (Figures 25c and 25d), we observe high-fidelity compositionality. Notably, in Llama-3.1-8B, the composite vectors at $k = 3$ and $k = 4$ exhibit a reinforcement effect, where the steering efficacy (Effect Score > 1.5) remains high across various (a, b)



(a) Qwen3-0.6B (b) Qwen2.5-32B (c) Llama-3.2-3B

(d) Llama-3.1-8B (e) Mistral-7B-v0.1 (f) Gemma-3-4b-it

Figure 26: Mechanistic analysis on the multi-cell location task: the impact of ablating the row alignment heads of various LLMs identified in Stage I on multi-row queries.

combinations. This suggests that the directionality of the column-shift vector is highly stable across different positions in the latent space.

- In the **Mistral and Gemma** models, the close alignment between the red baseline stars and the blue composite dots again indicates a high degree of linearity in the representation of tabular indices. This result provides additional support for the hypothesis that column shifts are encoded as approximately additive vector translations in the latent space.

Across the evaluated models, column shifts support approximately additive interventions over the tested offsets. This does not establish global linearity or identify which table dimensions were absent from pretraining; wider-table and boundary results appear in Appendix E.7.

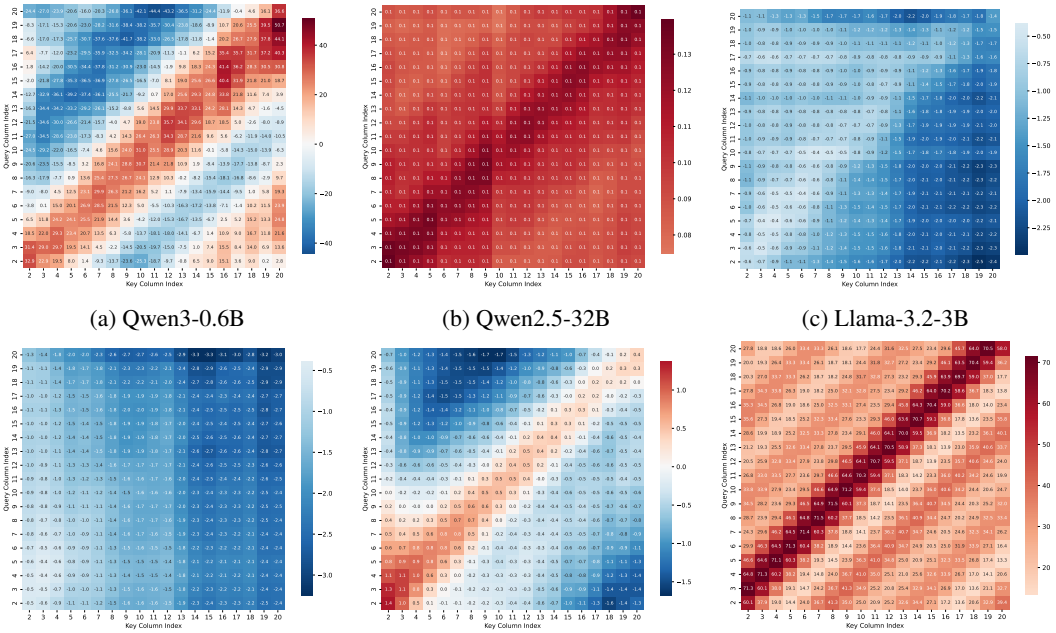
H.5 Generalization to Multi-Cell Location

To evaluate the universality and robustness of the parallel processing pipeline, we extend the mechanistic analysis of multi-cell retrieval to other models. Following the experimental setup in §G.5, we examine whether the specialized functional heads identified in the atomic task maintain their roles when processing multiple constraints concurrently.

H.5.1 Parallel Semantic Binding across LLMs

As illustrated in Figure 26, zero-ablating the top-20 *row alignment heads* triggers a consistent performance collapse across all models, validating that parallel semantic binding is a fundamental mechanism in transformer-based table understanding.

- **Qwen Series:** For Qwen3-0.6B and Qwen2.5-32B (Figures 26a and 26b), we observe a catastrophic drop in `exact` matches. For instance, Qwen2.5-32B shows a significant transition toward `not in table` and `wrong row` errors. This confirms that these heads are essential for anchoring the multiple query row constraints to their respective table headers. The similar transitions suggest that the row alignment operator naturally scales to multi-target scenarios via the attention mechanism's inherent parallelism.



(a) Qwen3-0.6B (b) Qwen2.5-32B (c) Llama-3.2-3B

(d) Llama-3.1-8B (e) Mistral-7B-v0.1 (f) Gemma-3-4b-it

Figure 27: Mechanistic analysis on the multi-cell location task: the Query-Key interaction scores for the top-20 coordinate-encoding heads of LLMs from Stage II on multi-column queries.

- **Llama Series:** In Llama-3.2-3B and Llama-3.1-8B (Figures 26c and 26d), the ablation results in a substantial increase in `row header` errors. The prevalence of `row header` errors further supports the hypothesis that without these alignment heads, the Llama series fails to bridge the gap between the query and the table content, defaulting to a literal copy of the query tokens.
- **Mistral and Gemma**: Reusing the same *row-alignment heads* and ablating them causes 69.5% of exact predictions on Mistral and 66.7% on Gemma to become errors, typically shifting to an incorrect row, a row header, or an incorrect cell outside the table. This confirms that the same semantic binding mechanism is reused under more complex multi-row queries.

H.5.2 Multi-Threaded Coordinate Localization across LLMs

Figure 27 presents the interaction scores $S_{j,l}$ for the *coordinate-encoding heads* under multi-column queries. The results demonstrate that the structural navigation mechanism generalizes effectively to parallel index resolution.

- **Qwen Series:** Figures 27a and 27b exhibit clear, multi-modal diagonal patterns. This indicates coordinate-aligned geometry for multiple queried columns under the Virtual-RoPE diagnostic.
- **Llama Series:** Consistent with the findings in the atomic task, Llama-3.2-3B and Llama-3.1-8B (Figures 27c and 27d) exhibit a negative diagonal pattern. The dark blue regions aligned with the queried column indices suggest that Llama utilizes these heads as inhibitory filters to mask non-target columns. In the multi-cell setting, this mechanism functions by suppressing all columns except those specified in the query, effectively isolating multiple KV-cache signals for parallel retrieval. While the sign of the interaction score differs from Qwen, the preservation of the structural alignment across multiple column indices confirms that the geometric localization logic is preserved.
- **Mistral and Gemma**: The same generalization also holds for Stage II. Reusing the *coordinate-encoding heads* identified in the atomic task, we again observe a clear diagonal pattern in the geometric interaction score for multi-column queries. The corresponding Pearson correlations are 0.996 on Mistral and 0.999 on Gemma, showing that the geomet-

ric localization mechanism transfers cleanly from atomic cell location to more complex multi-cell settings.

Quantitatively, the diagonal structure remains visible across the original model suite, with Pearson correlations of 0.908 for Qwen3-4B, 0.535 for Qwen3-0.6B, 0.993 for Qwen2.5-32B, 0.635 for Llama-3.2-3B, and 0.454 for Llama-3.1-8B (Table 21). In conclusion, the consistency of these patterns across varied models and scales reinforces the three-stage information flow. Regardless of model size or specific attention behavior (excitatory vs. inhibitory), LLMs resolve complex multi-cell queries by reusing the same specialized circuit identified in simpler tasks, executing semantic and geometric operations in a single, parallelized forward pass.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitation of this paper in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided the information for main experimental results in §2 and Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Considering that time is tight, we will release source codes as soon as this paper is accepted.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We have discussed the test details in §2 and Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report means and sample standard deviations across ten independently generated datasets in Appendix D.3. Deterministic decoding does not eliminate variability from random table and query generation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We discuss required computer resources in §E.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: All our research conform with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the societal impacts of this paper in Appendix B.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our usage of data and models does not have a high risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All our usage of creators or original owners of assets is properly credited and are the license and terms of use explicitly mentioned and properly respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not introduce new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We do not adapt crowdsourcing experiments and research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We have discussed the usage of LLMs in §2.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.