

Article

MMTR: Strategy-Guided Multimodal Table Reasoning with Reflective Self-Correction

Lixin Bai ^{1,2} , Yibo Ming ^{1,2} and Yanmin Chen ^{1,2,*}

¹ College of Computer Science and Technology, Xinjiang Normal University, Urumqi 830054, China; 107622024210518@stu.xjnu.edu.cn (L.B.); 107622023210527@stu.xjnu.edu.cn (Y.M.)

² Xinjiang Engineering Research Center for Smart Education and Application, Urumqi 830054, China

* Correspondence: cym@xjnu.edu.cn

Abstract

Although multimodal large language models (MLLMs) have achieved remarkable progress in visual question answering, they remain limited in tabular tasks that require fine-grained structured information perception and complex logical reasoning. This limitation primarily stems from the high density of structured information inherent in tables and the scarcity of high-quality instruction tuning data. To address these challenges and improve the model's reasoning accuracy in tables, we propose MMTR, a strategy-guided multimodal table reasoning method with reflective self-correction. Mechanistically, we design a dual-LoRA architecture: the Strategy LoRA is responsible for generating structured reasoning steps, while the Reflection LoRA verifies and self-corrects these initial outputs. Their synergy empowers the model with a closed-loop capability of “reasoning–reflection–correction”. On the data front, we construct StrTab-QA, a large-scale dataset comprising question-answering, negative, and reflection samples, providing diverse supervision signals. During training, we further introduce a progressive “reasoning-to-reflection” fine-tuning strategy to gradually achieve cross-modal alignment and structural adaptation. Furthermore, coupled with an adaptive resizing and padding scheme, our approach effectively preserves table structures and minimizes information distortion during visual encoding. Extensive experiments demonstrate that MMTR consistently outperforms strong baselines across multiple table reasoning benchmarks.

Keywords: multimodal large language models; visual question answering; self-correction; table reasoning

1. Introduction

As an efficient and structured way to present data, tables have become an active research topic [1–3] and play a crucial role in many fields such as financial analysis, scientific research, and daily life. Enabling machines to automatically understand and utilize the rich information in tables has long been a core problem in artificial intelligence. Traditional table understanding techniques, including recent methods based on large language models, have made notable progress. However, most of these approaches share a common limitation: they heavily rely on high-quality text-serialized tables, such as those in Markdown or HTML format. In many real-world scenarios, such as processing tables in scanned PDF documents, webpage screenshots, or photographs, the directly available input is a table image. Accurately converting such images into structured text is itself a highly challenging task [4].



Academic Editor: Paulo Quaresma

Received: 25 May 2026

Revised: 26 June 2026

Accepted: 27 June 2026

Published: 1 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

To address the problem of table image understanding, the research community has proposed an emerging direction known as Multimodal Table Understanding (MTU) [5]. This line of work aims to enable models to read and comprehend table content directly from visual images, similar to humans, and then complete a series of complex tasks such as question answering, analysis, and generation based on user instructions, thereby opening broad prospects for real-world applications. However, although general multimodal large models have demonstrated strong capabilities in various vision–language tasks, they often perform unsatisfactorily when handling tables, which are a unique hybrid of visual and textual structures [6,7]. This limitation mainly stems from three levels of challenges. At the perception level, the model must accurately recognize and localize text and numerical values within cells under complex backgrounds and diverse fonts. At the structured information level, character-level recognition alone is far from sufficient; the model must correctly parse the inherent two-dimensional row–column structure of tables, header hierarchies, and the alignment and affiliation relationships among cells. At the reasoning level, even when text and structured information are partially recognized, the model still needs to perform cross-cell logical, arithmetic, and analytical reasoning to complete complex tasks such as numerical comparison, aggregation, and conditional inference. These difficulties are clearly reflected in existing experimental results. For example, on the table mathematical reasoning dataset TABMWP [8], the early LLaVA v1.5 [9] model achieves only 16.59% accuracy, while the text-optimized Monkey model reaches merely 39.44%. These observations indicate that multimodal table understanding is not a simple vision–language problem but a comprehensive task that simultaneously faces challenges in perception, structured information modeling, and high-level reasoning.

To systematically address the above challenges, this paper proposes a multimodal large model (Multi-Modal Table Reasoning Model, MMTR). Different from existing multimodal table reasoning approaches that mainly focus on direct answer generation, MMTR introduces a reflective self-correction mechanism, enabling the model to not only verify reliable reasoning processes but also correct erroneous outputs. Specifically, after generating an initial reasoning chain and answer, the model evaluates the correctness of its own reasoning process. When the generated answer is correct, the model provides logical justification and supporting evidence for self-verification; when errors are detected, the model identifies the causes of errors and performs reasoning correction. Centered on reflective reasoning capability, this work further constructs instruction data containing structured reasoning, negative samples, and verification and correction supervision, and adopts a progressive training strategy that enables MMTR to gradually learn reliable multi-step reasoning ability and self-reflection capability in complex table scenarios.

The main innovations and contributions of this work are as follows:

- Construction of StrTab-QA, a large-scale multimodal table reasoning dataset consisting of question-answering pairs, negative samples, and reflection data. Unlike existing multimodal table datasets such as TABMWP, MMTab, and SynTab, which mainly focus on positive question-answering supervision, StrTab-QA incorporates three complementary components: question-answering samples for reasoning learning, negative samples containing rule-based and model-generated erroneous responses to expose diverse error patterns, and reflection data providing structured verification and correction supervision. This multi-component design provides richer supervision signals for training reflection-aware multimodal reasoning models beyond datasets relying solely on positive QA pairs.
- Proposal of a reflective self-correction mechanism based on a dual-LoRA architecture. Different from existing self-correction frameworks such as Reflexion and Self-Refine, which mainly rely on inference-time prompting and external feedback generation with-

out updating model parameters, MMTR learns verification and correction capabilities through dedicated reflection samples. Specifically, the Strategy LoRA is responsible for generating structured reasoning processes, while the Reflection LoRA learns both verification and correction behaviors. This mechanism provides supporting evidence for correct reasoning outputs and performs error localization, diagnosis, and correction when incorrect reasoning is identified, thereby enabling table-specific closed-loop “reasoning–reflection–correction” capability within a single lightweight backbone.

- Design of a progressive fine-tuning strategy and adaptive visual encoding scheme. Different from existing multimodal table reasoning models such as Table-LLaVA and SynTab-LLaVA, which mainly adopt conventional alignment and instruction-tuning pipelines, MMTR introduces a four-stage progressive training strategy. Specifically, this strategy separates cross-modal alignment, visual encoder specialization, Q-A reasoning, and reflective verification into different training stages. These stages are progressively organized to enable the model to gradually acquire reliable reasoning and reflection abilities. Meanwhile, the adaptive visual encoding scheme preserves the inherent structural information of table images during visual encoding and reduces information distortion. Extensive experiments show that MMTR achieves strong performance on the main table reasoning benchmark. It outperforms strong multimodal baselines with much larger model sizes. In addition, it shows consistent improvements under zero-shot cross-task and cross-instruction settings. This suggests that the learned reflection mechanism has good transferability.

2. Related Works

This research is situated in the cutting-edge field of multimodal table question answering [10–12]. The rapid advancement of this domain is primarily driven by three intersecting research trends: the continuous evolution of Multimodal Large Language Models (MLLMs), the progressive development of table question answering and reasoning methodologies, and the emerging technological advancements aimed at enhancing the reliability and consistency of model reasoning. Accordingly, we systematically review the related literature along these three dimensions.

2.1. Multimodal Large Language Model

In recent years, Multimodal Large Language Models (MLLMs) have achieved remarkable progress in visual question answering and cross-modal understanding tasks [13]. In the domain of general multimodal modeling, models such as OpenAI’s GPT-4V [14] and Google’s Gemini [15] have demonstrated powerful visual understanding capabilities. In contrast, the visual instruction tuning paradigm proposed by open-source models, represented by LLaVA, has significantly enhanced model generalization in visual QA tasks. Further research has integrated multimodal Chain-of-Thought (CoT) [16] or stronger language backbones into these models, enabling models like Qwen-VL [17], InternVL [18], and LLaVA-NeXT [19] to achieve notable progress in arithmetic and compositional reasoning. However, these methods are primarily oriented toward natural image scenes.

Furthermore, multimodal models designed for document and chart analysis, such as Donut [20] and Pix2Struct [21], have made certain progress in structured information perception. Yet, their focus remains predominantly on end-to-end text reading or layout parsing (e.g., converting images to HTML structures). Consequently, they struggle to support deep arithmetic reasoning and result consistency verification in complex tabular scenarios. Recently, Table-R1 [22] systematically explored inference-time scaling strategies in table reasoning tasks for the first time, significantly improving performance. However, this method overlooks issues such as structured information perception, visual distortion,

and cross-modal alignment present in real-world visual tables. Overall, there is an urgent need for a systematic approach oriented toward structured table reasoning.

2.2. Table Question Answering

Table reasoning tasks encompass Table Question Answering (TQA), Table Fact Verification (TFV), and structured information understanding. Early research primarily focused on pure-text table modeling; for example, TaPas [23] achieved end-to-end learning through table linearization and structured information encoding. With the development of multimodal technologies, the research focus has gradually shifted toward multimodal table understanding [24,25]. Datasets like TABMWP, TAT-QA [26], and TabMCQ [27] are primarily geared toward complex numerical reasoning and hierarchical structured information understanding, while TabFact [28] and PubHealthTab [29] are utilized for verifying tabular factual consistency.

To alleviate the scarcity of annotated multimodal table data, some studies have attempted to leverage large language models for automated data synthesis and expansion. For instance, SynTab [30] enhances model generalization by generating large-scale table QA samples. However, these methods predominantly concentrate on positive sample expansion, with limited efforts dedicated to the construction of error modeling and reflective supervision. Consequently, models lack targeted error feedback signals within complex reasoning chains.

2.3. Model Reasoning

Recently, inference-time scaling strategies have garnered widespread attention, with OpenAI's o1 model demonstrating the effectiveness of scaling compute during the inference phase [31,32]. To mitigate factual hallucinations during generation, several pure-text large model studies (e.g., Reflexion, Self-Refine) [33,34] have introduced self-verification and reflection mechanisms, significantly improving reasoning reliability via an iterative "generate-then-evaluate/correct" workflow. For instance, Chain-of-Thought (CoT) reasoning enhances logical transparency by explicitly generating intermediate reasoning steps [16]. Concurrently, in the domain of multimodal table understanding, a series of explorations targeting complex scenarios have emerged: RITT [35] proposes a retrieval-assisted table QA framework that integrates both textual and visual representations; TabPedia [36] achieves the seamless integration of table perception and comprehension tasks within a unified large model framework via a Concept Synergy mechanism; FinTab-LLaVA [37] significantly enhances professional reasoning capabilities on financial tables through domain-specific instruction tuning and curriculum learning; and M3TQA [38] constructs a large-scale multitask benchmark and instruction-tuning dataset spanning 97 languages, advancing cross-lingual table understanding.

However, most existing multimodal tabular methods still rely on a unidirectional, end-to-end generation paradigm. When confronted with structural distortions and complex cross-modal alignment issues in real-world visual tables, these methods struggle to explicitly identify and correct intermediate reasoning errors, easily leading to implicit yet hard-to-rectify factual deviations. Therefore, explicitly unifying structured information perception with reflection mechanisms remains a critical open problem in current research.

In summary, despite the progress made in multimodal table data construction and cross-modal representation learning, there remains a lack of a multimodal framework capable of unifying fine-grained structured information perception, complex logical reasoning, and result consistency verification. To this end, we propose MMTR, a reflective self-correcting framework for multimodal table reasoning. Specifically, we construct StrTab-QA, a large-scale dataset comprising question-answering samples, negative samples, and

reflection samples, to provide richer reasoning supervision signals. Concurrently, by employing Parameter-Efficient Fine-Tuning techniques represented by LoRA [39] alongside a progressive reasoning-to-reflection fine-tuning strategy, the model is empowered to first generate a structured reasoning path, and subsequently perform self-correction and consistency verification. This significantly enhances reasoning accuracy and result reliability in complex tabular scenarios.

Although MMTR shares the closed-loop reasoning-reflection paradigm with frameworks such as Reflexion and Self-Refine, there are several key differences in how reflection is realized. First, while prior approaches typically rely on in-context prompting at inference time to elicit self-correction, MMTR instantiates both reasoning and reflection as learnable components through two dedicated LoRA adapters (Strategy LoRA and Reflection LoRA), which are trained via a progressive reasoning-to-reflection fine-tuning strategy on StrTab-QA. Reflection LoRA is further trained with structured supervision that distinguishes between verification and correction behaviors: in verification cases, the model learns to justify correct answers using supporting evidence; in correction cases, it learns to identify error types, localize mistakes in reasoning steps, and produce revised solutions. This design provides explicit supervision signals for reflection beyond standard answer-level supervision.

Second, MMTR performs both reasoning and reflection within a single frozen Qwen2.5-3B backbone by switching between lightweight LoRA adapters at inference time, without requiring an additional critic model or external evaluator. This stands in contrast to Reflexion, which relies on a separate LLM-based evaluator to provide verbal feedback.

Third, MMTR is designed for multimodal table reasoning, where reflection must operate over both visual and textual signals. By integrating a hierarchical adaptive visual preprocessing pipeline, the model is better equipped to handle table-specific reasoning failures such as misalignment of rows and columns, cell boundary ambiguity, and numerical grounding errors. We further empirically evaluate the benefit of this learned reflection mechanism against a prompting-based reflection baseline in Section 4.6.2, showing that MMTR achieves additional gains beyond inference-time prompting strategies.

Table 1 summarizes these distinctions across representative methods, highlighting the unique combination of properties offered by MMTR.

Table 1. Comparison of MMTR with representative methods in terms of reflection mechanism and table reasoning design.

Method	Reflection Mechanism	Dedicated Reflection Training	Multimodal Table Reasoning	Learnable Reflection
Reflexion	Prompt-based	×	×	×
Self-Refine	Prompt-based	×	×	×
Table-LLaVA	None	×	✓	×
SynTab-LLaVA	None	×	✓	×
Table-R1	Prompt-based	×	✓	×
MMTR (Ours)	Reflection LoRA	✓	✓	✓

As shown in Table 1, MMTR integrates learnable reflection, dedicated reflection training, and multimodal table reasoning within a unified framework. These characteristics distinguish MMTR from both prompting-based self-correction methods and existing multimodal table reasoning approaches.

3. Methodology

This chapter introduces a strategy-guided multimodal table reasoning method with reflective self-correction. The method mainly consists of the following components. First, the StrTab-QA dataset construction. To address the insufficient supervision of structured information understanding and arithmetic reasoning in existing table data, this work systematically expands the TABMWP dataset. Second, the MMTR adopts Qwen2.5-3B as the core language reasoning engine, integrates the CLIP-ViT-336 visual encoder, and employs a multimodal projection layer to effectively align table visual features with language representations. Meanwhile, a hierarchical adaptive image preprocessing mechanism is introduced to prevent the loss of fine-grained details in input images. Third, the progressive reasoning-to-reflection fine-tuning strategy. To gradually develop the model's capabilities in table perception, reasoning, and verification, this work designs a progressive pipeline from cross-modal alignment to reflective self-correction. In this framework, the strategy module generates structured reasoning steps and preliminary answers, while the reflection module reviews and revises the reasoning process, thereby enhancing both numerical reasoning and self-correction abilities.

3.1. StrTab-QA Construction

To address challenges such as weak table structured information parsing, difficult numerical reasoning, and low output reliability, this work constructs three progressively connected datasets: a Q-A Dataset, a Negative Sample Dataset, and a Reflection Dataset. As shown in Table 2, existing multimodal table datasets such as TABMWP, SynTab, and MMTab provide only positive question-answer samples, offering no supervision signals for error recognition or self-correction. This absence of negative and reflection samples limits the model's ability to detect and recover from reasoning failures in complex tabular scenarios. StrTab-QA addresses this gap by explicitly incorporating all three sample types, providing richer and more targeted supervision for reflection-aware reasoning.

Table 2. Comparison of existing table-based datasets in terms of data composition.

Dataset	QA Samples	Negative Samples	Reflection Samples
TABMWP	✓	×	×
SynTab	✓	×	×
MMTab	✓	×	×
StrTab-QA	✓	✓	✓

The StrTab-QA construction pipeline is shown in Figure 1 (left). First, the table type is automatically identified based on the table code, and the corresponding designed prompt template is selected. Then, the table image, structured code, and prompts are fed into the Q-A generation module to sequentially produce the three types of data. The output of each stage serves as the input to the next stage, forming a progressively enhanced data construction strategy. The final StrTab-QA contains 23,059 training table images and 406k training samples. The images cover five table types: Ordinary (12,420), Price List (5664), Stem-and-Leaf Plot (3324), Schedule (1006), and Supply and Demand Schedule (645), as shown in Figure 1 (right).

To rigorously prevent data leakage and ensure the validity of our evaluation, the partition of our dataset strictly inherits the official data splits of the original TABMWP benchmark. This ensures that the 7686 test tables and the 23,059 training table images are strictly disjoint at the image level. Furthermore, to completely rule out any near-duplicate semantic leakage, we implemented an automated deduplication pipeline prior to training. Specifically, we matched the target answers of each test sample against those in the training

set. If an identical answer was found, we subsequently compared their corresponding question texts. Any training sample exhibiting both an identical question and an identical answer to a test sample was explicitly filtered and removed. These strict protocols guarantee that our evaluation results are not inflated by benchmark contamination.

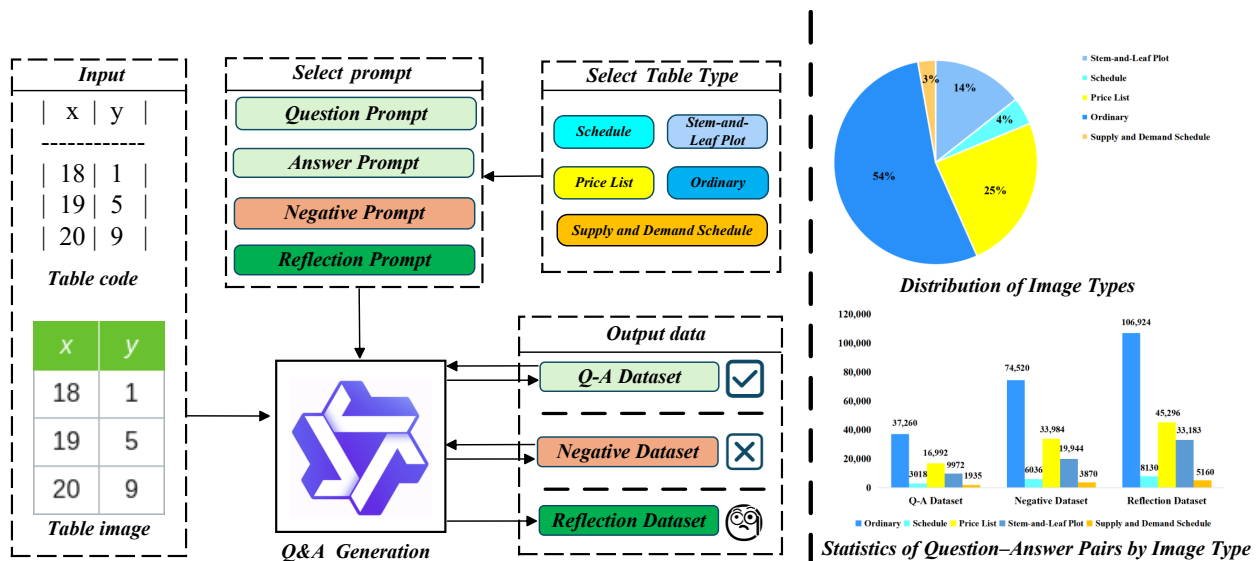


Figure 1. Data Construction Pipeline. (Left) Starting from table type classification, three types of samples are sequentially generated using type-specific prompts: Q-A samples, negative samples with injected reasoning errors, and reflection samples with error correction supervision. (Right) Distribution of question-answer pairs across five table types and three dataset tiers, showing the scale and diversity of StrTab-QA.

3.1.1. Q-A Dataset Construction

In the original TABMWP dataset, each table contains only a single arithmetic question-answer pair, and a portion of the samples heavily relies on candidate options. This makes it inadequate to evaluate multi-dimensional capabilities, such as structural understanding and diverse arithmetic reasoning in table scenarios. To address this limitation, we systematically expand the dataset, ensuring that each table encompasses one structural understanding question and two arithmetic reasoning questions. Notably, all arithmetic reasoning questions are formulated in an option-free, open-ended format. Specifically, based on table content and usage scenarios, tables are categorized into five types: Supply and Demand Schedule, Price List, Schedule, Stem-and-Leaf Plot, and Ordinary. For each table type, a dedicated Question Prompt and Answer Prompt are designed to guide the model in generating questions and corresponding reasoning processes consistent with the characteristics of that table type. During the generation of arithmetic Q-A pairs, the Qwen-VL-MAX model generates the question text using a predefined Question Prompt, while the reasoning process and final answer are produced based on a dedicated Answer Prompt. For structural understanding questions, a predefined set of structure-oriented questions is used to capture key structured information properties of tables, such as table layout, row-column relationships, and statistical characteristics. The corresponding answers are generated by the Qwen-VL-MAX model using Answer Prompt. To ensure data quality, all generated Q-A pairs are verified. After expansion and cleaning, approximately 69k high-quality training samples are obtained.

3.1.2. Negative Sample Dataset Construction

To generate the reflection dataset, this work builds a high-quality negative sample dataset based on the arithmetic Q-A pairs in the Q-A Dataset. The Negative Sample Dataset contains two subcategories. The first subcategory is constructed by programmatically injecting typical errors into otherwise correct reasoning paths, including slight numerical perturbations in arithmetic operations, selecting incorrect cells from the table, row or column misalignment, and misinterpretation of numerical semantics such as maximum values or interval ranges. These errors are highly similar to correct reasoning in form but contain substantive logical deviations, thereby enhancing the model's ability to detect fine-grained reasoning inconsistencies. The second subcategory is generated by directly prompting the model. Specifically, a dedicated Negative Prompt is designed to guide the model in generating incorrect reasoning processes and answers, and the Qwen-VL-MAX model is employed to produce the corresponding negative samples. To ensure overall quality and stylistic consistency between the two subcategories, the generated outputs are filtered and produced through multiple rounds of generation. Only samples with natural and fluent reasoning, coherent semantics, and embedded logical errors are retained. In total, 138k high-naturalness negative samples are constructed, including 92,236 rule-based samples and 46,118 model-generated samples. These error types are designed to capture common failure patterns in multimodal table reasoning rather than arbitrary perturbations.

3.1.3. Reflection Dataset Construction

The construction of the Reflection Dataset further guides the model to learn error correction. To this end, a dedicated Reflection Prompt is designed for both the arithmetic Q-A pairs in the Q-A Dataset and the Negative Sample Dataset. For negative samples, the Reflection prompt requires Qwen-VL-MAX to identify the error type, locate the error position, and generate the correct reasoning path and final answer based on the given "question, incorrect solution, and incorrect answer." For the arithmetic Q-A pairs in the Q-A Dataset, the Reflection prompt instead asks Qwen-VL-MAX to provide supporting evidence and justification for the correctness of the given "question, correct solution, and correct answer." By invoking Qwen-VL-MAX, this work generates reliable corrected or explanatory reasoning paths for each negative sample and each arithmetic Q-A pair in the Q-A Dataset, resulting in approximately 199k reflection samples. This dataset not only enables the model to learn correct table reasoning patterns but also improves the stability and reliability of its reasoning. During the Reflection Dataset construction process, Qwen-VL-MAX is used to generate reflective explanations and correction reasoning over existing samples, while the underlying error patterns are determined by the constructed negative sample dataset.

3.2. MMTR Architecture

3.2.1. Image–Text Processing

Table images are both text-dense and highly structured. Their understanding depends not only on character-level visual information but also on row–column alignment, cell boundaries, and spatial layout relationships. Based on these characteristics, this work adopts CLIP-ViT-336 as the visual encoder to obtain stable and high-quality visual representations under a unified input resolution. In implementation, MMTR first preprocesses the input table image and then feeds it into the CLIP-ViT-336 encoder. After encoding, 576 visual tokens are obtained, each with a dimension of 1024, which are used to capture the local visual features of different table regions and their spatial distribution.

However, real-world table images exhibit heterogeneity in spatial resolution, aspect ratio, and pixel density. Directly resizing them to a fixed resolution can easily cause structural

distortion or detail loss. To alleviate this issue, this work proposes a hierarchical adaptive image preprocessing method based on the maximum edge length. The method maps the input image into a multi-level resolution space (from 336×336 to 3024×3024 pixels) according to its maximum side length and employs an adaptive constraint mechanism to ensure that the input scale is neither excessively large, which would cause computational redundancy, nor too small, which would lead to detail loss. During preprocessing, the image is first resized proportionally according to its original aspect ratio using the Lanczos interpolation algorithm to preserve edge sharpness and structural integrity. The resized image is then centered on a white canvas of the target resolution to form a square input. Through the coupling of multi-scale discretized mapping and the adaptive constraint mechanism, the approach prevents information degradation caused by extreme scaling while preserving the recognizability of numerical symbols and maintaining representation consistency of heterogeneous inputs in the visual encoder's feature space. This preprocessing pipeline is kept consistent during both training and inference and serves as a key component of the end-to-end framework. The framework is shown in Figure 2.

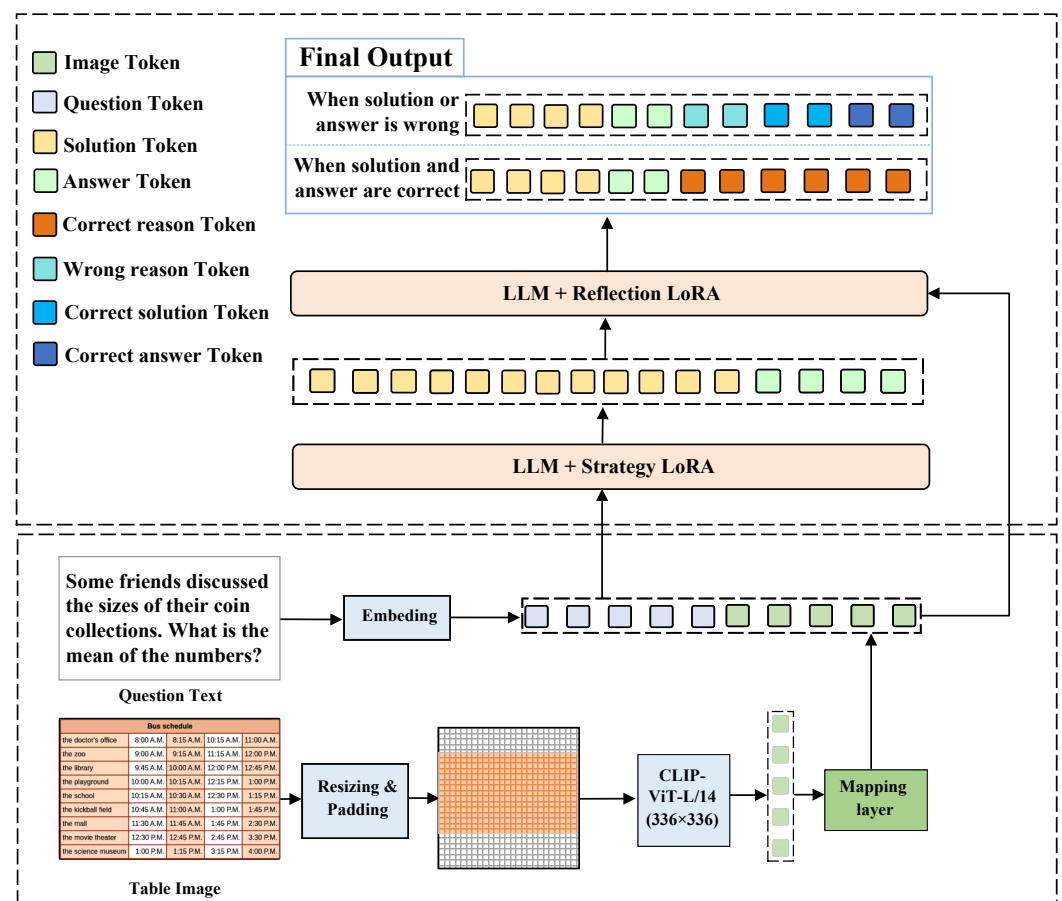


Figure 2. MMTR Architecture Diagram. Table images are encoded via CLIP-ViT-L/14 after adaptive resizing, and aligned with text embeddings through an MLP projection layer. The model operates in a two-stage sequential pipeline: Strategy LoRA generates a preliminary solution and answer, which are then verified or corrected by Reflection LoRA within the same frozen Qwen2.5-3B backbone.

After obtaining stable visual representations, the mismatch between the visual feature space and the language model embedding space must still be addressed. To this end, MMTR introduces a multimodal projection layer as a cross-modal interface to achieve effective alignment between visual features and language representations. The projection layer consists of a two-layer MLP, whose core function is to map the high-dimensional visual features produced by the visual encoder into the embedding space of the language

model, thereby completing dimensional alignment and semantic adaptation. The projected visual tokens can then be fused with text embeddings within a unified semantic space, providing consistent and structured cross-modal input representations for subsequent multi-step reasoning. Finally, the projected visual features are combined with textual embeddings and fed into the strategy module and the reflection module.

3.2.2. Reasoning Strategy

For the language model component, we adopt Qwen2.5-3B as the foundational reasoning model. To equip the model with both solution generation and result verification capabilities, we employ LoRA to fine-tune the language model, yielding two distinct adapters: the strategy module (Strategy LoRA) and the reflection module (Reflection LoRA). Specifically, Strategy LoRA is designed to generate structured reasoning steps and preliminary answers, whereas Reflection LoRA is responsible for verifying and self-correcting these initial outputs. During the inference pipeline, the model operates in a strictly sequential workflow: it first utilizes Strategy LoRA to generate the preliminary reasoning path and answer; subsequently, these intermediate results are directly fed into the model equipped with Reflection LoRA to produce reflective feedback and execute self-correction. The progressive training strategy for these two adapters will be detailed in the following section.

3.3. Progressive Reasoning-to-Reflection Fine-Tuning Strategy

The progressive reasoning-to-reflection fine-tuning strategy consists of four stages: initial alignment of the vision–language projector, table-specific adaptation of the visual encoder, training of the Q-A reasoning module, and construction of the reflective verification and correction module, as illustrated in Figure 3.

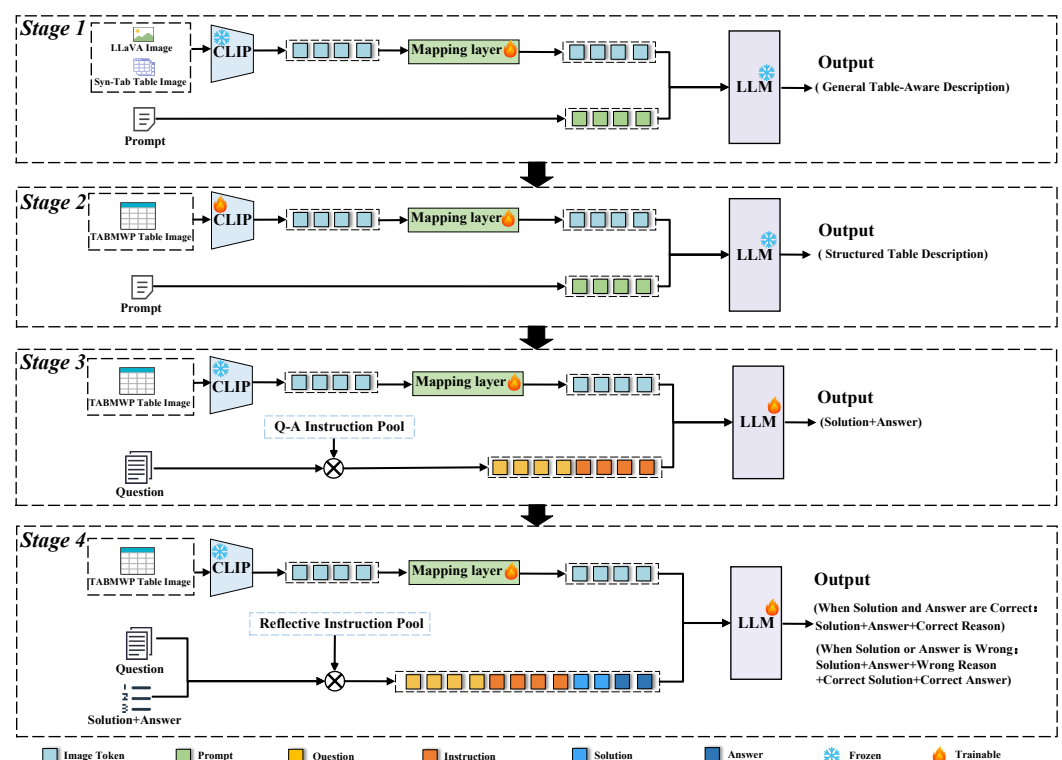


Figure 3. Progressive Reasoning to Reflective Fine-Tuning Strategy. Stage 1 trains only the projection layer for vision–language alignment; Stage 2 adapts the visual encoder to table-specific features via LoRA; Stage 3 trains Strategy LoRA for Q-A reasoning; Stage 4 trains Reflection LoRA for verification and correction. Frozen and trainable components are indicated by snowflake and flame icons respectively.

3.3.1. Stage 1: Initial Alignment of the Vision–Language Projector

The core objective of this stage is to establish a basic mapping between visual features and the language semantic space. To this end, all parameters of the visual encoder and the language model are frozen, and only the alignment layer connecting the two is trained. The training in this stage consists of two steps. First, the model is pretrained using the large-scale general image–text description data collected by LLaVA. This step aims to build a general and transferable vision–language connection. Subsequently, the MMTab pretraining instructions are randomly paired with SynTab table images and table descriptions to form an alignment-layer pretraining dataset, enabling the projector to further specialize in structured table patterns on top of the general capability. After this stage, the model gains the preliminary ability to transform visual regions of table images into feature vectors that can be effectively understood by the language model.

3.3.2. Stage 2: Table-Specific Adaptation of the Visual Encoder

The objective of Stage 2 is to further adapt general visual representations to the specific domain of tables, guiding the visual encoder to focus on fine-grained features closely related to structured information understanding, such as table borders, row–column alignment, and numerical characters. In this stage, the language model is frozen, while the parameters of the visual encoder are released and fine-tuned efficiently using LoRA. At the same time, the alignment layer continues to be optimized. The visual tower is pretrained using the TABMWP pretraining description data constructed in this work. Through backpropagated gradient signals, the visual encoder is encouraged to attend to visual cues that are crucial for answering table-related questions, such as the boundaries of merged cells, hierarchical relationships in table headers, and the precise positions of numerical values. As a result, the model gradually acquires the ability to interpret the structured information in tables more effectively.

3.3.3. Stage 3: Question Answering Reasoning Module

The core objective of Stage 3 is to endow the model with question-answering (Q-A) reasoning capabilities. To this end, we freeze the specialized visual encoder while keeping the multimodal alignment layer trainable, and perform LoRA fine-tuning on the language model. Meanwhile, we construct a dynamic instruction training paradigm to effectively mitigate the model’s tendency to overfit to fixed question formulations. Specifically, based on the Q-A dataset, instead of directly utilizing the raw question for each image-question pair, we introduce a structured Q-A Instruction Pool. This pool predefines dozens of semantically equivalent yet morphologically diverse instruction templates for each task category. During training, an instruction is randomly sampled from the corresponding pool, concatenated with the raw question, and fed into the model. The ground-truth answer is subsequently used as the supervision signal for generative training. Through this hybrid fine-tuning paradigm, we obtain an independent Strategy LoRA module. This module equips the model with the ability to extract generalized task intents from diverse instructions and execute reliable reasoning. For instance, whether a user prompts the model to “estimate...”, “calculate the approximate...”, or “count the rough number of...”, the model can accurately identify the underlying arithmetic aggregation intent, precisely locate relevant data within the table, and execute the correct calculation.

3.3.4. Stage 4: Reflective Verification and Correction Module

Stage 4 establishes a reflective fine-tuning paradigm to train the Reflection LoRA. By leveraging a specially constructed Reflection dataset and a Reflective Instruction Pool, we train the model to verify and correct its prior reasoning results, endowing it with

the higher-order capability to critically evaluate and rectify its own outputs. The detailed training mechanism proceeds as follows: the model's input is formulated as a standardized reflection unit, formulated as Image, Original Question, Reflection Instruction, Candidate Reasoning Chain, Candidate Answer. To ensure that the model grasps the generalized underlying logic of the reflection task rather than merely memorizing specific prompts, we introduce the Reflective Instruction Pool mechanism. This pool contains a diverse array of instruction templates. During each training step, an instruction is randomly sampled to encapsulate the input components. This design explicitly forces the model to master the universal paradigm of reflective reasoning, rather than relying on specific syntactic structures. The training objective of the reflection module is distinctly categorized into two modes, conditioned on the correctness of the input candidate answer:

- **Verification mode:** When the candidate answer is correct, the model is supervised using the logical justification and key evidence provided in the dataset. For example, for a problem that computes total price from unit price and quantity, the supervision signal is: "The solution is correct. The key evidence is that the unit price and quantity are accurately read from the second row of the table, and the multiplication is performed correctly. The result is consistent with the data."
- **Correction mode:** When the candidate answer is incorrect, the model is supervised using the error diagnosis and corrected reasoning provided in the dataset. For example, when an error occurs because the model misreads the start-time and end-time columns and thus computes the activity duration incorrectly, the supervision signal is: "The error occurs at step X. The error type is column misreading. The value from the end-time column is mistakenly used as the start time. The correct procedure is to read the value from the start-time column, followed by the corrected reasoning process and answer."

Under this paradigm, we freeze the visual encoder and keep the multimodal alignment layer trainable, while applying LoRA fine-tuning to the language model to obtain the independent Reflection LoRA adapter. This adapter enables the base model to perform self-reflection after Q-A reasoning. In the final inference pipeline, the initial solution and answer generated by the Strategy LoRA module are fed into the Reflection LoRA module for automatic review. If the module determines that the initial answer is correct, it outputs supporting logical justification and data evidence to complete self-verification. If the module detects an error, it triggers the correction procedure and outputs a complete error report that includes error localization, diagnosis, and the revised answer.

This reflection verification capability is cultivated through the four-stage progressive training strategy outlined below. Under this strategy, the model evolves from basic cross-modal alignment into a multimodal large language model capable of complex table reasoning and self-verification. The first two stages focus on structural modeling of table visual information. The third stage further develops stable Q-A reasoning ability under diverse instruction formulations. The fourth stage introduces verification and correction of the model's own reasoning through reflection fine-tuning. Together, these stages improve both the reliability of the model's outputs and the overall accuracy on table reasoning tasks.

4. Experiments

4.1. Model Configuration

We implement our MMTR framework based on the Qwen2.5-3B model. Across all four progressive training phases, we employ a cosine learning rate scheduler with a linear warm-up for the first 3% of the training steps. Specifically, in Phase 1, we set the batch size to 256 and the learning rate to 1×10^{-3} for 1 epoch. In Phase 2, we introduce LoRA

fine-tuning with a learning rate of 5×10^{-5} for 1 epoch. In Phase 3 for the strategy module and Phase 4 for the reflection module, we set the batch size to 8 and the learning rate to 2×10^{-5} , training the models for 4 and 3 epochs respectively. For all LoRA configurations throughout the fine-tuning process, we set the rank to 128, the scaling factor to 256, and the dropout rate to 0.05, while keeping the multi-modal projector trainable.

4.2. Datasets

The TABMWP test set is used as the primary benchmark for MMTR to systematically evaluate the model's multimodal reasoning ability across different table types. The test set contains 7686 samples, covering five table types: Ordinary (4143), Price List (1884), Stem-and-Leaf Plot (1082), Schedule (343), and Supply and Demand Schedule (234). Furthermore, to more rigorously evaluate the model's authentic reasoning capabilities, we removed the candidate options from the original multiple-choice questions to formulate all samples into a strict open-ended format, while keeping the original table images unchanged. To evaluate the cross-dataset generalization ability, we further utilized four external public datasets—TabMCQ, TAT-QA, TabFact, and PubHealthTab—using the same inference and evaluation protocol as TABMWP.

4.3. Evaluation Metrics

Accuracy is adopted as the primary evaluation metric to measure the consistency between the model's predicted answers and the ground-truth answers. Overall accuracy is defined as the proportion of correctly predicted samples among all test samples. On the TABMWP dataset, results are further reported by table type to comprehensively evaluate the model's performance across different table scenarios. The accuracy evaluation of the proposed model consists of two parts: for the strategy module, accuracy is computed by strict matching between its generated final answer and the ground truth; for the reflection module, Qwen2.5-72B-Instruct is employed as an external evaluator. Specifically, it is provided with the question, the ground-truth answer, and the model prediction.

Since the reflection training data are partially constructed using Qwen-VL-MAX, we further consider the potential concern regarding evaluator independence. To verify that the evaluator does not introduce systematic bias, we randomly sampled 200 reflection predictions and additionally evaluated them using a cross-family LLM evaluator, GPT-4o-mini. The instance-level agreement between Qwen2.5-72B-Instruct and GPT-4o-mini reached 99.50%, with a Cohen's κ of 0.963, and only one sample showed inconsistent judgments. These results indicate that the evaluation decisions are highly consistent across different model families, suggesting that the reported reflection performance is unlikely to be affected by evaluator bias.

4.4. Baseline Models

The proposed model is compared with several mainstream multimodal table understanding models, including the Qwen2.5-VL series, Monkey, the Table-LLaVA series, LLaVA v1.5, SynTab-LLaVA, InternVL-2.5, Ovis2, HIPPO, and MiniCPM-V-2.6. These models cover parameter scales ranging from 3B to 13B.

Qwen2.5-VL: This model supports visual parsing of natural images, document scans, charts, tables, and complex layouts within a unified framework, and can generate structured outputs. It is suitable for various visual question answering and table understanding tasks. The model introduces a window attention mechanism to improve training and inference efficiency and adopts dynamic resolution modeling to enhance adaptability to visual inputs at different scales.

Monkey: This is an efficient multimodal model designed for high-resolution visual input, with strong perception of dense text and fine-grained visual details. It also demon-

strates the ability to understand multi-granularity textual information, from short labels to rich semantic descriptions, effectively capturing contextual relationships between scenes and objects. Monkey shows strong overall performance on image captioning, general VQA, text-dense VQA, and table understanding benchmarks.

Table-LLaVA: This is a multimodal large language model for multimodal table understanding. It aims to perform semantic understanding and task reasoning directly from table images without converting tables into textual sequences such as Markdown or HTML. The model addresses the difficulty of obtaining high-quality textual table representations in real scenarios and explores a visual-centric solution for table understanding. It can handle various table structures and related tasks.

LLaVA v1.5: This is a general-purpose multimodal large language model designed for unified vision–language understanding and dialogue. It is instruction-tuned on multimodal instruction-following data automatically generated by GPT-4, which enhances its zero-shot generalization ability on unseen tasks. LLaVA-v1.5 shows stable performance on general visual question answering and multimodal dialogue tasks.

InternVL-2.5: This model follows the “ViT-MLP-LLM” architectural paradigm and introduces significant improvements in training strategy, data quality, and test-time scaling. It adopts a dynamic high-resolution training strategy and achieves strong performance on general visual understanding, table-type images, and complex VQA tasks, demonstrating good stability and generalization in multimodal reasoning and generation.

SynTab-LLaVA: This is a multimodal model for multimodal table understanding that jointly models local textual content and global structural relationships between cells in table images. It can handle different table types and layouts, including question answering, cell semantic description, and complex structural reasoning.

4.5. Main Experimental Analysis

On the open-ended TABMWP test set, we conducted a unified comparative evaluation between MMTR and several multimodal models. MMTR achieved a comprehensive performance breakthrough on this stricter benchmark, obtaining an overall accuracy of 94.32%, which significantly outperforms all compared baselines. Without substantially increasing the parameter scale, the model improves multimodal table understanding through systematic data construction and an training paradigm, as shown in Table 3.

Table 3. Performance comparison of MMTR with various multimodal models (%).

Model	LLM	Size	Type					All
			Ordinary	Price List	Stem-and-Leaf Plot	Schedule	Supply and Demand Schedule	
Open-source MLLM								
Qwen2.5-VL	Qwen2.5	3B	93.82	97.40	55.69	86.59	65.81	88.16
Monkey [40]	Qwen	7B	44.80	26.70	28.00	69.10	56.41	39.44
Table-LLaVA	Vicuna-1.5	7B	65.82	33.23	45.19	48.98	51.28	53.73
Table-LLaVA	Vicuna-1.5	13B	67.25	33.12	52.31	53.35	55.13	55.79
LLaVA v1.5	Vicuna-1.5	7B	21.22	10.46	4.53	8.45	51.71	16.59
SynTab-LLaVA	Vicuna-1.5	7B	88.08	88.85	57.95	80.47	84.62	83.58
InternVL-2.5	Internlm2.5	8B	91.31	93.79	62.48	92.42	100.00	88.17
Ovis2 [41]	Qwen2.5	8B	95.34	85.19	61.74	80.17	99.57	87.57
HIPPO [42]	Qwen2	8B	89.79	96.49	63.77	79.01	100.00	87.59
MiniCPM-V-2.6 [43]	Qwen2	8B	91.02	97.35	56.47	88.34	96.58	87.76
Ours								
MMTR	Qwen2.5	3B	96.37	97.77	80.28	92.13	98.72	94.32

Specifically, MMTR achieved the best performance on both the regular table and Price List, with accuracies of 96.37% and 97.77%, respectively. This demonstrates its robustness in handling conventional structured information and numerical computation. On the Stem-and-Leaf Plot, MMTR achieved an accuracy of 80.28%, surpassing other models. Since Stem-and-Leaf Plot require both precise visual structure parsing and abstract numerical distribution understanding, this result verifies the unique advantage of the proposed method in handling high-difficulty and irregular table reasoning tasks. MMTR also performed strongly on the timetable and Supply and Demand Schedule, achieving accuracies of 92.13% and 98.72%, respectively. These results place it within the top performance tier and demonstrate its overall competitiveness.

Notably, MMTR uses only 3B parameters, yet it outperforms several larger advanced models, including MiniCPM-V-2.6 and InternVL-2.5. This further confirms the effectiveness of the progressive reasoning-to-reflection fine-tuning strategy.

Compared with the baselines, general MLLMs (such as LLaVA v1.5 and Monkey) perform relatively poorly on table tasks, highlighting the unique challenges of table understanding. Although SynTab-LLaVA and Table-LLaVA show certain improvements, they still lag behind MMTR in complex numerical reasoning. These results further validate the effectiveness of the proposed strategy-based and reflection-enhanced multimodal table reasoning method.

4.6. Ablation Studies

In this section, we separately verify the effectiveness of the progressive training strategy, adaptive image processing, and the reflection instruction mechanism.

4.6.1. Ablation Study on the Progressive Reasoning-to-Reflection Fine-Tuning Strategy

To evaluate the effectiveness of each stage in the progressive training strategy, we conducted stage-wise ablation experiments on the TABMWP test set.

The results are shown in Table 4; it should be noted that in all experimental settings from Stage 1 to Stage 4, the multimodal alignment layer was always involved in training. The modules that were selectively frozen or unfrozen were only the visual encoder and the language model. When neither alignment layer pretraining nor visual tower pretraining was introduced and only Q-A instructions were used, the model achieved 90.67% accuracy. After adding alignment layer pretraining, the performance slightly improved to 90.88%, indicating that establishing stable vision–language feature alignment in advance helps model convergence, although its standalone contribution is relatively limited. After further introducing table-specialized pretraining for the visual tower, the accuracy significantly increased to 92.78%. This demonstrates that visual feature learning optimized for table structures is crucial for multimodal table Q-A and is the primary source of performance gain. With the full configuration and the addition of the reflection instruction mechanism, the model performance further improved to 94.32%, verifying the effectiveness of the explicit reflection process in reducing reasoning errors and improving the reliability of complex arithmetic reasoning. Overall, the results indicate that alignment layer pretraining, visual tower specialization, and the reflection instruction mechanism are functionally complementary. Among them, visual tower pretraining contributes the most, while the reflection mechanism enhances the stability and robustness of the model during reasoning.

Table 4. Partial ablation of the progressive reasoning to reflective fine-tuning strategy (%).

Alignment Layer Pretraining Data	Vision Tower Pretraining Data	Instruction Q-A	Instruction Reflection	Accuracy
×	×	✓	×	90.67
✓	×	✓	×	90.88
✓	✓	✓	×	92.78
✓	✓	✓	✓	94.32

4.6.2. Comparison with Prompt-Based Self-Correction

To distinguish MMTR from existing prompting-based self-correction frameworks, we construct a prompting-based baseline. Specifically, the reflection process is implemented via in-context instructions following the standard generate–evaluate–revise paradigm, without any parameter updates.

To ensure a more reliable evaluation of reflection behavior, we construct a 400-instance stratified subset across five table categories. The sampling strategy explicitly separates correct and incorrect predictions from the Strategy-only model. Specifically, for each category, we cap the number of incorrect samples at up to 20 to ensure sufficient coverage of correction cases. The remaining samples are filled with correctly predicted instances, sampled proportionally according to their natural distribution in the full evaluation set. This design ensures that correction behavior is adequately represented while preserving a realistic distribution of verification cases.

As shown in Table 5, we evaluate three pipelines under the same 400-instance stratified subset. The Strategy-only baseline achieves 79.20%, while the prompting-based reflection baseline improves performance to 82.71%. Our MMTR further achieves 84.46%, demonstrating consistent improvements over both baselines. This consistent gain suggests that the improvements are not solely due to prompting-based self-correction, but also benefit from the learned reflection capability encoded in the Reflection LoRA adapter.

Table 5. Accuracy under different reflection methods (%).

Reflection Method	Accuracy	Gain over Strategy Only
Strategy Only	79.20	—
Prompting-Based Reflection	82.71	+3.51
MMTR	84.46	+5.26

4.6.3. Reflection Behavior Analysis

To comprehensively evaluate the behavior of the reflection module and address the potential risk of over-correction, we conducted a detailed state-transition analysis on the TABMWP test set. The results, averaged across three independent runs (7686 samples per run), are categorized into four states: maintaining a correct answer ($C \rightarrow C$), incorrectly modifying an initially correct answer into a wrong one ($C \rightarrow W$), successfully fixing an error ($W \rightarrow C$), and failing to fix an error ($W \rightarrow W$).

As detailed in Table 6, the reflection module successfully rectifies an average of 157 errors ($W \rightarrow C$) while introducing only 38 over-corrections ($C \rightarrow W$). Furthermore, it correctly preserves 7093 predictions ($C \rightarrow C$) and fails to correct 398 initially incorrect cases ($W \rightarrow W$). Although the “reasoning–reflection–correction” closed loop inevitably introduces a small fraction of over-correction (0.50%), the number of successful corrections is more than four times larger, resulting in a net accuracy improvement of 1.55%.

Table 6. State-transition confusion matrix of the reflection module.

Transition	Count	Percentage (%)
$C \rightarrow C$	7093	92.28
$C \rightarrow W$	38	0.50
$W \rightarrow C$	157	2.05
$W \rightarrow W$	398	5.17
Net Accuracy Improvement	+119	+1.55

4.6.4. Reflection Data Scaling Analysis

To ensure the sampled subsets are representative of the full dataset, we adopt stratified sampling across five table categories, sampling 10% and 25% of instances within each category. Furthermore, the 10% subset is constructed as a strict subset of the 25% subset, ensuring that performance differences across scales reflect data volume alone, independent of data composition.

As shown in Table 7, as the reflection data scale increases, the performance gains exhibit a diminishing-return trend. Notably, using only 25% of the reflection dataset already achieves 91.6% of the full-data improvement. This result indicates that the explicit reflection process plays an important role in improving model performance.

Table 7. Accuracy under different reflection data volumes (%).

Reflection Data	Accuracy	Gain over 0
0	92.78	—
10	93.63	+0.85
25	94.19	+1.41
100	94.32	+1.54

4.6.5. Ablation Study on Image Preprocessing

To analyze the impact of the image preprocessing strategy at different stages, we separately controlled whether image preprocessing was applied during the training and inference phases, and conducted a combinational ablation study on the TABMWP test set. As shown in Table 8, the results show that when the hierarchical adaptive scaling strategy is used in both training and inference, the model achieves the highest accuracy of 92.78%. Compared with the setting that fully adopts the default preprocessing, MMTR shows improved performance. This indicates that maintaining consistent image scale and distribution between training and inference is beneficial for stable table structure perception and subsequent reasoning. The ablation results further reveal the underlying mechanism. When image preprocessing is applied only during inference, the accuracy of MMTR improves to 87.90%. However, when the strategy is used only during training, the performance drops sharply to 52.76%.

Table 8. Image preprocessing ablation study (%).

Training Stage	Inference Stage	Accuracy
Default	Default	62.88
Default	Adaptive Scaling	87.90
Adaptive Scaling	Default	52.76
Adaptive Scaling	Adaptive Scaling	92.78

This asymmetry cannot be fully explained by a generic distribution shift. As shown in Figure 4, we further analyzed the test set by grouping samples according to image

aspect ratio, defined as $r = \max(w, h) / \min(w, h)$, under all four training/inference preprocessing combinations. Both Default/Adaptive Scaling and Adaptive Scaling/Adaptive Scaling remain relatively stable across different aspect ratios. Although the accuracy of Default/Default also decreases as the aspect ratio increases, the decline is substantially steeper for Adaptive Scaling/Default. For near-square images ($r \in [1.0, 1.3)$), the two settings achieve comparable accuracies of 81.03% and 82.73%, respectively. However, for images with aspect ratios $r \geq 2.5$, the accuracy of Adaptive Scaling/Default drops to 14.34%, less than half of that achieved by Default/Default (35.09%).

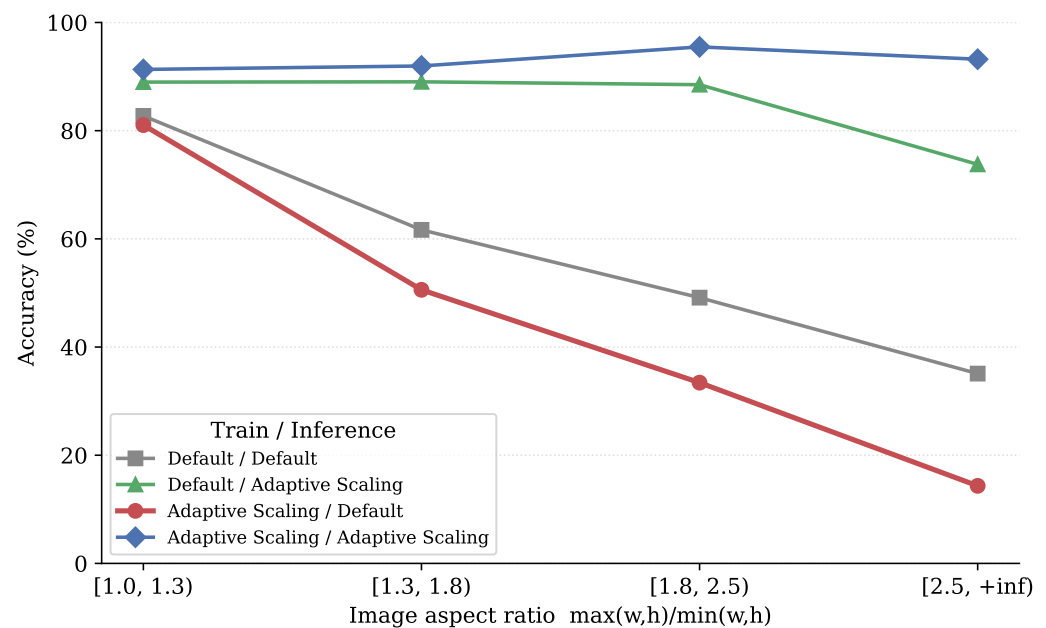


Figure 4. Accuracy vs. image aspect ratio under different preprocessing settings. Adaptive Scaling/Default exhibits substantially greater performance degradation on high-aspect-ratio tables, suggesting the adverse effect of inconsistent preprocessing between training and inference.

These results suggest that an encoder trained exclusively on aspect-ratio-preserved inputs exhibits reduced adaptability when table content is cropped or compressed during inference. The resulting performance degradation is substantially greater than the standard decline observed under the fully-default setting. This indicates that the observed degradation is not merely a consequence of the increased difficulty posed by larger aspect ratios, but is closely associated with the input mismatch caused by inconsistent preprocessing between training and inference.

4.6.6. Ablation Study on Instruction Generalization

To evaluate the efficacy of the Instruction Pool on the model's generalization ability, we conducted an ablation study.

As shown in Table 9, During training, the model learned from randomly sampled diverse templates, while during inference, we compared fixed versus random instruction settings. The results reveal negligible performance fluctuations. In the strategy module, the “random-random” setup maintains a accuracy of 92.61%, showing a minimal difference from the optimal “fixed-fixed” setup of 92.78%. Similarly, in the reflection module, the “random-random” setting achieves a high accuracy of 93.81%, yielding a mere 0.51% drop compared to the “fixed-fixed” setup of 94.32%.

Table 9. Instruction generalization ablation study (%).

Q-A Instruction	Reflection Instruction	Strategy Module Accuracy	Reflection Module Accuracy
Random	Random	92.61	93.81
Fixed	Random	92.78	94.20
Random	Fixed	92.48	94.15
Fixed	Fixed	92.78	94.32

Although fixed instructions offer a minor advantage by providing stable contextual boundaries, the minimal degradation under maximum uncertainty underscores the core benefit of the Instruction Pool. By introducing instruction diversity, the model is prevented from overfitting to specific templates and better simulates the high variance of real-world human queries. Overall, this demonstrates that our dynamic tuning method effectively reduces reliance on rigid syntactic forms, enabling robust task-semantic generalization.

4.7. Generalization Experiments

To evaluate the generalization ability of MMTR across diverse domains, we conducted zero-shot experiments on four external benchmark datasets: TabMCQ (sourced from science exams), TAT-QA (based on real-world financial reports), TabFact (comprising open-domain tables from Wikipedia), and PubHealthTab (focusing on the public health domain). These datasets cover both Table Question Answering (TQA) and Table Fact Verification (TFV) tasks. To further analyze the performance differences between MMTR and existing multimodal large language models on unseen datasets, we included several open-source multimodal large language models as comparative baselines, including Qwen2.5-VL, InternVL2.5, Ovis2, HIPPO, MiniCPM-V-2.6, LLaVA v1.5, and Monkey.

As shown in Table 10, compared with the strategy-only variant, MMTR equipped with the reflection module achieved accuracy improvements of 2.33% and 1.68% on TabMCQ and TAT-QA, as well as 2.81% and 1.90% on TabFact and PubHealthTab, respectively. These consistent gains across diverse tasks demonstrate that the effectiveness of the reflection module is not limited to specific reasoning patterns learned from the training data.

Table 10. Zero-shot generalization experiments (%).

Model	TQA		TFV	
	TabMCQ	TAT-QA	TabFact	PubHealthTab
InternVL2.5-8B	87.27	52.59	71.31	78.37
Qwen2.5-VL	85.62	61.79	68.61	68.64
HIPPO	85.13	60.75	60.75	76.16
Ovis2	82.90	68.78	81.71	84.55
MiniCPM-V-2.6	83.68	51.55	78.48	75.08
LLaVA v1.5	-	2.97	18.90	-
Monkey	17.89	12.31	22.56	18.89
Strategy	73.28	17.62	46.60	56.49
Strategy + Reflection	75.61	19.30	49.41	58.39

Compared with general-purpose MLLMs like LLaVA-1.5 and Monkey, which serve as strict zero-shot references since their fully open-sourced training data explicitly excludes these benchmarks, MMTR achieved higher accuracy across all four datasets. Meanwhile, advanced models such as Qwen2.5-VL, InternVL2.5, Ovis2, HIPPO, and MiniCPM-V-2.6 obtained higher absolute scores on certain tasks. However, compared with the strict zero-shot baselines, these advanced models are developed with large-scale, broadly sourced

pre-training data and larger-scale training resources. Therefore, their higher scores may partly benefit from the broader multimodal knowledge and stronger general capabilities acquired during pre-training.

Overall, the reflection module brings consistent positive improvements across both TQA and TFV tasks. The consistent gains across four unseen benchmarks under a strict zero-shot setting demonstrate that the learned error-detection and correction capabilities can transfer beyond the training distribution.

4.8. Stability Evaluation

We conducted three independent repeated experiments on the final model. As shown in Table 11, the accuracy of the strategy module remained completely consistent, while the reflection module exhibited minimal variation, with an average accuracy of 94.32% and a standard deviation of 0.025. Rather than indicating training-time stability, these results reflect deterministic inference under fixed weights, ensuring reproducibility under identical evaluation settings.

Table 11. Repeated trial evaluation (%).

Type	Run 1	Run 2	Run 3	Average
Strategy	92.78	92.78	92.78	92.78
Strategy + Reflection	94.33	94.30	94.35	94.32

4.9. Statistical Significance of the Reflection Module

To verify that the improvement from the strategy module to the reflection module is statistically meaningful rather than incidental, we conduct McNemar’s exact test on the paired per-sample predictions of $n = 7686$ test samples within each individual run. As shown in Table 12, the improvement is statistically significant in every run ($p < 10^{-17}$ in all cases), and the bootstrap 95% confidence interval for the accuracy gain excludes zero in every run, confirming that the observed gain is unlikely to have arisen by chance.

Table 12. Statistical significance of the reflection module’s improvement.

Run	Accuracy Gain	<i>p</i> -Value	95% Bootstrap CI
Run 1	+1.55%	3.481×10^{-18}	[+1.21%, +1.91%]
Run 2	+1.52%	5.742×10^{-18}	[+1.17%, +1.87%]
Run 3	+1.57%	8.454×10^{-19}	[+1.22%, +1.94%]

4.10. Failure Mode Analysis

To gain a more comprehensive understanding of MMTR’s capabilities across diverse table structures, we analyze the performance and error patterns across different table types. As shown in Table 3, MMTR achieves accuracies above 92% on four table categories. However, the performance on stem-and-leaf plots is notably lower, reaching only 80.28%, which is the lowest among all categories.

To better understand this difference, we further analyze the transition patterns between the strategy and reflection modules in Table 13. Stem-and-leaf plots show both the highest correction rate (10.04% wrong→correct) and the highest over-correction rate (3.14% correct→wrong), which are substantially higher than those of other table types. This suggests that the errors in this category are closely related to the unique structure of stem-and-leaf plots. Unlike standard row–column tables, these plots require the model to interpret shared stem values, associate them with corresponding leaf values, and recon-

struct the complete numbers before performing calculations. Therefore, most errors are likely caused by difficulties in visual structure understanding rather than reasoning itself.

Table 13. Harmful vs. beneficial flip rates by table type.

Table Type	Correct→Wrong	Wrong→Correct	Net Gain
Normal	0.10%	0.67%	+0.56%
Stem-and-Leaf	3.14%	10.04%	+6.90%
Price List	0.00%	0.58%	+0.58%
Supply-and-Demand	0.00%	0.00%	0.00%
Schedule	0.00%	2.92%	+2.92%

These observations suggest that future improvements should focus more on enhancing the visual encoder’s ability to handle irregular layouts and spatial relationships. In contrast, supply-and-demand schedules achieve the highest accuracy (98.72%) with no over-correction, indicating that their regular structure makes them easier to process despite having a relatively small number of samples ($n = 234$).

4.11. Inference Overhead and Cost-Performance Trade-Off

We quantify the inference cost of the reflection mechanism on 100 randomly sampled TABMWP test instances (Table 14). On average, the Strategy module requires 3.72 s and generates 73.6 tokens, while the Reflection module adds 2.89 s and 56.5 tokens, corresponding to a 77.6% overhead in inference latency and a 76.8% overhead in token consumption. This result is expected, as the Reflection module performs a full generation process rather than a simple lightweight check. Unlike iterative refinement strategies, where the computational cost may increase with additional refinement rounds, our method introduces a fixed and predictable overhead.

Table 14. Inference latency and token consumption (100 test samples).

Module	Avg. Latency (s)	Avg. Tokens
Strategy	3.72	73.6
Reflection (additional)	2.89	56.5
Total	6.61	130.0

In return for this additional cost, MMTR reduces the error rate on the TABMWP dataset from 7.22% to 5.68% (Table 4), representing an attractive cost-performance trade-off between reliability gains and inference latency for applications where accuracy is the primary concern.

4.12. Case Study

To deeply analyze the reasoning performance of different models, we present case studies based on two typical tabular scenarios: a Stem-and-Leaf Plot and a Price List, as illustrated in Figure 5. The figure details two distinct output scenarios. The first is Correction mode, where the Strategy LoRA module yields an incorrect preliminary result, and the Reflection LoRA module intervenes to correct it. The second is Verification mode, where the Strategy LoRA module’s initial output is correct, and the Reflection LoRA module verifies and maintains conclusion consistency. We compare our proposed MMTR against baselines including Qwen2.5-VL-3b and Table-LLaVA-3b. In the Correction mode scenario, the Stem-and-Leaf plot features irregular spatial layouts and implicit numerical relationships, causing all models to exhibit comprehension or reasoning errors during the initial phase. Qwen2.5-VL-3b, Table-LLaVA-3b, and the preliminary output of MMTR

all incorrectly conclude that the number of eligible magazines is 6. However, MMTR's reflection module successfully identifies the flaw in the initial reasoning and executes self-correction: the actual qualifying data should include 4 values with a stem of 1 and leaf ≥ 2 , plus 1 value each for stems 2, 3, and 4, ultimately arriving at the correct answer of 7. In contrast, the baseline models fail to recover from their initial errors. In the Verification mode scenario (Price List), Qwen2.5-VL-3b correctly calculates the total price as \$32.37 but paradoxically concludes that funds are insufficient, whereas Table-LLaVA-3b errs early in the arithmetic phase, leading to completely derailed subsequent inference. In stark contrast, the strategy module of MMTR generates a structured reasoning path, and the reflection module further performs rigorous consistency verification on the calculation process and conclusion. By confirming logical self-consistency, it outputs the correct affirmative answer, effectively avoiding the pitfall of correct arithmetic paired with contradictory conclusions.

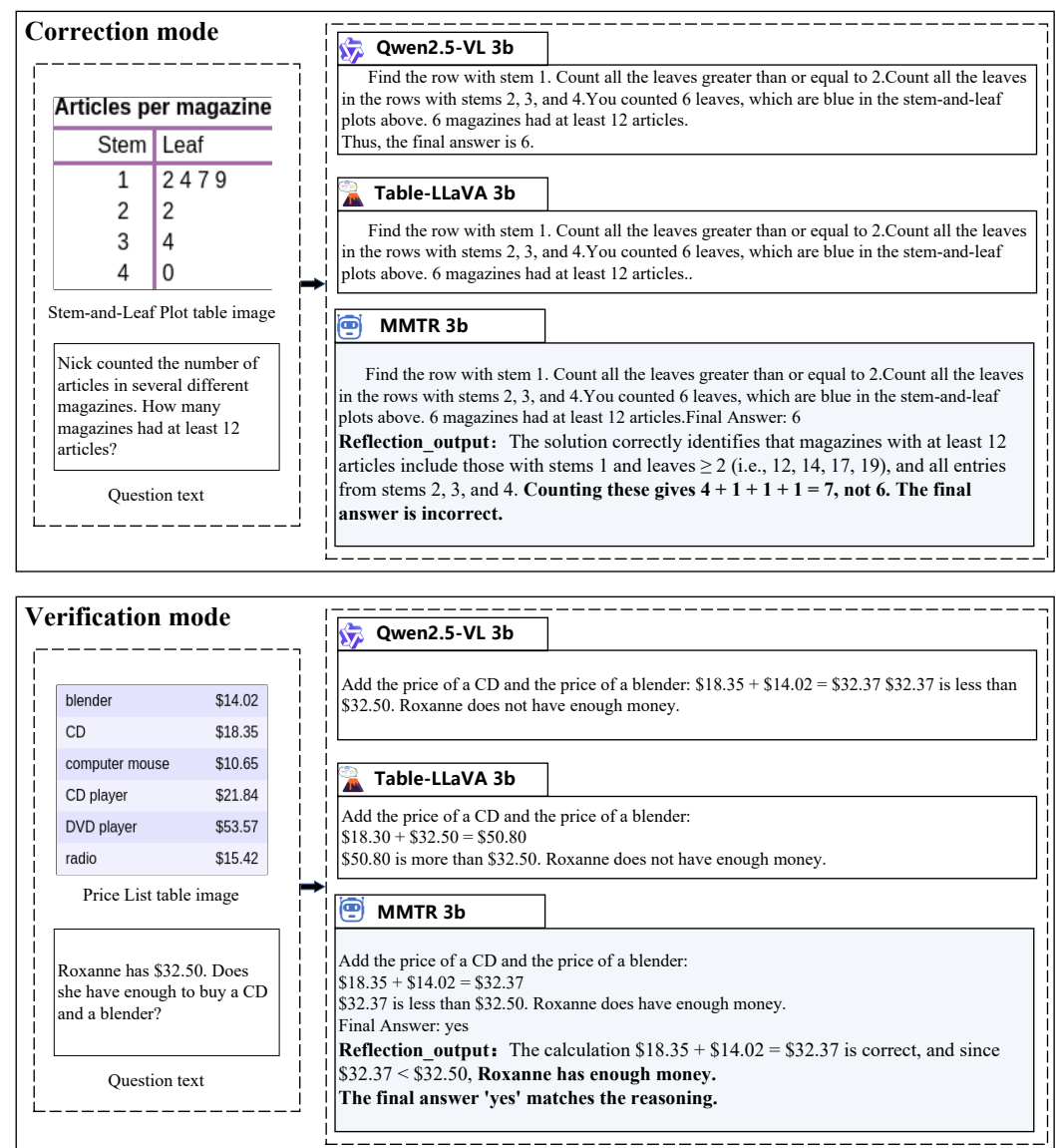


Figure 5. Case Analysis Diagram. Two scenarios are illustrated: Correction mode (Stem-and-Leaf Plot), where the Reflection LoRA corrects a miscount error that all baseline models fail to recover from; and Verification mode (Price List), where the Reflection LoRA confirms the logical consistency of a correct answer that Qwen2.5-VL-3B contradicts despite correct arithmetic.

These cases demonstrate that in complex tabular reasoning tasks, relying solely on preliminary generated results makes models highly susceptible to incorrect outputs caused

by layout misinterpretations, arithmetic mistakes, or improper conclusion formulation. MMTR constructs a structured reasoning path via its strategy module to enhance step completeness, and crucially leverages the reflection module to explicitly identify and rectify deviations in the initial reasoning. This substantiates that the introduction of the reflection mechanism significantly enhances the reliability and stability of tabular reasoning, thereby demonstrating the effectiveness and superiority of our proposed method in multimodal table understanding tasks.

5. Conclusions

This work addresses three major challenges in multimodal table understanding: difficult structural parsing, weak numerical reasoning, and low output reliability. To this end, we propose MMTR, a strategy-guided multimodal table reasoning method with reflective self-correction. Crucially, we introduce a dual-LoRA architecture: a Strategy LoRA explicitly generates structured reasoning steps, while a Reflection LoRA verifies and corrects these initial outputs, endowing the model with a “reasoning–reflection–correction” pipeline. Supported by our newly constructed StrTab-QA dataset and a progressive “reasoning-to-reflection” fine-tuning strategy, along with an adaptive image scaling method, our lightweight MMTR (based on Qwen2.5-3B) jointly enhances table structure understanding, arithmetic reasoning, and self-correction capabilities. On the test set of TABMWP featuring only the free-form question answering format, MMTR achieved an overall accuracy of 94.32%, surpassing several 7B- and 13B-parameter baseline models with only 3B parameters. It also shows improved performance on complex tasks such as Stem-and-Leaf Plot, and demonstrates promising cross-task and cross-instruction generalization.

Building on the proposed reflection-enhanced training framework, several avenues remain for further exploration. First, incorporating more real-world cross-domain tables, especially tables with diverse noise patterns such as scanned documents, distorted reports, and domain-specific formats, could further improve reasoning depth and generalization in challenging real-world scenarios. Second, although our current reflective dataset is constructed based on representative tabular reasoning errors, collecting more reflective samples from actual model failures could further improve the effectiveness and interpretability of the reflection mechanism in real-world applications. Finally, exploring more efficient and expressive table structure encoders, and further strengthening model robustness to low-quality or distorted table images, may continue to improve overall performance on complex table understanding and reasoning tasks.

Author Contributions: Conceptualization, L.B.; methodology, L.B. and Y.M.; software, Y.M.; validation, Y.M. and Y.C.; formal analysis, Y.C.; data curation, L.B.; writing—original draft preparation, L.B.; writing—review and editing, L.B., Y.M. and Y.C.; visualization, Y.M.; supervision, Y.C.; funding acquisition, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: Scientific Research Startup Fund for Doctors (Postdocs) of Xinjiang Normal University (Grant No. XJNUZBS2526).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

- Deng, X.; Sun, H.; Lees, A.; Wu, Y.; Yu, C. Turl: Table understanding through representation learning. *ACM SIGMOD Rec.* **2022**, *51*, 33–40.
- Pasupat, P.; Liang, P. 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)*; Association for Computational Linguistics: Beijing, China, 2015; pp. 1470–1480.
- Zhang, T.; Yue, X.; Li, Y.; Sun, H. 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)*; Association for Computational Linguistics: Mexico City, Mexico, 2024; pp. 6024–6044.
- Zhong, X.; ShafieiBavani, E.; Jimeno Yepes, A. Image-based table recognition: Data, model, and evaluation. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 564–580.
- Mathur, S.V.; Bafna, J.S.; Kartik, K.; Khandelwal, H.; Shrivastava, M.; Gupta, V.; Bansal, M.; Roth, D. Knowledge-aware reasoning over multimodal semi-structured tables. In *Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024*; Association for Computational Linguistics: Miami, FL, USA, 2024; pp. 14054–14073.
- Kim, Y.; Yim, M.; Song, K.Y. Tablevqa-bench: A visual question answering benchmark on multiple table domains. *arXiv* **2024**, arXiv:2404.19205.
- Yang, B.; Zhang, Y.; Liu, D.; Freitas, A.; Lin, C. Does table source matter? benchmarking and improving multimodal scientific table understanding and reasoning. *arXiv* **2025**, arXiv:2501.13042.
- Lu, P.; Qiu, L.; Chang, K.W.; Wu, Y.N.; Zhu, S.C.; Rajpurohit, T.; Clark, P.; Kalyan, A. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv* **2022**, arXiv:2209.14610.
- Liu, H.; Li, C.; Wu, Q.; Lee, Y.J. Visual instruction tuning. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 34892–34916. [\[CrossRef\]](#)
- Chen, W.; Chang, M.W.; Schlenger, E.; Wang, W.; Cohen, W.W. Open question answering over tables and text. *arXiv* **2020**, arXiv:2010.10439.
- Jin, N.; Siebert, J.; Li, D.; Chen, Q. A survey on table question answering: Recent advances. In *Proceedings of the China Conference on Knowledge Graph and Semantic Computing*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 174–186.
- Talmor, A.; Yoran, O.; Catav, A.; Lahav, D.; Wang, Y.; Asai, A.; Ilharco, G.; Hajishirzi, H.; Berant, J. Multimodalqa: Complex question answering over text, tables and images. *arXiv* **2021**, arXiv:2104.06039.
- Yin, S.; Fu, C.; Zhao, S.; Li, K.; Sun, X.; Xu, T.; Chen, E. A survey on multimodal large language models. *Natl. Sci. Rev.* **2024**, *11*, nwae403. [\[CrossRef\]](#) [\[PubMed\]](#)
- Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv* **2023**, arXiv:2303.08774.
- Team, G.; Anil, R.; Borgeaud, S.; Alayrac, J.B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A.M.; Hauth, A.; Millican, K.; et al. Gemini: A family of highly capable multimodal models. *arXiv* **2023**, arXiv:2312.11805.
- Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24824–24837. [\[CrossRef\]](#)
- Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. Qwen2.5-VL Technical Report. *arXiv* **2025**, arXiv:2502.13923. [\[CrossRef\]](#)
- Chen, Z.; Wang, W.; Cao, Y.; Liu, Y.; Gao, Z.; Cui, E.; Zhu, J.; Ye, S.; Tian, H.; Liu, Z.; et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv* **2024**, arXiv:2412.05271.
- Li, F.; Zhang, R.; Zhang, H.; Zhang, Y.; Li, B.; Li, W.; Ma, Z.; Li, C. Llava-next-interleave: Tackling multi-image, video, and 3D in large multimodal models. *arXiv* **2024**, arXiv:2407.07895.
- Kim, G.; Hong, T.; Yim, M.; Park, J.; Yim, J.; Hwang, W.; Yun, S.; Han, D.; Park, S. Donut: Document understanding transformer without ocr. *arXiv* **2021**, arXiv:2111.15664.
- Lee, K.; Joshi, M.; Turc, I.R.; Hu, H.; Liu, F.; Eisenschlos, J.M.; Khandelwal, U.; Shaw, P.; Chang, M.W.; Toutanova, K. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *Proceedings of the International Conference on Machine Learning*; PMLR: Honolulu, HI, USA, 2023; pp. 18893–18912.
- Yang, Z.; Chen, L.; Cohan, A.; Zhao, Y. Table-r1: Inference-time scaling for table reasoning tasks. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Suzhou, China, 2025; pp. 20616–20635.
- Herzig, J.; Nowak, P.K.; Müller, T.; Piccinno, F.; Eisenschlos, J. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Beijing, China, 2020; pp. 4320–4333.
- Zeng, J.; Wu, Z.; Zheng, R.; Xue, W.; Wang, C.; Yu, X.; Zhang, T.; Yuan, S.; Zhu, T. M-TBQA: Multimodal table-based question answering. In *Proceedings of the 2023 4th International Conference on Machine Learning and Computer Application*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 227–231.

25. Zheng, M.; Feng, X.; Si, Q.; She, Q.; Lin, Z.; Jiang, W.; Wang, W. Multimodal table understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Bangkok, Thailand, 2024; pp. 9102–9124.
26. Zhu, F.; Lei, W.; Huang, Y.; Wang, C.; Zhang, S.; Lv, J.; Feng, F.; Chua, T.S. 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)*; Association for Computational Linguistics: Beijing, China, 2021; pp. 3277–3287.
27. Jauhar, S.K.; Turney, P.; Hovy, E. Tabmccq: A dataset of general knowledge tables and multiple-choice questions. *arXiv* **2016**, arXiv:1602.03960.
28. Chen, W.; Wang, H.; Chen, J.; Zhang, Y.; Wang, H.; Li, S.; Zhou, X.; Wang, W.Y. Tabfact: A large-scale dataset for table-based fact verification. *arXiv* **2019**, arXiv:1909.02164.
29. Akhtar, M.; Cocarascu, O.; Simperl, E. PubHealthTab: A public health table-based dataset for evidence-based fact checking. In *Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2022*; Association for Computational Linguistics: Seattle, WA, USA, 2022; pp. 1–16.
30. Zhou, B.; Gao, Z.; Wang, Z.; Zhang, B.; Wang, Y.; Chen, Z.; Xie, H. Syntab-llava: Enhancing multimodal table understanding with decoupled synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2025; pp. 24796–24806.
31. Jaech, A.; Kalai, A.; Lerer, A.; Richardson, A.; El-Kishky, A.; Low, A.; Helyar, A.; Madry, A.; Beutel, A.; Carney, A.; et al. Openai o1 system card. *arXiv* **2024**, arXiv:2412.16720.
32. Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 11809–11822. [[CrossRef](#)]
33. Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; Yao, S. Reflexion: Language agents with verbal reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 8634–8652. [[CrossRef](#)]
34. Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegrefe, S.; Alon, U.; Dziri, N.; Prabhume, S.; Yang, Y.; et al. Self-refine: Iterative refinement with self-feedback. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 46534–46594. [[CrossRef](#)]
35. Zhou, W.; Mesgar, M.; Adel, H.; Friedrich, A. RITT: A Retrieval-assisted framework with Image and Text Table representations for table question answering. In *Proceedings of the 4th Table Representation Learning Workshop*; Association for Computational Linguistics: Vienna, Austria, 2025; pp. 86–97.
36. Zhao, W.; Feng, H.; Liu, Q.; Tang, J.; Wei, S.; Wu, B.; Liao, L.; Ye, Y.; Liu, H.; Zhou, W.; et al. Tabpedia: Towards comprehensive visual table understanding with concept synergy. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 7185–7212. [[CrossRef](#)]
37. Park, H.; Lee, J.; Oh, H. Fintab-llava: Finance domain-specific table understanding multimodal llm using fintmd. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 235–246.
38. Zha, Z.; Qi, P.; Bao, X.; Tian, M.; Qin, B. M3TQA: Multi-View, Multi-Hop and Multi-Stage Reasoning for Temporal Question Answering. In *Proceedings of the ICASSP 2024—2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: New York, NY, USA, 2024; pp. 10086–10090.
39. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. Lora: Low-rank adaptation of large language models. *arXiv* **2022**, arXiv:2106.09685.
40. Li, Z.; Yang, B.; Liu, Q.; Ma, Z.; Zhang, S.; Yang, J.; Sun, Y.; Liu, Y.; Bai, X. Monkey: Image resolution and text label are important things for large multi-modal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2024; pp. 26763–26773.
41. Lu, S.; Li, Y.; Chen, Q.G.; Xu, Z.; Luo, W.; Zhang, K.; Ye, H.J. Ovis: Structural embedding alignment for multimodal large language model. *arXiv* **2024**, arXiv:2405.20797.
42. Liu, Z.; Wang, H.; Li, X.; Xiong, Q.; Yang, X.; Gu, Y.; Yan, Y.; Shi, Q.; Li, F.; Yu, G.; et al. Hippo: Enhancing the table understanding capability of large language models through hybrid-modal preference optimization. *arXiv* **2025**, arXiv:2502.17315.
43. Yao, Y.; Yu, T.; Zhang, A.; Wang, C.; Cui, J.; Zhu, H.; Cai, T.; Li, H.; Zhao, W.; He, Z.; et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv* **2024**, arXiv:2408.01800.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.