STAR-Table: Spatial Textual Anchoring and Reconstruction for hierarchical Table reasoning via OCR

Anonymous ACL submission

Abstract

Hierarchical tables are ubiquitous in real-world scenarios, yet their complex data structure poses severe challenges to the reasoning capabilities of Large Language Models (LLMs). The core difficulty lies in precisely aligning complex topological structures with multi-level semantic dependencies, where alignment failures inevitably cause cross-level aggregation errors and hallucinations. To address this, we propose **STAR-Table**, a framework synergizing visual perception with symbolic reasoning. Unlike traditional text-only approaches, we leverage specialized OCR models to extract pixel-level geometric coordinates, establishing explicit Spatial-Textual Anchors. By formulating these anchored spatial priors as explicit structural hints, our framework guides the LLM to reconstruct the table's hierarchical topology, then translating the unstructured visual layout into a deterministic, flattened database. Finally, we design an adaptive QA module fuses VLM semantic summaries with multi-step SQL execution to conduct robust table-based Reasoning tasks. Experimental results on the HiTab benchmark demonstrate that **STAR-Table** achieves an accuracy of 81.47%, significantly outperforming specialized fine-tuned models and advanced prompting paradigms.

1 Introduction

As carriers of structured data, tables play an indispensable role in financial statements, statistical reports and scholarly documents (Cafarella et al., 2008). A table is regarded as hierarchical if its header exhibits a multi-level structure (Wang et al., 2021a). In real-world scenarios, flat tables have a limited capacity to store multi-dimensional data; consequently, hierarchical tables are significantly more prevalent (Cheng et al., 2022). Unlike their flat counterparts, these tables implement multi-dimensional organization through nested headers and complex cell merging. While

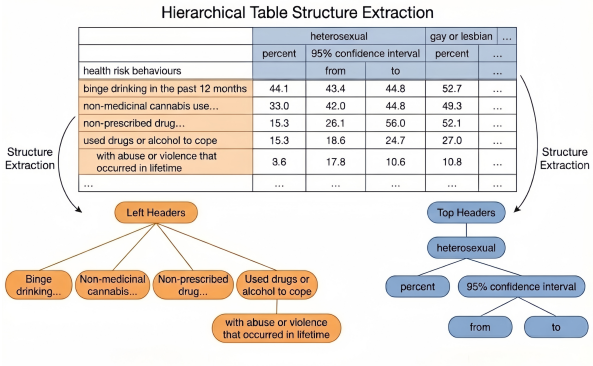


Figure 1: Structural extraction of a hierarchical table. The visual layout is parsed into topological trees, representing the top ($\mathcal{H}_{top}$) and left ($\mathcal{H}_{left}$) header hierarchies. Extracting this logical schema is fundamental for accurate Table QA.

this structure facilitates high-density information representation with enhanced logical coherence, it poses significant challenges for Large Language Models (LLMs), which require further comprehension in intricate hierarchical layout and semantic dependencies.

Despite the remarkable success of recent advancements on flat tables for reasoning tasks like Table Question Answering (Pasupat and Liang, 2015) and Fact Verification (Chen et al., 2019), extending these capabilities to hierarchical tables presents a significantly greater challenge. To successfully perform deeper logical processing on such complex layouts, models must systematically reconstruct these hierarchical headers to bridge the gap between visual arrangement and symbolic reasoning. As illustrated in Figure 1, the logical identity of each data cell is determined by its specific path within the two hierarchical trees, requiring the model to align visual spans with their corresponding semantic ancestors. The difficulty of this task is twofold. On one hand, the inherent two-dimensional spatial characteristics of tables establish complex topological connections that are

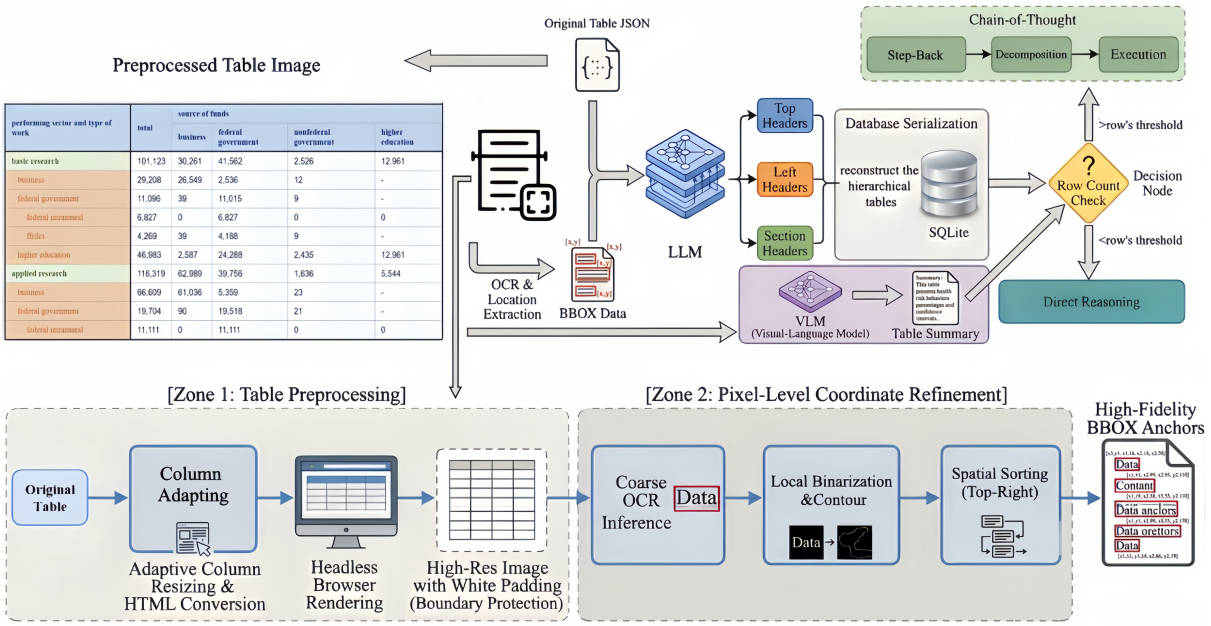


Figure 2: **The overall architecture of our proposed framework.**Extracted visual layouts and original JSON data are fused via LLMs to parse top, left, and section headers. These are serialized into a deterministic SQLite database, using recursive header flattening to resolve nesting ambiguity. Concurrently, a VLM generates a semantic summary. Finally, an adaptive reasoning module fuses both outputs, dynamically routing queries by row count: complex tables trigger a CoT pipeline, while smaller ones use direct reasoning.

exceptionally difficult to capture through standard text. On the other hand, unlike flat key-value pairs, the semantics of data cells in hierarchical tables are governed by multi-level header paths. Consequently, complex queries often necessitate cross-level reasoning, where models must traverse the header tree to perform aggregation or comparison across distinct hierarchical levels. Failure to rigorously comprehend these structural constraints and cross-cell dependencies directly leads to reasoning fallacies, such as aggregation errors or semantic hallucinations.

To address these challenges, existing solutions have explored various paradigms. Prevailing approaches (Liu et al.; Herzig et al., 2020) attempt to model hierarchical structures by serializing two-dimensional layouts into one-dimensional autoregressive sequences. However, this linearization inevitably disrupts the inherent topological connections, making it difficult to reconstruct the recursive parent-child dependencies essential for defining a cell's semantic scope. Humans can intuitively grasp tabular hierarchies through visual observation, indicating that visual representations offer a comprehensive global layout perspective. While recent studies such as MatCha (Liu et al., 2023) have incorporated Vision-Language Models (VLMs) to simulate this process, these general-purpose models still lack the granularity required to capture fine-grained spatial details. This limitation motivates our core intuition: modern specialized Optical Character Recognition (OCR) models, such as PaddleOCR (Cui et al., 2025), DeepSeek-OCR (Wei et al., 2025), and MinerU (Wang et al., 2024a), provide superior fine-grained spatial perception. By leveraging OCR, we can model tables across two dimensions: a geometric dimension for precise hierarchical parsing, and a semantic dimension that integrates VLMs and large language models (LLMs) for robust logical reasoning. This synergy enables a robust reconstruction of hierarchical structures, providing a solid foundation for downstream question answering (QA) tasks.

To realize this synergy, we propose STAR-Table, a geometry-aware pipeline that integrates fine-grained visual perception with deterministic symbolic execution. For hierarchical table-based reasoning that integrates visual perception with textual processing, with its overall architecture depicted in Figure 2. Specifically, our framework extracts dense spatial coordinates from the OCR engine to serve as explicit structural priors. By mapping these physical coordinates to textual elements, the LLM can precisely resolve spanning cells and

nested hierarchies, translating 2D visual layouts into an executable symbolic schema. Specifically, we first transform tables from various original formats into standard images, then employ an OCR model to extract textual content alongside its corresponding spatial coordinates. Subsequently, we leverage an LLM, conditioned on the raw visual information, to comprehend the layout and reconstruct the hierarchical structure of the table. Based on the parsed structure, the raw table content is transformed into a structured database file. Finally, we design a tailored LLM prompting strategy to execute the table QA task.

Our main contributions are summarized as follows: **1. Vision-Symbolic Synergetic Paradigm**: We propose a novel geometry-aware pipeline that synergizes visual perception with symbolic reasoning to address the challenge of table-based reasoning over complex hierarchical tables. **2. Fine-Grained Visual Perception**: We introduce an optimized OCR module with pixel-level coordinate refinement. Distinct from generic VLM-based approaches that often overlook fine-grained spatial details, our method leverages precise physical coordinates and raw visual cues to accurately capture the topology of nested headers and spanned cells. **3. Deterministic Structure Reconstruction**: We devise a geometry-guided prompting strategy to transform unstructured visual layouts into executable databases. By implementing schema flattening and dynamic sub-table instantiation, we effectively resolve the information loss problem inherent in linearization-based methods. **4. Robust Reasoning and Performance**: Experimental results on the HiTab benchmark demonstrate that our method achieves superior accuracy compared to existing baselines. Beyond these substantial performance gains, our database-decoupled reasoning paradigm demonstrates remarkable robustness against highly complex tabular structures and adversarial perturbations.

2 Related Work

2.1 Hierarchical Tables

Prior table question answering (Table QA) datasets primarily featured simple table structures. Prominent examples include WikiTableQuestions (Pasupat and Liang, 2015) and Spider (Yu et al., 2018), which predominantly consist of flat tables.To bridge this gap, Parikh (Parikh et al., 2020) introduced ToTTo, a dataset containing a subset of complex tables with hierarchical content.Subsequently, in the financial domain, FinQA (Suster et al., 2021) and TAT-QA (Zhu et al., 2021) introduced hybrid data QA tasks, requiring models to possess numerical reasoning capabilities across both textual and tabular data. HiTab (Cheng et al., 2022) dedicates to hierarchical table QA and generation; and it explicitly highlights the challenges posed by non-standard structures through the introduction of Hierarchical Indexing. Furthermore, MultiHiertt (Zhao et al., 2022) proposed a task involving Question Answering over multiple hierarchical tables, demanding that models handle cross-table entity linking and numerical aggregation. The goal of our work is to transform these hierarchical tables into a format that facilitates subsequent reasoning by LLMs while preserving the complete information of the original tables to the greatest extent possible.

2.2 Prompting Strategies and Question Decomposition for LLM-based Table QA

Leveraging Chain-of-Thought (CoT) to elicit reasoning in LLMs has become one of the dominant methods for table-based reasoning. For instance, Chain-of-Table (Wang et al.) proposes simulating the reasoning process by dynamically generating intermediate tables, while Dater (Ye et al., 2023) employs row and column decomposition methods to address the challenges of reasoning over large-scale tables. Furthermore, Least-to-Most (Zhou et al.) and DecomP (Khot et al.) introduced mechanisms that encourage LLMs to generate and solve sub-questions during the reasoning process. These approaches have demonstrated significant improvements in handling questions that require multi-step reasoning and complex calculations.

Researchers have also explored incorporating external tools to address tasks requiring high precision, such as numerical computation. Program of Thoughts (PoT) (Chen et al.) and PAL (Gao et al., 2023) proposed generating Python programs and executing them via an interpreter to solve specific reasoning problems. Similarly, Rajkumar (Rajkumar et al., 2022) introduced LLM-based Text-to-SQL methods and applied them to Table QA tasks. To further extend capabilities, Binder (Cheng et al.) generates SQL or Python programs and extends their capabilities by calling LLMs as APIs in the programs.

In this work, we maximize information preservation by converting raw table files into database

files through a series of operations. Simultaneously, we introduce a comprehensive QA workflow incorporating question step-back and sub-question decomposition to enhance the LLM's performance in solving complex problems.

3 Preliminaries

In this section, we formally define the hierarchical table structure, followed by the formulation of the table-based reasoning task.

Definition of Table Structure We formally define a geometry-aware atomic table unit as a tuple $\mathcal{T} = (\mathcal{C}, \mathcal{H}, t)$, where t denotes the table title. The Content set $\mathcal{C}$ comprises data entries, where each cell $c \in \mathcal{C}$ is encapsulated as a dual-modal entity comprising textual content v and a spatial coordinate b. Specifically, we define b as a 4-dimensional vector $b \in \mathbb{R}^4$ ($b = [x_{min}, y_{min}, x_{max}, y_{max}]$), representing the geometric span of the cell on the 2D image plane, where coordinates correspond to the normalized horizontal and vertical boundaries. The Header set $\mathcal{H}$ is the disjoint union of top ($\mathcal{H}_{top}$), left ($\mathcal{H}_{left}$), and section ($\mathcal{H}_{sec}$) headers; distinct from $\mathcal{C}$, headers semantically constrain data cells while sharing the same spatial attributes defined by b.

In hierarchical layouts, we model dependencies as directed trees: a relation $h_i \rightarrow h_j$ denotes that h_i is the parent (refinement) of h_j. To facilitate structural representation, we introduce virtual nodes denoted as $root_{top}$ and $root_{left}$, to serve as the logical roots for the top header hierarchy $\mathcal{H}_{top}$ and the left header hierarchy $\mathcal{H}_{left}$ respectively. A complex composite table is thus formalized as a collection $\mathbf{T} = \{T_1, \ldots, T_n\}$.

Definition of Task Given a target table T and a natural language question sequence $Q = \{w_1, w_2, \ldots, w_L\}$, the goal of Table QA is to generate the correct answer A. Our proposed framework can be formalized as a mapping function:

$$\mathcal{A} = F(T, Q, I)$$

where I represents the instructional context within the workflow, encompassing all prompt inputs used to guide the models.

4 Methodology

4.1 Overview

We propose an end-to-end framework designed to address hierarchical table-based reasoning tasks. As illustrated in Figure 2. First, we standardize diverse raw table formats into images and employ an optimized OCR module to extract textual primitives accompanied by precise physical coordinates. Second, guided by geometric spatial cues, we utilize LLMs to reconstruct hierarchical structures—including nested headers and table sections—and transform them into an executable relational database. Finally, we design a retrieval-augmented pipeline that synergizes query decomposition strategies with deterministic SQL execution, supplemented by VLM-generated table summaries, to achieve robust reasoning for complex multi-hop queries.

4.2 OCR-Driven Visual Perception

The core rationale for leveraging OCR is that a complex table's logical hierarchy is inherently encoded in its 2D layout. By extracting precise bounding boxes, OCR translates visual cues into explicit geometric constraints. This enables robust hierarchical tree reconstruction via coordinate overlap, bypassing the fragility of flattened text sequences. However, realizing this advantage in practice presents significant challenges.The efficacy of fine-grained OCR heavily relies on the visual clarity and structural consistency of the input, while raw tables have diverse formats which inevitably degrade coordinate localization. Driven by this observation, we design a unified preprocessing pipeline to standardize the input tables. First, to eliminate structural ambiguity, we standardize diverse source formats, explicitly anchor the first row as the table title, and enforce a strict top-down, left-to-right reading order. Furthermore, we recognize that dense textual layouts may cause OCR recognition errors. To mitigate this, we introduce a content-adaptive column resizing strategy that dynamically optimizes the horizontal spatial distribution, granting text sufficient visual separation. Finally, to ensure pristine geometric fidelity, these tables are converted into HTML and rendered by a headless browser into high-resolution images, yielding a perfectly standardized representation for downstream OCR parsing.

While modern OCR engines exhibit robust general text recognition, they often struggle with the extreme density and rigid geometry of large-scale hierarchical tables. So we optimized the engine initialization for tabular scenarios to maximize the detection recall of fine-grained textual elements within table layout. Considering text situated near

image boundaries may suffer from recognition degradation and marginal omissions, we applied fixed-width white padding around the input images; this preprocessing step mitigates recognition degradation at image boundaries. More critically, standard OCR bounding boxes tend to be excessively loose, encompassing background space that causes overlapping proposals in densely packed cells. To tackle the problem, we introduced a pixel-level coordinate refinement mechanism. For each coarse proposal generated by the OCR engine, we performed local binarization and contour detection within its Region of Interest (ROI) using OpenCV (Bradski, 2000). Finally, the refined coordinates are mapped back to the original image space. All text blocks are spatially sorted following a top-down, left-to-right sequence. The final output includes the bounding box coordinates for each object, which serve as high-fidelity anchors for the LLM in downstream reasoning stages.

4.3 Structure Extraction and Database Instantiation

Although the initial OCR phase yields textual primitives and spatial coordinates, raw OCR outputs are intrinsically insufficient to resolve the topology of hierarchical tables. To recover the logical hierarchy from unstructured visual layouts, we propose a geometry-guided prompting strategy to guide LLMs in precise structural inference.

First, to reconstruct the top header hierarchy $\mathcal{H}_{top}$ as a directed tree, we derive the maximum visual column count via vertical projection profiling of the spatial coordinates. This geometric prior is injected into the instructional context as a hard constraint for the LLM. To ensure topological consistency, we enforce a leaf node alignment protocol, mandating that the leaf node count in the generated hierarchy strictly matches the estimated data dimensionality. This physical grounding prevents logical hallucinations, ensuring that every bottom-level semantic branch maps to an actual data column. Upon completion of this semantic parsing, we determine the vertical spatial boundary (derived from the y_{max} attributes of the top headers) to rigorously delineate the header region from the data content. Geometrically, this lowest bounding line definitively separates the overarching structure definitions from the numerical data grid. Subsequently, we apply a parallel mechanism to extract the left header hierarchy $\mathcal{H}_{left}$, utilizing the estimated row count to guide the row-level topology. By symmetry, this exploits horizontal spans to resolve parent headers that logically encompass multiple vertically aligned sub-rows. Finally, to tackle complex layouts and isolate the section headers $\mathcal{H}_{sec}$, the LLM evaluates dual-modal cells (incorporating both textual and bounding box features), prioritizing blocks followed by significant visual dividers. For ambiguous cases, we apply semantic filtering to the textual content to prune high-frequency statistical terms while retaining locally unique terms. This approach is driven by the structural intuition that true section headers typically break the standard grid alignment to act as macroscopic category dividers for subsequent rows. This isolates all header components, completing the extraction of the complete header set $\mathcal{H}$ and forming the table's primary topological skeleton.

Following structure extraction, we transform the table into a relational database to facilitate precise SQL execution (Cheng et al.).

Translating a hierarchical topology $\mathcal{H}$ into a flat relational schema inherently risks semantic ambiguity and structural loss. Driven by the intuition that a node's full semantic scope is defined by its ancestry path, we first introduce a recursive schema flattening strategy. We concatenate labels from the root ($root_{top}$ or $root_{left}$) down to the leaf nodes (Wang et al., 2021b). This generates globally unique composite keys (e.g., Europe-UK-London), explicitly preserving the hierarchical context within the database schema. Furthermore, we observe that visual hierarchies often contain abstract intermediate nodes that lack corresponding data entries in $\mathcal{C}$. To prevent the instantiation of vacant records, we implement context-aware row alignment to anchor extracted left row indices to raw data rows, ensuring that only leaf nodes linked to valid values are mapped to executable records. For layouts containing sub-tables, we utilize detected section headers to trigger dynamic instantiation, automatically spawning isolated sub-tables for distinct semantic blocks. Finally, the processed data is populated into a SQLite database. To prevent data loss, rows lacking explicit header matches are heuristically recovered via spatial proximity. This pipeline guarantees structural fidelity, providing a robust symbolic foundation for deterministic SQL query generation.

4.4 QA Pipeline

To tackle Table QA tasks necessitating both numerical calculation and semantic inference, we propose a novel pipeline rooted in question de-

composition and Retrieval-Augmented Generation (RAG) (Lewis et al., 2020). Initially, we leverage a VLM to generate a descriptive summary for each table, aiming to enhance the LLM's contextual understanding of the table semantics. Furthermore, prior to the QA process, we extract the schema and the top-five data rows from each database table. This data profiling step serves as a grounded context for the LLM, significantly mitigating potential hallucinations during subsequent Text-to-SQL generation (Rajkumar et al., 2022).

To balance efficiency with accuracy, we implement an adaptive routing mechanism based on table scale. For small tables, we adopt a Direct QA strategy, feeding the full serialized table database directly into the LLM. This approach bypasses the overhead of SQL generation, effectively mitigating overreasoning and preventing hallucinations often caused by unnecessarily complex intermediate steps on simple data.

To tackle complex multi-hop queries and overcome the limitations of single-modality approaches,we leverage a three-step chain of thought (Wei et al., 2022). First we address the error-proneness of direct complex SQL generation by implementing a Step-back Prompting strategy. Then the system is instructed to decompose the user query into 1–3 atomic sub-questions, generate corresponding SQL queries for each, and finally synthesize the main SQL query based on the intermediate execution results. Acknowledging that SQL struggles to capture unstructured semantics such as table titles or footnotes, our framework integrates the symbolic SQL execution results with the VLM-generated background summary during the final answer generation. This ensures the output is both data-driven and contextually aligned with the specific requirements of the question. Ultimately, this framework effectively synergizes the precision of symbolic computation with the generalization capabilities of LLMs, ensuring high system robustness even when confronting complex reasoning tasks or diverse table contents.

Algorithm 1: Adaptive Multi-stage Table QA Pipeline

Input: Database $\mathcal{D}$, VLM Summary S_{vlm}, User Query $\mathcal{Q}$
Output: Final Answer $\mathcal{A}$
$C_{schema} \leftarrow$ GetTableSchema($\mathcal{D}$) ; `// Retrieve column names and data types (e.g., REAL, TEXT)`
$C_{values} \leftarrow$ GetUniqueValues($\mathcal{D}, col =$ 'row_name') ; `// Ground valid entities to prevent exact-match hallucination`
$Ctx_{light} \leftarrow \{C_{schema}, C_{values}, S_{vlm}\}$;
$Ctx_{full} \leftarrow Ctx_{light} \cup$ TopRowsHTML($\mathcal{D}, 5$);
$N_{row} \leftarrow$ CountAllRows($\mathcal{D}$);
if $N_{row} < \tau_{row}$ **then**
  $\mathcal{D}_{html} \leftarrow$ SerializeFullTableToHTML($\mathcal{D}$);
  $\mathcal{A} \leftarrow \text{LLM}_{\text{direct}}(\mathcal{Q}, \mathcal{D}_{html}, S_{vlm})$;
else
  $\mathcal{Q}_{sb} \leftarrow$ StepBackPrompt($\mathcal{Q}, Ctx_{light}$);
  $\{q_1, \ldots, q_k\} \leftarrow$ Decompose($\mathcal{Q}, \mathcal{Q}_{sb}, Ctx_{light}$);
  $F_{extra} \leftarrow \emptyset$;
  for $i \leftarrow 1$ **to** k **do**
    $sql_i \leftarrow$ Text2SQL(q_i, Ctx_{full})
    $res_i \leftarrow \mathcal{D}$.execute($sql_i$);
    $ans_i \leftarrow \text{LLM}_{\text{sub_ans}}(q_i, res_i, sql_i)$;
    F_{extra}.add($q_i \rightarrow ans_i$);
  `// Execute Main Query and Synthesize Final Answer`
  $sql_{main} \leftarrow$ Text2SQL($\mathcal{Q}, Ctx_{full}$);
  $res_{main} \leftarrow \mathcal{D}$.execute($sql_{main}$);
  $\mathcal{A} \leftarrow \text{LLM}_{\text{final_cot}}(\mathcal{Q}, res_{main}, F_{extra}, S_{vlm})$;
return $\mathcal{A}$;

5 Experiments

5.1 Experimental Setup

We evaluate on the HiTab (Cheng et al., 2022) test set to assess complex hierarchical parsing and numerical aggregation. Our pipeline integrates InternVL-8B (Chen et al., 2024) (VLM), PaddleOCR (Cui et al., 2025), and GPT-4.1 (OpenAI, 2023). We compare against diverse baselines (details in Appendix C).

5.2 Evaluation Metric

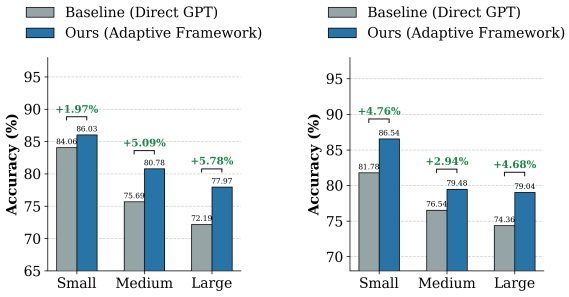
Following standard protocols (Cheng et al., 2022), we adopt execution accuracy as our primary metric. An answer is considered correct if it achieves exact entity matching or falls within an acceptable floating-point tolerance for numerical reasoning.

5.3 Main Results

Table 1 presents our results on the HiTab benchmark, where the proposed framework achieves an accuracy of 81.47%. This performance indicates that integrating explicit visual-spatial anchoring provides essential structural constraints for resolving complex hierarchical tables. Notably, employing standard GPT-4, our method exceeds specialized fine-tuned models like TableGPT2-72B (75.57%) and TableLlama (64.71%), demonstrating that geometry-guided extraction effectively bridges the domain gap without computationally expensive training. Furthermore, translating visual layouts into deterministic SQLite databases alleviates the hallucinations and attention decay common in direct GPT-4 QA. Finally, compared

Table 1: Comparison of our method with other baselines on the HiTab dataset. The results of the baselines are mainly taken from their original papers.

Category	Model	Accuracy (%)
Traditional Systems	MAPO (Liang et al., 2018)	44.80
	NSM (Liang et al., 2017)	43.50
Pre-trained Models	TaPEx (Liu et al.)	52.70
	OmniTab (Jiang et al., 2022)	62.80
Specialist LLMs	TableLlama-7B (Zhang et al., 2024)	64.71
	TableLoRA (He et al., 2025)	58.56
	TableGPT2-7B (Su et al., 2024)	70.27
	TableGPT2-72B (Su et al., 2024)	75.57
General LLMs	GPT-4 (OpenAI, 2023)	71.92
	GPT-4.1 (OpenAI, 2023)	78.32
In-context Learning	CoT (Wei et al., 2022)	73.61
	Chain-of-Table (Wang et al., 2024b)	56.94
	ROT (Zhang et al., 2025)	76.70
	ST-Raptor (Tang et al., 2025)	35.13
	TAMO (Li et al.)	66.39
STAR-Table	**GPT-4.1 (Ours)**	**81.47**



(a) By Row Count (b) By Total Cell Count

Figure 3: Performance comparison between the direct GPT baseline and our adaptive framework across different table size partitions.

to traditional baselines such as TaPEx (52.70%) and OmniTab (62.80%), our OCR-driven approach mitigates hierarchical information loss by preserving tree integrity which conventional sequence linearization often obscures, thereby facilitating accurate cross-level reasoning.

To evaluate our framework's scalability, we partitionize the test set into Small, Medium, and Large subsets using quantile thresholds. Figure 3 illustrates that while the direct GPT baseline's accuracy drops drastically from 84.06% to 72.19% due to attention dilution in lengthy linearized contexts, our framework exhibits exceptional resilience. The performance gap consistently widens as table size increases. By utilizing Adaptive Multi-stage Routing and deterministic SQL execution, our system retrieves only relevant sub-tables, eliminating massive irrelevant data to preserve focused LLM attention on extensive hierarchical structures.

5.4 Ablation Study

To isolate the impacts of visual-spatial perception and global semantics, we independently ablated the OCR and VLM modules. To evaluate robustness against structural complexity, we stratified the HiTab test set by Header Hierarchy Level (L): the maximum depth of the hierarchical tree. Table 2 details performance across flat, two-tier, and deeply nested topologies, highlighting each module's efficacy in resolving escalating structural challenges.

Table 2: Ablation study on Visual-Spatial Perception. Substituting our tailored OCR pipeline with an off-the-shelf parser (MinerU) or entirely removing spatial anchors leads to severe performance degradation, particularly in deeply nested tables (L $\geq$ 3).

Model Variant	Overall	Header Hierarchy Level		
		L = 1	L = 2	L $\geq$ 3
Ours (Full Model)	**81.47**	**84.74**	**82.93**	**75.93**
w/ MinerU Parser	60.66	65.96	61.16	52.88
w/o OCR Anchoring	59.71	65.11	56.78	47.21

OCR model ablation. To evaluate the necessity of explicit spatial perception, we ablated our tailored OCR module, replacing it first with a text-only baseline and then with MinerU, another general OCR model. Eliminating spatial coordinates severely degraded overall accuracy to 59.71%, with performance on deeply nested tables plummeting to 47.21%. Error analysis reveals that lacking geometric constraints causes irreversible hierarchical flattening and severe sub-table ambiguity. Furthermore, MinerU yielded a certain extent improvements, remaining below our full model. This demonstrates that flattened text sequences inherently lack the geometric constraints required to resolve complex 2D topologies. Ultimately, explicit spatial anchoring is indispensable for accurate structural reconstruction.

Table 3: Ablation study on Global Semantic Understanding. The absence of the VLM summary impairs the model's ability to align non-standard semantics, with errors amplifying as hierarchy depth increases.

Model Variant	Overall	Header Hierarchy Level		
		L = 1	L = 2	L $\geq$ 3
Ours (Full Model)	**81.47**	**84.74**	**82.93**	**75.93**
w/o VLM Summary	62.65	67.37	60.52	50.31
Performance Drop (Δ)	(-18.82)	(-17.37)	(-22.41)	(-25.62)

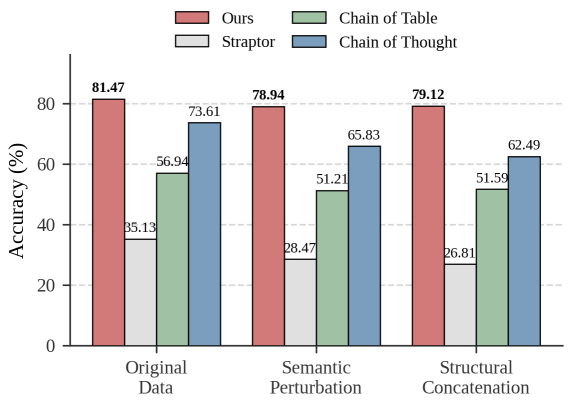


Figure 4: Robustness evaluation under semantic perturbation and structural concatenation.

VLM semantic ablation. In this experiment, we removed the global table summary generated by the VLM, relying exclusively on local cell content for reasoning. Removing the VLM-generated global summary degrades overall QA accuracy to 62.65%, with a precipitous drop to 50.31% on deeply nested tables ($L \geq 3$), despite maintaining structural parsing integrity. This underscores the necessity of high-level semantic context for holistic reasoning. Case analyses reveal two primary causes for this decline. First, without the VLM to visually resolve ambiguous abbreviations and implicit temporal data, the LLM struggles to accurately map user queries to database fields during SQL generation. Second, the absence of global prior knowledge exacerbates hallucinations during multi-hop reasoning, depriving the LLM of a crucial auxiliary verification mechanism.

5.5 Robustness Analysis

To validate that our proposed framework possesses a deep comprehension of table semantics and structure rather than merely overfitting to the dataset distribution, we designed two sets of adversarial perturbation experiments. Based on the original test set, we constructed two variant datasets to stress-test the model's stability across two dimensions: internal semantic noise and external structural variation. As illustrated in Figure 4, our framework consistently outperforms competitive baselines across all perturbed scenarios.

Semantic Perturbation Test. In real-world tabular data, non-standard header naming and irrelevant noise columns are common disturbances. To evaluate robustness against real-world tabular noise, we constructed a semantic perturbation dataset, $\mathcal{D}_{sem}$. Specifically, we injected one to three distractor columns with random values to test schema linking, and applied synonym replacement or case inversion to original headers to assess semantic alignment. Under these perturbations, our framework exhibits exceptional resilience, with accuracy degrading by only 2.53%. In contrast, Chain of Thought and Chain of Table experienced sharper declines of 7.78% and 5.73%, respectively, while specialist models like ST-Raptor deteriorated severely to 28.47%. This robustness is primarily attributed to our VLM-generated global summaries and schema flattening strategy, which synergistically filter incoherent noise and align non-standard header variants.

Structural Concatenation Test. When processing long documents or complex reports, models often encounter multiple independent tables stacked vertically. To evaluate resistance to cross-table aggregation errors common in long documents, we constructed a structural perturbation dataset, $\mathcal{D}_{struct}$. We concatenated a base table with an unrelated, title-less noise table of matching dimensions to create a deceptively continuous data block. Under this perturbation, our framework maintained 79.12% accuracy, demonstrating remarkable resilience against the 11.12% drop seen in Chain of Thought. This stability stems from our Section Header Detection module, which identifies semantic boundaries and isolates noise into separate SQLite sub-tables. Unlike traditional sequence linearization that conflates cross-table values, our SQL-based decoupling ensures strict data integrity, outperforming CoT and Chain of Table by over 16.6% and 27.5%, respectively.

6 Conclusion

We present a geometry-aware framework that synergizes visual perception with symbolic reasoning for complex hierarchical table QA. By explicitly anchoring header topologies via OCR coordinates and transforming unstructured layouts into deterministic SQLite databases, our approach resolves the structural information loss inherent in traditional linearization. Achieving 81.47% accuracy on HiTab, our framework exhibits exceptional robustness against semantic and structural noise, establishing a highly interpretable and reliable parsing-to-execution paradigm for reasoning.

7 Limitations

While our framework demonstrates exceptional reasoning capabilities on complex hierarchical tables, it presents two primary limitations. First, as a pipeline-based approach, the topological reconstruction inherently relies on the precision of the upstream OCR module. For heavily degraded documents or low-resolution image renderings, inaccurate bounding box extractions may induce cascading alignment errors during spatial anchoring. Second, the multi-stage parsing-to-execution paradigm—involving VLM summarization, iterative LLM decomposition, and dynamic SQL execution—incurs higher computational overhead and inference latency compared to traditional end-to-end linearized models. Future work will explore unifying these perception and reasoning modules into a more lightweight, end-to-end multimodal architecture while preserving strict geometric rigor.

References

Gary Bradski. 2000. The opencv library. *Dr. Dobb's Journal of Software Tools*.

Michael J Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. 2008. Webtables: exploring the power of tables on the web. *Proceedings of the VLDB Endowment*, 1(1):538–549.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. Tabfact: Alarge-scale dataset for table-based fact verification.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24185–24198.

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

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, et al. Binding language models in symbolic languages. In *The Eleventh International Conference on Learning Representations*.

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, et al. 2025. Paddleocr 3.0 technical report. *arXiv preprint arXiv:2507.05595*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Xinyi He, Yihao Liu, Mengyu Zhou, Yeye He, Haoyu Dong, Shi Han, Zejian Yuan, and Dongmei Zhang. 2025. Tablelora: Low-rank adaptation on table structure understanding for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22376–22391.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. 2022. Omnitab: Pretraining with natural and synthetic data for few-shot table-based question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 932–942.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Decomposed prompting: A modular approach for solving complex tasks. In *The Eleventh International Conference on Learning Representations*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Liyao Li, Chao Ye, Wentao Ye, Yifei Sun, Zhe Jiang, Haobo Wang, Jiaming Tian, Yiming Zhang, Ningtao Wang, Xing Fu, et al. Table as a modality for large language models.

Chen Liang, Jonathan Berant, Quoc Le, Kenneth Forbus, and Ni Lao. 2017. Neural symbolic machines: Learning semantic parsers on freebase with weak

supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23–33.

Chen Liang, Mohammad Norouzi, Jonathan Berant, Quoc V Le, and Ni Lao. 2018. Memory augmented policy optimization for program synthesis and semantic parsing. *Advances in Neural Information Processing Systems*, 31.

Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Eisenschlos. 2023. MatCha: Enhancing visual language pretraining with math reasoning and chart derendering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12756–12770, Toronto, Canada. Association for Computational Linguistics.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. Tapex: Table pre-training via learning a neural sql executor. In *International Conference on Learning Representations*.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

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

Nitarshan Rajkumar, Raymond Li, and Dzmitry Bahdanau. 2022. Evaluating the text-to-sql capabilities of large language models. *arXiv e-prints*, pages arXiv–2204.

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

Simon Suster, Pieter Fivez, Pietro Totis, Angelika Kimmig, Jesse Davis, Luc de Raedt, and Walter Daelemans. 2021. Mapping probability word problems to executable representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3627–3640, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zirui Tang, Boyu Niu, Xuanhe Zhou, Boxiu Li, Wei Zhou, Jiannan Wang, Guoliang Li, Xinyi Zhang, and Fan Wu. 2025. St-raptor: Llm-powered semi-structured table question answering. *Proceedings of the ACM on Management of Data*, 3(6):1–27.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, et al. 2024a. Mineru: An open-source solution for precise document content extraction. *arXiv e-prints*, pages arXiv–2409.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021a. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1780–1790.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021b. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 1780–1790, New York, NY, USA. Association for Computing Machinery.

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

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, et al. 2024b. Chain-of-table: Evolving tables in the reasoning chain for table understanding. *arXiv e-prints*, pages arXiv–2401.

Haoran Wei, Yaofeng Sun, and Yukun Li. 2025. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. 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*, pages 174–184.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018*

Conference on Empirical Methods in Natural Language Processing, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2024. Tablellama: Towards open large generalist models for tables. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6024–6044.

Xuanliang Zhang, Dingzirui Wang, Keyan Xu, Qingfu Zhu, and Wanxiang Che. 2025. RoT: Enhancing table reasoning with iterative row-wise traversals. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 559–579, Suzhou, China. Association for Computational Linguistics.

Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. 2022. MultiHiertt: Numerical reasoning over multi hierarchical tabular and textual data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6588–6600, Dublin, Ireland. Association for Computational Linguistics.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, et al. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online. Association for Computational Linguistics.

Appendix

A Implementation Details

Hardware and Environment All local model inferences, including the visual semantic extraction via the Vision-Language Model and the spatial coordinate anchoring via the OCR pipeline, were executed on a single NVIDIA A100 GPU. The overall framework and evaluation scripts were implemented in Python.

Model Configurations For the core reasoning, schema linking, and code generation tasks, we utilized the GPT-4 (OpenAI, 2023) and GPT-4.1 large language models, accessed via the Azure OpenAI API. To balance deterministic SQL syntax generation with natural language fluidity, we employed a dynamic temperature scheduling strategy across different phases of the pipeline:

- **SQL Generation:** The temperature was strictly set to 0.01 to enforce deterministic schema linking, exact value matching, and syntactically rigid SQL formulation.
- **Query Decomposition & Step-Back:** The temperatures were set to 0.2 and 0.3, respectively, to encourage adequate semantic abstraction while maintaining strict logical fidelity to the original user query.
- **Final Answer Synthesis:** The temperature was set to 0.2 to synthesize the deterministic SQL execution results into coherent and concise natural language responses.

Vision and OCR Backends For the global semantic understanding phase, we employed InternVL-8B (Chen et al., 2024) as the base VLM to generate the high-level table summaries. For the fine-grained geometric parsing, PaddleOCR (Cui et al., 2025) served as the primary extraction engine. The specific dual-engine configurations and pixel-level refinement strategies for the OCR module are comprehensively detailed in Appendix E.

B Dataset Details: HiTab

To rigorously evaluate our geometry-aware reasoning framework, we utilize the HiTab dataset (Cheng et al., 2022), a comprehensive benchmark constructed from authentic statistical reports.

Overview and Statistics HiTab is sourced from highly professional and complex data portals, including Statistics Canada (StatCan) and the National Science Foundation (NSF). The dataset comprises a total of 3,597 tables and 10,686 question-answering pairs. It aims to evaluate a model's capability to comprehend non-standard table layouts and perform complex semantic parsing over authentic, open-domain analytical reports.

Structural Complexity Unlike standard relational databases (e.g., WikiTableQuestions or Spider) that feature flat, one-dimensional schemas, tables in HiTab exhibit profound structural and topological complexity. The dataset is predominantly characterized by deeply nested top column headers, indented left row headers (indicating parent-child hierarchical relationships), spanning cells, and implicit cross-dimensional category alignments. This high degree of two-dimensional spatial complexity inherently fails traditional sequence-to-sequence linearization approaches, making it an ideal touchstone for evaluating our spatial anchoring and parsing-to-execution paradigm.

Reasoning Challenges Beyond structural comprehension, resolving queries in HiTab intrinsically demands multi-hop numerical reasoning. The QA pairs frequently require cross-level entity aggregation, conditional filtering, and mathematical calculations such as summation, differencing, and rate computation (e.g., calculating the percentage change between two hierarchical data nodes). Consequently, models must achieve exact spatial alignment before any symbolic or numerical execution can yield accurate results.

C Baseline Descriptions

To comprehensively evaluate the effectiveness of our proposed framework, we compare our method against a wide range of competitive baselines, categorized as follows:

Traditional Systems We include MAPO (Liang et al., 2018) and NSM (Liang et al., 2017). These semantic parsing systems utilize hierarchy-aware logical forms and weakly supervised sequence-to-action models to execute symbolic operations over tables.

Pre-trained Table Models We select TaPEx (Liu et al.) and OmniTab (Jiang et al., 2022). These are encoder-decoder models pre-trained on large-scale

tabular data, establishing strong baselines for table comprehension tasks.

Specialist LLMs We compare against LLMs specifically fine-tuned or adapted for tabular reasoning tasks. This includes TableLlama-7B (Zhang et al., 2024), parameter-efficient adaptation approaches like TableLoRA (He et al., 2025), and the recent state-of-the-art TableGPT2 (Su et al., 2024) series.

General LLMs We evaluate powerful closed-source Large Language Models, specifically GPT-4 and GPT-4.1, relying on their strong zero-shot and few-shot reasoning capabilities across unstructured and structured contexts.

Advanced Prompt Engineering We evaluate leading prompting paradigms designed to enhance reasoning, including the standard Chain-of-Thought (CoT) (Wei et al., 2022) for step-by-step logic derivation, and Chain-of-Table (Wang et al., 2024b), which dynamically evolves table contents throughout the reasoning chain. Furthermore, we include recent state-of-the-art frameworks such as ROT (Zhang et al., 2025) (implemented with Llama 3.3), ST-Raptor (Tang et al., 2025) (with GPT-4.1 and InternVL-8B), and the structure-aware multi-hop reasoning approach TAMO (Li et al.).

D Error Analysis

To identify the current boundaries and bottlenecks of the STAR-Table framework, we conducted a rigorous manual error analysis on a random sample of 50 failure cases. By filtering out false negatives caused by formatting mismatches (e.g., exact string matching penalizing "Quebec" vs. "que", or "107" vs. "107.0"), we classified the genuine reasoning failures into three primary stages of our pipeline.

Category 1: OCR & Visual Recognition Failures ($\sim$16%) While our padding and dual-engine strategies mitigate most visual errors, highly degraded document renderings or extremely dense cell layouts occasionally cause the OCR module to miss characters or merge adjacent text blocks. This upstream noise inherently corrupts the raw data before any reasoning can occur.

Category 2: Structural Parsing & Database Conversion Errors ($\sim$38%) Even with accurate OCR coordinates, the LLM occasionally struggles to reconstruct implicit hierarchical relationships (e.g., deeply nested headers or indented sub-categories). This leads to misaligned columns or flattened schemas during the SQLite database instantiation.

- **User Query:** *"What was the percent change of the srr of injury hospitalization for unintentional falls among the aboriginal females?"*
- **Ground Truth:** `-57.9`
- **Model Output:** `-49`
- **Analysis:** The pipeline incorrectly linked the aggregated parent category ("aboriginal population", which has a value of `-49`) to the child node ("females") during the 2D-to-SQL structural parsing. This hierarchical flattening caused the subsequent SQL query to retrieve data from the wrong depth of the tree.

Crucially, our analysis reveals that the LLM's database conversion errors are heavily concentrated in this type of hierarchical loss. This indicates a strong inter-module dependency: enhancing the precision of the upstream OCR module (e.g., capturing micro-level pixel indentations or subtle spatial gaps) could provide stronger geometric constraints, thereby directly reducing the LLM's structural parsing error rate.

Category 3: SQL Generation & QA Reasoning Failures ($\sim$46%) The vast majority of errors occur in the final reasoning phase. In these instances, the database is perfectly constructed, but the LLM generates flawed SQL logic or fails to align with the query's semantic intent. These failures primarily manifest as entity-value confusion or multi-hop mathematical calculation errors.

- **User Query (Semantic Mismatch):** *"Which sells more, moderate-growth enterprises or total average?"*
- **Ground Truth:** `mge` (Moderate-growth enterprises)
- **Model Output:** `303,046`
- **Analysis:** The LLM successfully located the correct cell via SQL but failed to synthesize the categorical entity name requested by the "which" query, returning the raw sales figure instead.
- **User Query (Calculation Failure):** *"How many times aboriginal women aged 15 or*

older were more likely than same-aged non-aboriginal men to report disabilities?"

- **Ground Truth:** `1.76`
- **Model Output:** `5`
- **Analysis:** To answer "how many times", the model must retrieve two disjoint values and execute a division. The erroneous output suggests the LLM wrote a flawed SQL aggregation formula that fetched the wrong denominator, highlighting the ongoing fragility of symbolic mathematical reasoning.

E OCR Implementation Details

To accurately extract text bounding boxes and content from diverse unstructured tables, we implemented a robust, customized Optical Character Recognition (OCR) pipeline based on PaddleOCR. To mitigate common tabular OCR errors such as cross-column text merging or confusing graphical lines with text, our pipeline employs a dynamic dual-engine strategy alongside tailored pre- and post-processing steps.

Image Pre-processing Tables frequently contain text elements positioned extremely close to the image boundaries, which degrades detection accuracy. To resolve this, we apply a dynamic, proportional padding strategy prior to inference. Specifically, we pad the image with a constant white border equivalent to 15% of both the original height and width ($pad_h = 0.15 \times H, pad_w = 0.15 \times W$). Coordinates are deterministically restored to the original scale post-extraction.

Dual-Engine OCR Configuration We maintain two distinct OCR engines to balance high recall with structural safety:

- **Aggressive Engine (Primary):** Designed for high recall to capture fragmented or faint text. Configuration: `text_det_limit_side_len = 4000`, `det_db_unclip_ratio = 2.2` (expanded bounding boxes), and `det_db_box_thresh = 0.5`.
- **Conservative Engine (Fallback):** Triggered only when anomalies are detected. Configuration: `text_det_limit_side_len = 3200`, `det_db_unclip_ratio = 1.6` (tighter boxes), `det_db_box_thresh = 0.6`, with `use_textline_orientation = True` activated.

Anomaly Detection and Fallback Logic We heuristically analyze the initial outputs from the Aggressive Engine to detect topological anomalies. The pipeline automatically falls back to the Conservative Engine up to 5 times if any of the following criteria are met:

1. **Row Merging:** A bounding box horizontally spans $> 92\%$ of the image width with a height > 15 pixels.
2. **Column Merging:** A bounding box vertically spans $> 85\%$ of the image height.
3. **Line Misclassification:** A bounding box exhibits an extreme aspect ratio ($width/height > 60$), indicating table borders misidentified as text.

Pixel-Level Bounding Box Refinement While the Aggressive Engine ensures high recall (via `unclip_ratio = 2.2`), it occasionally yields oversized bounding boxes that overlap with adjacent cells. To address this, we apply an OpenCV-based contour tightening algorithm. A binary threshold (Otsu's method) is applied to the localized Region of Interest (ROI) expanded by 5 pixels. We then compute the minimal bounding rectangle enveloping the actual text contours. To prevent over-shrinking or collapsing boxes into lines, this refinement is aborted if the tightened width or height is reduced by more than 50% compared to the original detection, or if the absolute height falls below 8 pixels.

F Prompt Templates

The specific system instructions and contextual user prompts for each distinct phase in **STAR-Table** are detailed below. Variables dynamically injected into the prompts are denoted by angle brackets, such as $\langle$*Variable*$\rangle$.

F.1 Top Column Header Extraction Prompt

This prompt is designed to extract the top hierarchical headers. We explicitly inject the estimated column count as a hard constraint to enforce the alignment of leaf nodes with data columns.

F.2 Left Row Header Extraction Prompt

Once top headers are resolved, their lowest y-coordinate serves as the boundary for extracting left row headers.

Prompt 1: Top Column Header Extraction

Role: You are an expert in Table Structure Recognition.
Task: Extract *exclusively* the Top Column Headers.
Input Modalities:

1. **OCR List:** Visual boxes.
   ⟨*Inserted OCR Bounding Box List with IDs and Coordinates*⟩
2. **Reference Text:**
   ⟨*Inserted Ground Truth Raw Header Texts*⟩
3. **Geometric Prior:** The table body appears to have roughly ⟨**Estimated Column Count**⟩ columns of data.

Crucial Constraints:

1. **Spatial Localization:** Top Headers are strictly located at the uppermost region (minimal y-coordinates).
2. **Column Exclusion:** Elements residing in the leftmost column (minimal x-coordinates) serve as Row Headers and must be explicitly *ignored*.
3. **Leaf-Node Alignment Protocol:**
   - Within hierarchical headers, the terminal sub-headers (leaf nodes) must maintain a one-to-one correspondence with the underlying numerical data columns.
   - *Validation Mechanism:* The total count of extracted leaf-node items must align with the geometric prior ($\approx$ ⟨*Estimated Column Count*⟩).
   - *Example:* If 5 columns are estimated and only the parent node "Year" is extracted (1 item), its corresponding children (e.g., "Q1", "Q2") must also be explicitly delineated.

Output Specification: JSON list of objects strictly adhering to the schema: `{"text", "ids", "parent_text"}`.

Figure 5: The system prompt for Top Column Header Extraction. It incorporates spatial geometric priors (e.g., coordinate localization) and logical constraints (e.g., leaf-node alignment) to enforce strict topological extraction of the top hierarchical elements.

F.3 Section Header Extraction Prompt

This prompt utilizes both visual emptiness and structural repetition logic to isolate macroscopic category dividers within the table body.

F.4 Vision-Language Model (VLM) Semantic Summary Prompt

This prompt is passed to the VLM alongside the rendered table image to generate a global semantic context, aiding the downstream SQL execution phase.

F.5 Text-to-SQL Generation Prompt

For tables exceeding the row-count threshold, the framework routes the query to the SQL execution pipeline. This prompt forces the LLM to ground its SQL generation in the explicitly provided SQLite schemas, HTML sample rows, and the categorical value distributions extracted during the database instantiation phase.

F.6 Chain-of-Thought Answer Synthesis Prompt

After executing the generated SQL queries, the system aggregates the results. This prompt utilizes a Chain-of-Thought (CoT) mechanism to synthesize the final answer, prioritizing deterministic SQL outputs while leveraging the VLM global summary and decomposed sub-query results for contextual verification.

Prompt 2: Left Row Header Extraction

Role: You are an expert in Table Structure Recognition.
Task: Extract the Left Row Headers (stub column), rigorously capturing any inherent hierarchical structures.
Input Modalities:

1. **OCR List:** Visual boxes.
   $\langle$*Inserted OCR Bounding Box List with IDs and Coordinates*$\rangle$

2. **Contextual Anchors:**
   - Top Header IDs: $\langle$*List of Identified Top Header IDs*$\rangle$
   - **Structural Boundary:** Top Headers terminate at $y = \langle$*Top Header Bottom Y-Coordinate*$\rangle$.
   - **Geometric Prior:** The table body appears to have roughly $\langle$**Estimated Row Count**$\rangle$ rows of data.

Crucial Constraints:

1. **Spatial Constraint:** Selections are strictly limited to elements positioned below the structural boundary ($y > \langle$*Top Header Bottom Y-Coordinate*$\rangle$).

2. **Horizontal Positioning:** Target the leftmost textual elements within each corresponding row.

3. **Hierarchical Indentation:**
   - Row headers frequently exhibit hierarchy (e.g., "North America" $\rightarrow$ "USA").
   - Sub-headers may present horizontal indentation shifts (shifted right), but functionally remain Row Headers.

4. **Leaf-Node Alignment Protocol:**
   - The total count of terminal row headers (leaf nodes) must mathematically correspond to the geometric prior ($\approx \langle$*Estimated Row Count*$\rangle$).
   - Indented child nodes must not be omitted.

Output Specification: JSON list of objects strictly adhering to the schema: `{"text", "ids", "parent_text"}`.

Figure 6: The system prompt for Left Row Header Extraction. This prompt utilizes the computed bounding y-coordinate of the top headers to constraint the spatial search region for left headers, addressing internal hierarchical structures.

Prompt 3: Section Header Extraction

Role: You are an expert in Table Structure Recognition.
Task: Identify intra-table Section Headers (macroscopic category dividers).
Contextual Anchors:

- Top Headers: ⟨*List of Identified Top Header IDs*⟩
- Left Headers: ⟨*List of Identified Left Header IDs*⟩ (Validated Row Headers)

Input Modality:
⟨*Inserted OCR Bounding Box List with IDs and Coordinates*⟩
Heuristic Decision Flow: Evaluate candidates in strict systematic order.

Phase A: Visual Isolation (High-Confidence Candidates)
Identify boundaries utilizing spatial emptiness.

1. **Right-Side Sparsity:** The candidate exhibits substantial empty space unequivocally to its right, lacking data entries.
2. **Structural Lone Nodes:** The candidate occupies an entire row independently.
3. **Exclusion Criterion:** Strictly exclude the global table title (typically localized at the minimal y-coordinate). Section Headers are confined to the table body.

Phase B: Structural Periodicity (Ambiguity Resolution)
Apply cautiously for candidates missed by Phase A (e.g., headers with adjacent aggregations).

1. **Semantic Uniqueness (Critical Filter):**
   - Candidates accompanied by adjacent data must be strictly **UNIQUE** within the local scope.
   - **Discard globally repetitive terms:** Statistical terms (e.g., "Total", "Subtotal", "Average") or repeating items (e.g., "Revenue" appearing 5 times) are NOT Section Headers. They are Data Rows.
   - **Accept unique categorical blocks:** Terms like "North America", "2024", "Segment A" (acting as block titles) can be Section Headers despite adjacent data.
2. **Vertical Sub-group Periodicity:**
   - **Strict Vertical Alignment:** Examine the vertical axis within the identical x-coordinate span.
   - **Pattern Definition:** A valid Section Header serves as the parent to a specific sequence of sub-items that semantically repeats beneath parallel headers.
3. **Exclusion Criterion:** A putative header must span and group **multiple underlying records**; singleton wrappers corresponding to a single row are invalid.

Output Specification: Standard JSON list encompassing string IDs.
Note: If no section headers satisfy these geometric and semantic criteria, explicitly return an empty list `[]`. Do not force a selection, as non-sectioned tables are prevalent.

Figure 7: The system prompt for Section Header Extraction. A sequential heuristic logic flow is adopted to systematically isolate macroscopic category dividers via spatial heuristics and structural periodicity checks.

Prompt 4: VLM Semantic Summary

Role: You are an expert Vision-Language Model.
Task: Analyze the overarching semantic context of the provided table image to facilitate downstream Text-to-SQL querying on a flattened relational schema.
Deliverable: Synthesize a **concise domain-specific formulation summary** (100–150 words).
Strict Guidelines:

1. **Data Exclusion:** Strictly refrain from transcribing specific numerical cell values or enumerating row instances.
2. **Layout Abstraction:** Since the target schema is topologically flattened, avoid describing physical spatial arrangements (e.g., "merged cells", "nested rows").
3. **Semantic Focus:** Concentrate purely on business logic, categorical dimensions, and core concepts.

Required Analytical Dimensions:

1. **Global Context:** Formulate the primary subject matter and the spatio-temporal scope (e.g., "Life expectancy demographics in Canada, 2011").
2. **Column Concept Dimensions:** Abstract the visual headers into categorical attributes (e.g., "Variables are segmented across Population Groups and Biological Sexes").
3. **Row Granularity:** Define the semantic identity of an individual logical record (e.g., "Each row quantifies metrics for a specific Age Group cohort").
4. **Metric Definition:** Specify the unit of measurement governing the continuous data distributions (e.g., "Values assert chronological life expectancy in years").

Objective: This summary aims to ground the SQL generator in real-world nomenclature, explicitly resolving abbreviations or implicit domain variables (e.g., extrapolating "inuit_le" or "male_le").

Figure 8: The system prompt for VLM Semantic Summary generation. This isolates descriptive visual layout characteristics and enforces purely abstract semantic extraction across global context, entity definitions, and metrics, essential to anchor symbolic multi-step reasoning.

Prompt 5: Schema-Grounded Text-to-SQL Generation

Role: You are a precise Data Scientist.
Task: Write a SQLite SQL query to answer the user question. The database contains multiple sub-tables. You must identify the relevant table(s) and columns to execute the query.
Input Modalities (For each sub-table):

1. **Schema Definition:** DDL statements detailing column types (e.g., `REAL` for numeric calculation, `TEXT` for strings).
2. **Categorical Values:** ⟨*Exhaustive List of Unique Values in the Row Index Column*⟩
3. **Sample Rows:** A serialized HTML table providing a visual and structural preview of the data format.

Contextual Metadata Hints:

- The explicitly defined Row Name column typically acts as the semantic index.
- Indentation markers (e.g., "→") in the Row Name column strictly imply hierarchical parent-child relationships.

Crucial Constraints:

1. **Schema Linking Rule:** First, explicitly identify and declare which specific table and columns map to the semantic concepts within the user question.
2. **Exact Value Matching:** When filtering strings via the `WHERE` clause, you MUST utilize the exact string formats provided in the categorical value lists. (e.g., If the list contains 'females → 1', do not write '1 year old female' in the SQL).

Output Specification:

- **Constraints:** ⟨*Explicitly state the Table, Column, and Value constraints identified*⟩
- **SQL:** ⟨*The executable SQLite query*⟩

Figure 9: The system prompt for Text-to-SQL Generation. It integrates deterministic schema definitions, HTML samples, and strict value-matching constraints to mitigate hallucinations during cross-table SQL formulation.

Prompt 6: Chain-of-Thought Final Answer Synthesis

Role: You are an expert Data Analyst.
Task: Synthesize the provided deterministic query results and contextual metadata to formulate a precise final answer to the User Query.
Input Modalities:

1. **Table Background Summary:** ⟨*Global semantic context generated by the VLM*⟩
2. **Main Executed Logic:**
   - SQL Executed: ⟨*The primary SQL query*⟩
   - Resulting Sub-table: ⟨*Markdown representation of the SQL execution output*⟩
3. **Extra Information (Decomposed Sub-queries):** ⟨*Results from prior multi-hop reasoning steps, if applicable*⟩

Chain of Thought (CoT) Verification Protocol:

1. **Analyze Result:** Evaluate the Main Sub-table Result. Does it contain the target answer or is it empty?
2. **Contextual Fallback:** If the primary result yields "No Data", consult the Extra Information or the Table Background Summary for implicit answers.
3. **Mathematical Verification:** If the query necessitates arithmetic (e.g., sum, difference, percentage) and the sub-table only yields raw numerical components, perform the explicit calculation.
4. **Trust Decision Rule:** If the Main Sub-table Result contains a valid value, **TRUST IT** over any contradictory "No data" claims in the Extra Information.

Output Specification:

- **Thought:** ⟨*Step-by-step reasoning explicitly following the verification protocol*⟩
- **Answer:** ⟨*A natural language sentence encapsulating the finding*⟩
- **[Direct Answer]:** ⟨*A strict list format, e.g., [15.5] or ['canada']*⟩

Figure 10: The system prompt for Final Answer Synthesis. It enforces a strict logical hierarchy, mandating the LLM to prioritize the deterministic output of the main SQL execution while employing contextual metadata for ambiguity resolution and multi-hop mathematical calculations.