THINK IN ROWS AND COLUMNS: DECOUPLED LATENT VISUAL REASONING FOR MULTIMODAL TABLES

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal table reasoning requires Large Vision Language Models to jointly recognize fine-grained cell content and reason over structural relations across rows, columns, and headers. Existing methods either generate answers directly or externalize intermediate computation through textual traces, programs, and image operations. While latent visual reasoning offers a more efficient alternative, current methods lack explicit organization according to table structure. To this end, we propose **Decoupled Latent Visual Reasoning** (DLVR), which organizes continuous latent computation into progressive row, column, and subtable stages while separating structural organization from semantic content. DLVR is trained through decoupled latent supervised fine-tuning followed by **Latent Intervention GRPO**, which employs a **Latent Contribution Reward** to promote the predictive contribution of generated latent components. To supervise this process, we further construct **TableLVR-41K**, a corpus with structure-guided reasoning routes and stage-specific visual and content annotations. Experiments with Qwen2.5-VL and Qwen3-VL across seven benchmarks demonstrate consistent improvements over generic latent reasoning, and explicit table reasoning methods. Ablation studies further validate the complementary contributions of structure-content decoupling, latent supervision objectives, and intervention-based optimization.

1 INTRODUCTION

Tables are widely used to organize and communicate structured information in financial reporting (Chen et al., 2021), scientific data analysis (Moosavi et al., 2021), and web information access (Pasupat & Liang, 2015). Unlike free-form text, table semantics arise jointly from cell content and structural relations defined by row and column headers, hierarchical headers, merged cells, and dependencies across rows and columns (Wang et al., 2021; Smock et al., 2022). Recent advances in Large Vision Language Models (LVLMs) have enabled direct processing of table images, stimulating growing interest in multimodal table understanding and reasoning (Deng et al., 2024; Zheng et al., 2024; Nguyen & Okatani, 2026; Zhu et al., 2026). However, reliable reasoning remains challenging because LVLMs must recognize fine grained textual and numerical content, recover diverse table structures, associate cells with the correct headers, and trace long range dependencies across multiple rows and columns (Titiya et al., 2026).

Existing approaches first improve the general table understanding capabilities of LVLMs through instruction tuning, synthetic data, or table-aware representations (Zheng et al., 2024; Zhao et al., 2024; Zhou et al., 2025; Jiang et al., 2025b). However, these models typically generate answers directly without explicitly modeling how relevant evidence is selected and composed across rows and columns. To strengthen the reasoning process, recent studies either fine-tune models using privileged table representations, preference optimization, reinforcement learning, or process supervision (Wang et al., 2026a; Jiang et al., 2025a; Kang et al., 2025; Liu et al., 2026a; Nguyen & Okatani, 2026; Liu et al., 2026b), or explicitly manipulate task-relevant evidence through visual editing, code execution, and progressive subtable extraction (Fu et al., 2025; Zhu et al., 2026; Yu et al., 2026). Although effective, these methods often externalize intermediate reasoning through language, programs, serialized tables, or modified images, requiring lengthy generation or additional operations. This motivates an internal reasoning mechanism that preserves two-dimensional table structure without explicitly exposing intermediate steps.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Latent visual reasoning offers a promising alternative by performing intermediate computation in continuous visual representation spaces, thereby avoiding lengthy textual rationales and repeated image processing (Li et al., 2026a; Wang et al., 2026b; Dong et al., 2026). However, existing methods are primarily designed for general visual scenes and focus largely on improving reasoning efficiency by replacing explicit intermediate outputs with compact latent computation. The latent states are typically supervised using individual regions, auxiliary images, or unstructured visual features, without explicitly capturing the complementary dependencies along table rows and columns or controlling how evidence is progressively organized. It therefore remains unclear *how latent visual reasoning can be organized according to the relational structure of tables to enable a structured and controllable reasoning process while retaining its efficiency advantages.*

To address these challenges, we introduce **Decoupled Latent Visual Reasoning** (DLVR), a table specific paradigm that extends latent reasoning from a single question relevant region to a structured sequence of visual states. As shown in Figure 1, given a table image and a question, instead of reconstructing a single generic visual region, DLVR decomposes question relevant evidence into complementary row and column latent states and subsequently integrates them into a compact subtable representation. Within each stage, structural organization and semantic content are represented by separate latent states and subsequently fused to recover the corresponding visual semantics. We develop a two stage training pipeline for learning this process. First, decoupled latent supervised fine-tuning jointly optimizes a **decoupled latent reconstruction loss** and a cross entropy loss, enabling each latent stage to separate structural organization from semantic content while reconstructing the corresponding visual evidence and generating the final answer. We then introduce **Latent Intervention GRPO** (LI-GRPO) with a **Latent Contribution Reward** (LCR), which assigns higher rewards when intact latent components provide stronger support for answer prediction. To provide stage-specific supervision, we further construct **TableLVR-41K**, a training corpus with 41.6K examples annotated with structure-guided reasoning routes, stage-specific evidence images, and serialized cell content. During inference, the model follows either reasoning order without intermediate supervision, explicit textual rationales, or external image operations.

Extensive experiments on Qwen2.5-VL (Bai et al., 2025b) and Qwen3-VL (Bai et al., 2025a) across seven multimodal table reasoning benchmarks demonstrate the effectiveness and generality of DLVR. Across both backbones, DLVR consistently outperforms direct answering, textual chain-of-thought, generic latent visual reasoning variants, and explicit table reasoning methods. Ablation studies attribute these gains to the complementary effects of decoupled latent supervision and LI-GRPO, rather than to the additional training data alone.

Our contributions are summarized as follows.

- We propose **Decoupled Latent Visual Reasoning** (DLVR), which performs structure-guided progressive reasoning while separating structural and semantic information in latent space. To supervise this process, we construct TableLVR-41K, a training corpus with reasoning routes and stage-specific visual and content annotations.
- We introduce a two stage training framework combining decoupled latent supervised fine tuning with **Latent Intervention GRPO** (LI-GRPO), which promotes latent utilization through a **Latent Contribution Reward** (LCR).
- Experiments on Qwen2.5-VL and Qwen3-VL demonstrate consistent improvements over strong baselines, while extensive ablations validate each component of our framework.

2 Related Work

Multimodal Table Reasoning. Existing methods improve LVLM table understanding through instruction tuning, synthetic data, and table-aware representations (Zheng et al., 2024; Zhao et al., 2024; Zhou et al., 2025; Jiang et al., 2025b). Reasoning-oriented approaches further employ structured supervision, preference optimization, or reinforcement learning (Wang et al., 2026a; Jiang et al., 2025a; Kang et al., 2025; Liu et al., 2026a;b). Other methods externalize intermediate computation through language and code, image modification, or progressive subtable extraction (Nguyen & Okatani, 2026; Yu et al., 2026; Fu et al., 2025; Zhu et al., 2026). These approaches either generate answers directly or rely on explicit reasoning traces and external operations. In contrast, DLVR internalizes table-structured reasoning within continuous latent visual states.

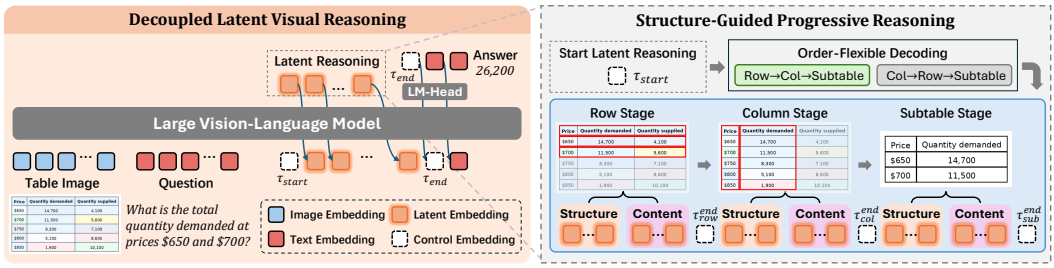


Figure 1: **Inference workflow of Decoupled Latent Visual Reasoning.**

Latent Visual Reasoning. Latent visual reasoning performs intermediate computation in continuous representation spaces, providing an efficient alternative to explicit textual or visual reasoning traces (Hao et al., 2025). Existing methods construct latent states from auxiliary images, relevant regions, sparse visual cues, or unified visual-textual workspaces, and optimize them through reconstruction, reinforcement learning, or task-adaptive supervision (Yang et al., 2026; Li et al., 2026a; Wang et al., 2026b; Dong et al., 2026; Li et al., 2026b; Jiang et al., 2026; Kang et al., 2026b;a). However, these methods are primarily designed for general visual scenes and organize latent computation around regions or unstructured visual features, without explicitly modeling the complementary row and column dependencies of tables. DLVR instead organizes latent computation into progressive row, column, and subtable stages, decouples structural organization from semantic content, and promotes latent-state utilization through intervention-based optimization.

3 Decoupled Latent Visual Reasoning

We introduce **Decoupled Latent Visual Reasoning** (DLVR), a table-specific framework that performs progressive three-stage reasoning with structure-content decoupling in a continuous visual representation space. As illustrated in Figure 1, given a table image and a question, DLVR progressively identifies relevant row-wise and column-wise evidence and integrates their intersection into a localized subtable representation with decoupled latent reasoning. As shown in Figure 2, DLVR is trained through a two-stage pipeline. We first perform decoupled latent SFT to teach LVLMs the progressive latent reasoning process and the structure-content decomposition. We then introduce **Latent Intervention GRPO** (LI-GRPO), which uses a **Latent Contribution Reward** to promote the use of generated latent states in final answer prediction.

3.1 Method Overview

Our framework builds upon a standard LVLM comprising a vision encoder f_v, a multimodal projector g_v, and a language model f_θ with a token embedding layer E. Given a table image $\mathbf{X}$ and a question $\mathbf{q}$, the model obtains visual embeddings $\mathbf{V} = g_v(f_v(\mathbf{X}))$ and question embeddings $\mathbf{Q} = E(\mathbf{q})$. Following latent visual reasoning (Li et al., 2026a), DLVR performs autoregressive latent computation as $\mathbf{h}_t = f_\theta([\mathbf{V}; \mathbf{Q}; \mathbf{z}_{<t}])$, where each hidden state $\mathbf{z}_t = \mathbf{h}_t$ is directly fed back as the input embedding at the next latent position, enabling continuous reasoning in latent space.

Structure-Guided Progressive Reasoning. Unlike generic latent visual reasoning, DLVR organizes its latent trajectory according to the two-dimensional structure of tables. It progressively performs a row stage trained to encode evidence associated with relevant rows, a column stage trained to capture relevant columns and their header associations, and a subtable stage that composes their intersection into compact evidence for answer generation.

We introduce global control tokens τ_{start} and τ_{end} to delimit the latent reasoning process. For each stage $m \in \{\text{row}, \text{column}, \text{subtable}\}$, τ_m^{str} and τ_m^{con} designate its structural and content latent positions, respectively, while τ_m^{end} marks the transition to the next stage. Each component token is repeated K times to allocate a fixed-length latent block.

To avoid coupling reasoning to a single predefined order, DLVR supports two reasoning orders,

$$\rho_{\text{row}} = (\text{row}, \text{column}, \text{subtable}), \qquad \rho_{\text{col}} = (\text{column}, \text{row}, \text{subtable}). \tag{1}$$

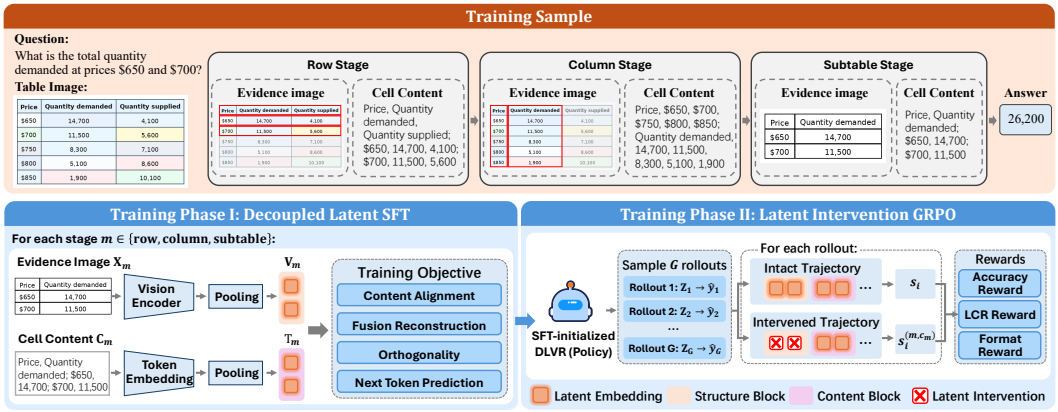


Figure 2: **Training overview.** *Top:* A row-first sample from TableLVR-41K. *Bottom:* Two-stage training pipeline, where decoupled latent SFT learns structure-content representations and LI-GRPO uses LCR to promote the predictive use of latent states.

During supervised fine-tuning, each instance is randomly assigned one of the two reasoning orders, reducing the model's dependence on a fixed row-column sequence. At inference, the decoder follows either order based on the next-token scores of $\tau^{\text{str}}_{\text{row}}$ and $\tau^{\text{str}}_{\text{column}}$. This provides order flexibility rather than question-specific route optimization.

Structure-Content Decoupling. Since table reasoning requires both cell semantics and structural relations, encoding them in a single latent representation may entangle these complementary signals. We therefore divide each reasoning stage into separate structural and content latent blocks.

For each component $c \in \{\text{str}, \text{con}\}$, the latent block $\mathbf{Z}^c_m$ consists of the hidden states generated at the K positions designated by $(\tau^c_m)^K$. Thus, for a selected route ρ, the complete control-augmented latent trajectory is

$$\widetilde{\mathbf{Z}}_\rho = \left[\bar{\boldsymbol{\tau}}_{\text{start}}, \mathbf{Z}^{\text{str}}_{\rho_1}, \mathbf{Z}^{\text{con}}_{\rho_1}, \bar{\boldsymbol{\tau}}^{\text{end}}_{\rho_1}, \mathbf{Z}^{\text{str}}_{\rho_2}, \mathbf{Z}^{\text{con}}_{\rho_2}, \bar{\boldsymbol{\tau}}^{\text{end}}_{\rho_2}, \mathbf{Z}^{\text{str}}_{\text{sub}}, \mathbf{Z}^{\text{con}}_{\text{sub}}, \bar{\boldsymbol{\tau}}^{\text{end}}_{\text{sub}}, \bar{\boldsymbol{\tau}}_{\text{end}}\right], \tag{2}$$

where $\bar{\boldsymbol{\tau}} = E(\tau)$ denotes the embedding of a control token, and (ρ_1, ρ_2) is either $(\text{row}, \text{column})$ or $(\text{column}, \text{row})$. The content states represent the textual and numerical semantics of relevant cells, while the structural states are encouraged to encode complementary organizational information, including header-cell associations and dependencies across rows and columns.

3.2 Two-Stage Training Pipeline

3.2.1 Structure-Guided Supervision Construction

To provide stage-specific supervision, we construct **TableLVR-41K** from the training and development tables of InfoTabs (Gupta et al., 2020), TabFact (Chen et al., 2020), TabMWP (Lu et al., 2023), TAT-QA (Zhu et al., 2021), and WTQ (Pasupat & Liang, 2015). We render the tables as images and prompt a vision-language teacher to generate question-answer pairs across six reasoning types, covering cell lookup, conditional selection, comparison, arithmetic reasoning, ranking, and fact verification, together with a minimal supporting subtable. By matching this subtable to the original table, we derive the supporting rows and columns, construct the corresponding row, column, and subtable evidence images, and serialize their retained cells as content supervision $\mathbf{C}_m$. Each example is randomly assigned a row-first or column-first route, with the subtable stage placed last. After filtering, the corpus contains 41,567 SFT instances, from which 10,000 examples without stage supervision are sampled for LI-GRPO. Figure 2 shows an example, with further construction details and statistics provided in Appendix A.

3.2.2 Decoupled Latent Supervised Fine-Tuning

We first perform supervised fine-tuning to teach the model the structure-guided progressive reasoning trajectory and establish the structure-content decomposition. Each training instance consists of

a table image $\mathbf{X}$, a question $\mathbf{q}$, an answer $\mathbf{y}$, a reasoning route ρ, and stage-specific supervision for the row, column, and subtable stages. The route specifies the order of the first two stages, while each stage comprises a structural latent block followed by a content latent block.

For each stage $m \in \mathcal{M} = \{\text{row}, \text{column}, \text{subtable}\}$, we construct an evidence image $\mathbf{X}_m$ and serialize its relevant cell content as $\mathbf{C}_m$. The evidence image and serialized content are respectively encoded and pooled into K visual and content target states,

$$\mathbf{V}_m = \text{Pool}_K\left(g_v(f_v(\mathbf{X}_m))\right), \qquad \mathbf{T}_m = \text{Pool}_K\left(E(\mathbf{C}_m)\right), \qquad \mathbf{V}_m, \mathbf{T}_m \in \mathbb{R}^{K\times d}, \tag{3}$$

where d denotes the latent dimension. A learnable projection P maps the content targets into the visual latent space. We then derive a residual representation

$$\mathbf{S}_m = \mathbf{V}_m - P(\mathbf{T}_m), \tag{4}$$

which captures the stage-level visual information not explained by the serialized content and serves as a structural proxy during teacher forcing. For a selected route ρ, the control-augmented teacher-forcing sequence is

$$\widetilde{\mathbf{U}}_\rho = \left[\bar{\boldsymbol{\tau}}_{\text{start}}, \mathbf{S}_{\rho_1}, \mathbf{T}_{\rho_1}, \bar{\boldsymbol{\tau}}^{\text{end}}_{\rho_1}, \mathbf{S}_{\rho_2}, \mathbf{T}_{\rho_2}, \bar{\boldsymbol{\tau}}^{\text{end}}_{\rho_2}, \mathbf{S}_{\text{sub}}, \mathbf{T}_{\text{sub}}, \bar{\boldsymbol{\tau}}^{\text{end}}_{\text{sub}}, \bar{\boldsymbol{\tau}}_{\text{end}}\right], \tag{5}$$

where (ρ_1, ρ_2) is either $(\text{row}, \text{column})$ or $(\text{column}, \text{row})$. Following a teacher-forcing strategy, the structural proxies $\mathbf{S}_m$ and content targets $\mathbf{T}_m$ are inserted into their corresponding latent positions, allowing the model to learn the progressive trajectory conditioned on the preceding multimodal and latent context.

Decoupled Latent Reconstruction. During each reasoning stage m, the model produces structural and content hidden-state sequences $\mathbf{H}^{\text{str}}_m$ and $\mathbf{H}^{\text{con}}_m$ at their respective latent positions. Since the textual and numerical semantics of relevant cells can be explicitly serialized, the content states are directly trained to reconstruct the corresponding content targets:

$$\mathcal{L}_{\text{con}} = \frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}} \frac{1}{Kd}\left\|\mathbf{H}^{\text{con}}_m - \mathbf{T}_m\right\|_F^2. \tag{6}$$

In contrast, table structure is inherently relational and does not admit a unique standalone representation. We therefore do not directly regress the structural states toward the residual proxies. Instead, the generated structural and content states are fused to reconstruct the complete visual representation of each stage:

$$\widehat{\mathbf{v}}_m = \text{Avg}_K\left(\mathbf{H}^{\text{str}}_m + P\left(\mathbf{H}^{\text{con}}_m\right)\right), \qquad \mathbf{v}_m = \text{Avg}_K\left(\mathbf{V}_m\right), \tag{7}$$

where Avg_K performs average pooling over the K latent positions. The resulting fusion reconstruction loss is

$$\mathcal{L}_{\text{fus}} = \frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}} \frac{1}{d}\left\|\widehat{\mathbf{v}}_m - \mathbf{v}_m\right\|_2^2. \tag{8}$$

As the content states are explicitly grounded by $\mathcal{L}_{\text{con}}$, reconstructing the complete visual representation encourages the structural states to supply complementary organizational information.

To further prevent the two components from encoding redundant information, we minimize their pairwise cosine correlations:

$$\mathcal{L}_{\text{orth}} = \frac{1}{|\mathcal{M}|K^2}\sum_{m\in\mathcal{M}} \left\|\text{Norm}\left(\mathbf{H}^{\text{str}}_m\right)\text{Norm}\left(\mathbf{H}^{\text{con}}_m\right)^\top\right\|_F^2, \tag{9}$$

where $\text{Norm}(\cdot)$ applies row-wise ℓ_2 normalization. Together, $\mathcal{L}_{\text{con}}$ anchors explicit cell semantics, $\mathcal{L}_{\text{fus}}$ preserves the complete stage-level visual information, and $\mathcal{L}_{\text{orth}}$ promotes the separation of structural and content states.

Next-Token Prediction Loss. Alongside latent reconstruction, the model learns the discrete boundary-control and answer tokens $\{a_t\}_{t=1}^{T_d}$ using the standard next-token prediction objective:

$$\mathcal{L}_{\text{NTP}} = -\frac{1}{T_d}\sum_{t=1}^{T_d} \log p_\theta\left(a_t \mid \mathbf{X}, \mathbf{q}, \widetilde{\mathbf{U}}_\rho, a_{<t}\right). \tag{10}$$

The structural and content latent positions are excluded from this objective.

Joint Objective. The complete supervised fine-tuning objective jointly optimizes latent reconstruction and discrete token generation:

$$\mathcal{L}_{\mathrm{SFT}} = \mathcal{L}_{\mathrm{NTP}} + \lambda_{\mathrm{con}}\mathcal{L}_{\mathrm{con}} + \lambda_{\mathrm{fus}}\mathcal{L}_{\mathrm{fus}} + \lambda_{\mathrm{orth}}\mathcal{L}_{\mathrm{orth}}, \tag{11}$$

where λ_{con}, λ_{fus}, and λ_{orth} balance the three decoupled latent objectives.

3.2.3 Latent Intervention GRPO

After supervised fine-tuning, we further optimize DLVR using **Latent Intervention GRPO** (LI-GRPO). Unlike the SFT stage, this phase does not require stage-specific evidence images or serialized contents. Given an image-question pair, the policy samples a group of G responses, each containing a generated latent trajectory $\widetilde{\mathbf{Z}}_i$ and a textual answer $\widehat{\mathbf{y}}_i$. Following latent GRPO (Li et al., 2026a), the generated latent states are recorded during rollout and replayed when computing the answer-token probabilities for policy optimization.

Latent Contribution Reward. Outcome rewards alone do not guarantee that the generated latent states are used in answer prediction. A model may produce a correct answer while bypassing its intermediate latent trajectory. To measure the predictive contribution of these states, LCR compares the likelihood of the same generated answer under the intact trajectory and an intervened trajectory.

For the i-th rollout, the predictive support provided by the intact latent trajectory is measured by the average log-likelihood of its generated answer:

$$s_i = \frac{1}{T_i}\sum_{t=1}^{T_i} \log p_\theta\left(\widehat{y}_{i,t} \mid \mathbf{X}, \mathbf{q}, \widetilde{\mathbf{Z}}_i, \widehat{y}_{i,<t}\right), \tag{12}$$

where T_i is the answer length. For each stage $m \in \mathcal{M}$, we randomly select one component $c_m \sim \mathrm{Uniform}\{\mathrm{str}, \mathrm{con}\}$ and intervene on its generated states. The intervened trajectory $\widetilde{\mathbf{Z}}_i^{(m,c_m)}$ is obtained by replacing the selected latent block $\mathbf{Z}_{i,m}^{c_m}$ with $[\bar{\boldsymbol{\tau}}_m^{c_m}]^K$, thereby preserving its length, position, and component identity while removing the information encoded in its generated recurrent states. The intervened answer score is computed using the same generated answer:

$$s_i^{(m,c_m)} = \frac{1}{T_i}\sum_{t=1}^{T_i} \log p_\theta\left(\widehat{y}_{i,t} \mid \mathbf{X}, \mathbf{q}, \widetilde{\mathbf{Z}}_i^{(m,c_m)}, \widehat{y}_{i,<t}\right). \tag{13}$$

The contribution of the selected component is quantified by the decrease in answer likelihood after intervention. The final Latent Contribution Reward aggregates the contribution scores across the three reasoning stages:

$$R_i^{\mathrm{LCR}} = \frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}} \tanh\left(s_i - s_i^{(m,c_m)}\right). \tag{14}$$

The intact and intervened answer scores are computed without gradient and affect optimization only through the group-relative advantages. A positive reward indicates that the intact component supports the generated answer, while a non-positive score indicates little or negative contribution. Random component selection provides a stochastic estimate of structural and content contributions without exhaustively evaluating all interventions.

LI-GRPO Objective. Since LCR measures latent contribution rather than answer correctness, we combine it with accuracy and format rewards:

$$R_i = \lambda_{\mathrm{acc}}R_i^{\mathrm{acc}} + \lambda_{\mathrm{fmt}}R_i^{\mathrm{fmt}} + \lambda_{\mathrm{LCR}}R_i^{\mathrm{LCR}}, \qquad \widehat{A}_i = \frac{R_i - \mu_R}{\sigma_R + \epsilon}, \tag{15}$$

where R_i^{acc} evaluates answer correctness, R_i^{fmt} verifies the required output format, and μ_R and σ_R are the mean and standard deviation of rewards within the rollout group.

The recorded latent states are replayed to restore the context preceding answer generation. Using the token-wise importance ratio $r_{i,t}(\theta) = \pi_\theta(\widehat{y}_{i,t} \mid \mathbf{X}, \mathbf{q}, \widetilde{\mathbf{Z}}_i, \widehat{y}_{i,<t}) / \pi_{\theta_{\mathrm{old}}}(\widehat{y}_{i,t} \mid \mathbf{X}, \mathbf{q}, \widetilde{\mathbf{Z}}_i, \widehat{y}_{i,<t})$, the LI-GRPO objective is

$$\mathcal{L}_{\text{LI-GRPO}} = -\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\Big[\min\left(r_{i,t}(\theta)\widehat{A}_i,\ \mathrm{clip}(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon)\widehat{A}_i\right) - \beta D_{\mathrm{KL}}\left(\pi_\theta \| \pi_{\mathrm{ref}}\right)\Big]. \tag{16}$$

The loss is evaluated only on answer tokens, while latent interventions are performed exclusively for reward estimation during training and are disabled at inference.

3.3 INFERENCE

At inference, DLVR requires only the original table image and question. After generating τ_{start}, the decoder follows either the row-first or column-first order based on the scores of the corresponding stage-control tokens. It performs K recurrent steps for each structural and content block, transitions across stages using the stage-end tokens, and generates the final answer after τ_{end}. The two orders provide execution flexibility rather than question-specific route optimization.

4 EXPERIMENTAL SETUP

Datasets. For training, we use TableLVR-41K described in Section 3.2.1. For evaluation, we follow Zhu et al. (2026) and use seven structure-aware table reasoning tasks provided by MMTab (Zheng et al., 2024), including four open-ended table QA benchmarks (i.e., WTQ (Pasupat & Liang, 2015), HiTab (Cheng et al., 2022), TAT-QA (Zhu et al., 2021), and AIT-QA (Katsis et al., 2022)), and three table fact verification benchmarks, (i.e., TabFact (Chen et al., 2020), InfoTabs (Gupta et al., 2020), and PubHealthTab (Akhtar et al., 2022)). All reported experiments are conducted on the test split of each benchmark. More details are described in Appendix B.1.

Baselines. We compare DLVR with representative baselines from four categories. (I) *General-purpose LVLMs* include the closed-source GPT-4o-mini and the open-source Qwen2.5-VL-7B and Qwen3-VL-8B (Bai et al., 2025b;a) with direct answering (DA) and textual chain-of-thought prompting (CoT). (II) *Table-optimized LVLMs* include Table-LLaVA (Zheng et al., 2024), Table-R1 (Kang et al., 2025), HIPPO (Wang et al., 2026a), and DISCO followed by either DA or CoT (Zhu et al., 2026). (III) *Think with image* include Table-GLS (Zhu et al., 2026) based methods. (IV) *Latent visual reasoning methods* include the official LVR checkpoint[1] (Li et al., 2026a) and two variants trained on TableLVR-41K. LVR-Random uses randomly sampled regions from the input table as latent visual targets, while LVR-Subtable uses the image of subtable stage as supervision. We additionally include TableLVR-41K-LoRA, which directly employs LoRA-based SFT on each backbone using TableLVR-41K without latent reasoning. More details are in Appendix B.2.

Implementation Details. We implement DLVR on Qwen2.5-VL-7B-Instruct and Qwen3-VL-8B-Instruct, freezing the vision encoder and multimodal projector while optimizing the language model with LoRA (Hu et al., 2022). During SFT, each structural and content block contains $K = 4$ latent states, and both models are trained for one epoch with a learning rate of 5×10^{-5}, a per-device batch size of 1, 8 gradient accumulation steps. We set $(\lambda_{\text{con}}, \lambda_{\text{fus}}, \lambda_{\text{orth}})$ to $(0.01, 0.1, 0.01)$ for Qwen2.5-VL and $(0.1, 0.5, 0.01)$ for Qwen3-VL. LI-GRPO is initialized from the corresponding SFT checkpoint and trained for one epoch on 10K examples using a learning rate of 5×10^{-6}, a per-device batch size of 8, and 4 rollouts per prompt. The LCR weights are set to 0.2 and 0.1, with LoRA configurations $(r, \alpha) = (8, 16)$ and $(16, 32)$ for Qwen2.5-VL and Qwen3-VL, respectively. Additional hyperparameters and training details are provided in the Appendix B.3.

5 RESULTS AND ANALYSIS

Main Results. Table 1 reports the results on seven multimodal table reasoning tasks. We can observe that (I) DLVR outperforms direct fine-tuning on the same TableLVR-41K corpus across most benchmarks and achieves the best average performance for both backbones, demonstrating the effectiveness of structured latent reasoning beyond the benefit of training data alone. (II) Compared with direct answering and textual CoT, DLVR yields more reliable performance across question answering and fact verification tasks, indicating that continuous latent states provide an effective intermediate reasoning space for tables. (III) DLVR also improves over generic LVR and its region-supervised variants, validating the importance of organizing latent computation according to complementary row, column, and subtable dependencies. (IV) Compared with explicit reasoning method

[1] https://huggingface.co/vincentleebang/LVR-7B

Table 1: **Evaluation results on multimodal table reasoning tasks**, where $_O$ denotes the out-of-domain dataset that is unseen for our methods and ↑ indicates that introducing DLVR improves over TableLVR-41K-LoRA baseline. The best result among the same category of methods are **bolded**.

Model	Method	Question Answering				Fact Verification		
		WTQ	TAT-QA	HiTab$_O$	AIT-QA$_O$	TabFact	InfoTabs	PubHealthTab$_O$
Table-optimized LVLMs								
Table-LLaVA 7B	**SFT**	18.43	12.82	10.09	5.48	59.85	65.26	51.03
Table-LLaVA 13B	**SFT**	20.41	15.67	10.85	6.06	65.00	66.91	48.46
Qwen2-VL-7B-Table-R1	**GRPO**	50.30	48.06	58.20	-	73.40	62.80	-
MiniCPM-V-2.6	**HIPPO (DPO)**	55.77	**60.75**	**63.00**	-	**82.27**	**75.74**	-
Qwen3-VL-8B	**DiSCo+DA**	50.16	37.95	33.50	**75.73**	69.28	63.44	63.34
	DiSCo+CoT	**54.88**	37.31	26.40	73.78	75.34	71.00	**68.33**
General-purpose LVLMs								
GPT-4o-mini	**DA**	30.06	28.37	20.94	54.01	46.54	59.28	48.61
Qwen2.5-VL-7B	**DA**	50.78	37.31	34.72	**79.25**	63.21	60.81	50.87
	CoT	49.22	25.76	**43.65**	58.90	70.08	**69.72**	**70.60**
Qwen3-VL-8B	**DA**	49.47	36.01	35.47	71.04	70.21	45.59	44.18
	CoT	**62.36**	**43.52**	28.17	56.75	**78.14**	67.98	57.52
Think with Image								
Qwen3-VL-8B	**Table-GLS**	**62.15**	**45.85**	**29.76**	55.58	**77.05**	**73.59**	72.76
	DiSCo+Table-GLS	57.11	40.54	27.35	**76.71**	75.41	72.67	**77.14**
Latent Visual Reasoning								
Qwen2.5-VL-7B	**LVR**	51.36	27.07	38.26	61.06	76.48	46.96	35.07
	LVR-Random	57.30	36.92	42.45	86.69	78.10	71.20	79.35
	LVR-Subtable	55.55	37.82	42.07	82.00	77.01	71.65	78.27
Qwen2.5-VL-7B	**TableLVR-41K-LoRA**	57.73	38.60	41.43	81.21	76.99	60.06	64.06
	DLVR *(ours)*	**59.94** ↑	36.40	43.15 ↑	**89.04** ↑	78.35 ↑	72.30 ↑	76.88 ↑
Qwen3-VL-8B	**TableLVR-41K-LoRA**	58.84	**40.16**	44.16	73.78	**78.58**	74.17	78.84
	DLVR *(ours)*	59.35 ↑	38.08	**44.61** ↑	85.91 ↑	78.51	**74.78** ↑	**83.78** ↑

Table-GLS, DLVR achieves stronger overall performance without exposing intermediate reasoning traces or performing external image operations, particularly on OOD benchmarks. This indicates that the learned latent representations can generalize beyond the training data sources. (V) The consistent gains across Qwen2.5-VL and Qwen3-VL further demonstrate that DLVR can be effectively transferred across different LVLM backbones. (VI) The relatively weaker performance on TAT-QA reflects its arithmetic-intensive financial questions, which require accurate numerical operations and answer normalization beyond evidence localization.

Ablation Study. We further evaluate the main components of DLVR. As shown in Table 2, **removing LI-GRPO** and retaining only the first-stage SFT (i.e., *w/o LI-GRPO*) generally degrades performance, showing that reinforcement learning further improves the learned latent reasoning process. **Removing structure-content decoupling** (i.e., *w/o Decoupling*) causes substantial degradation across the representative benchmarks. **Removing** $\mathcal{L}_{\text{con}}$ (i.e., *w/o* $\mathcal{L}_{\text{con}}$) similarly weakens performance, particularly on InfoTabs, confirming the importance of explicitly grounding content states. **The orthogonality objective** provides complementary gains by reducing redundancy between structural and content representations. Finally, **removing LCR** (i.e., *w/o LCR*) also lowers overall performance, indicating that intervention-based optimization encourages the generated latent states to support answer prediction. Complete results are presented in Appendix C. Moreover, we compare the default order-flexible decoding with two forced reasoning orders during inference. Their nearly identical performance indicates that DLVR is robust to the ordering of the row and column stages and that its improvements do not depend on selecting a particular reasoning order. Additional results and analysis are provided in Appendix D.

Table 2: Ablation study on representative benchmarks.

Method	Question Answering			Fact Verification	
	WTQ	HiTab	AIT-QA	TabFact	InfoTabs
DLVR (ours)	**59.94**	**43.15**	**89.04**	**78.35**	**72.30**
w/o LI-GRPO	56.93	42.64	87.28	78.12	72.22
w/o Decoupling	56.84	42.39	85.32	78.22	66.67
w/o $\mathcal{L}_{\text{orth}}$	59.00	42.07	86.50	77.82	69.63
w/o $\mathcal{L}_{\text{con}}$	59.07	42.77	87.08	77.82	66.44
w/o LCR	59.69	42.89	88.65	77.93	71.93

Sensitivity to Latent Block Length. Figure 3 studies the sensitivity of DLVR to the latent block length K. Increasing K from 2 to 4 improves all three representative datasets and the overall aver-

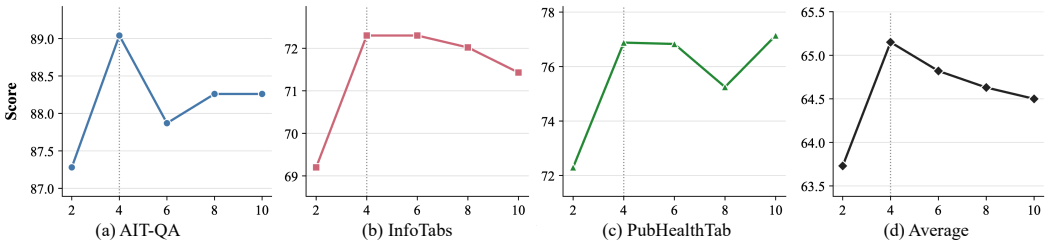


Figure 3: Results under different latent block lengths K. Each stage contains K structural and K content states, and the dotted line indicates the configuration used in our main experiments.

Table 3: Comparison of inference efficiency, where Time and Tokens denote the average inference time and generated tokens per sample, respectively.

Method	InfoTabs		AIT-QA		PubHealthTab	
	Time (s) ↓	Tokens ↓	Time (s) ↓	Tokens ↓	Time (s) ↓	Tokens ↓
CoT	3.69	131.41	5.90	200.40	3.13	111.62
DiSCo+CoT	2.56	88.97	5.59	190.13	2.50	88.26
Table-GLS	10.28	358.82	10.24	331.45	10.90	379.15
DiSCo+Table-GLS	12.80	450.07	9.87	352.46	12.65	447.77
DLVR (ours)	**1.64**	**38.35**	**2.07**	**43.68**	**1.65**	**38.47**

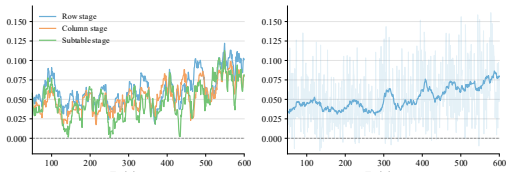


Figure 4: Training dynamics of the aggregate and stage-wise LCR during LI-GRPO.

age, indicating that excessively short blocks may provide insufficient capacity for latent reasoning. Further increasing K brings no consistent benefit. AIT-QA declines after $K = 4$, InfoTabs becomes saturated, and PubHealthTab exhibits a non-monotonic trend with a later recovery at $K = 10$. Nevertheless, the average performance gradually decreases beyond $K = 4$, suggesting that longer trajectories may introduce redundant computation without improving overall reasoning. We therefore adopt $K = 4$, corresponding to eight latent states per stage. The complete results and analysis are presented in Appendix E

Generation Efficiency. Table 3 compares the inference efficiency of DLVR with explicit reasoning methods using Qwen3-VL-8B under the Transformers library. DLVR requires only 1.79 seconds and 40.17 generated tokens per sample on average, including its latent states, control tokens, and final answer. Compared with CoT and DiSCo+CoT, DLVR reduces the number of generated tokens by 72.8% and 67.2%, and inference time by 57.9% and 49.7%, respectively. The gains are more pronounced over Table-GLS and DiSCo+Table-GLS, reducing generation length by 88.7–90.4% and inference time by 82.9–84.8%. These results show that internalizing intermediate table reasoning into compact continuous states substantially reduces both generation and runtime overhead.

Analysis of LCR. Figure 4 presents the evolution of LCR during LI-GRPO. Despite step-level variance caused by stochastic rollouts and randomly sampled latent interventions, the smoothed aggregate LCR exhibits a clear upward trend throughout training. The row, column, and subtable rewards show similar improvements, suggesting that LI-GRPO progressively strengthens the predictive contribution of latent components across the complete reasoning trajectory rather than concentrating on a single stage.

6 Conclusion

In this paper, we present **Decoupled Latent Visual Reasoning** (DLVR), which organizes latent table reasoning into progressive row, column, and subtable stages while separating structural and semantic information. Supported by **TableLVR-41K**, DLVR combines decoupled latent supervised fine-tuning with **Latent Intervention GRPO** and the **Latent Contribution Reward**. Experiments across seven benchmarks and two LVLM backbones demonstrate consistent improvements over direct, explicit, and generic latent reasoning methods. Ablations further validate the complementary contributions of decoupled supervision and intervention-based optimization, highlighting structured latent computation as an effective paradigm for multimodal table reasoning.

AI USE STATEMENT

Generative AI tools were used for synthetic data generation and manuscript preparation. Specifically, GPT-5.4-mini served as a vision-language teacher to generate questions, answers, and supporting subtables for TableLVR-41K, as described in Appendix A. The generated samples were processed using completeness, formatting, and evidence-alignment checks before inclusion. OpenAI ChatGPT was also used to assist with language editing and presentation refinement. Generative AI tools were not used for other research tasks requiring disclosure under the ICLR policy. All AI-assisted content was reviewed and revised by the authors, who take full responsibility for the final manuscript and its scientific claims.

REFERENCES

Mubashara Akhtar, Oana Cocarascu, and Elena Simperl. PubHealthTab: A public health table-based dataset for evidence-based fact checking. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 1–16, 2022.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Ke qin Chen, Xue-Jing Liu, Jia-Lin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jian-Qiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*, 2020.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3697–3711, 2021.

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

Naihao Deng, Zhenjie Sun, Ruiqi He, Aman Sikka, Yulong Chen, Lin Ma, Yue Zhang, and Rada Mihalcea. Tables as texts or images: Evaluating the table reasoning ability of LLMs and MLLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 407–426, August 2024.

Shuai Dong, Siyuan Wang, Xingyu Liu, Chenglin Li, Haowen Hou, and Zhongyu Wei. Interleaved latent visual reasoning with selective perceptual modeling. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29316–29335, July 2026.

Xingyu Fu, Minqian Liu, Zhengyuan Yang, John Richard Corring, Yijuan Lu, Jianwei Yang, Dan Roth, Dinei Florencio, and Cha Zhang. ReFocus: Visual editing as a chain of thought for structured image understanding. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 17783–17805, 13–19 Jul 2025.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. INFOTABS: Inference on tables as semi-structured data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2309–2324, 2020.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason E Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Itxz7S4Ip3.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Houcheng Jiang, Jiajun Fu, Junfeng Fang, Chen Gao, Xiang Wang, Xiangnan He, and Yong Li. Univlr: Unifying text and vision in visual latent reasoning for multimodal llms. *arXiv preprint arXiv:2605.11856*, 2026.

Jun-Peng Jiang, Yu Xia, Hai-Long Sun, Shiyin Lu, Qingguo Chen, Weihua Luo, Kaifu Zhang, De-Chuan Zhan, and Han-Jia Ye. Multimodal tabular reasoning with privileged structured information. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 106877–106903, 2025a.

Jun-Peng Jiang, Tao Zhou, De-Chuan Zhan, and Han-Jia Ye. Compositional condition question answering in tabular understanding. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 27831–27850, 13–19 Jul 2025b.

Haoqiang Kang, Yinpeng Chen, Luyang Liu, Jesper Sparre Andersen, Abhijit Ogale, Baochen Sun, Lichan Hong, and Ed H Chi. Scaffolding minds: Optimizing latent visual target representations for multimodal reasoning. *arXiv preprint arXiv:2608.19669*, 2026a.

Jiaxuan Kang, Siyu Chen, Mingda Li, Mingjie Liu, Tianyue Wang, Zhaoyang Wei, Yongheng Zhang, Yanchao Hao, and Zheng Wei. Lut: Latent utility training for visual reasoning. *arXiv preprint arXiv:2608.00743*, 2026b.

Xiaoqiang Kang, Shengen Wu, Zimu Wang, Yilin Liu, Xiaobo Jin, Kaizhu Huang, Wei Wang, Yutao Yue, Xiaowei Huang, and Qiufeng Wang. Can GRPO boost complex multimodal table understanding? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12631–12644, November 2025.

Yannis Katsis, Saneem Chemmengath, Vishwajeet Kumar, Samarth Bharadwaj, Mustafa Canim, Michael Glass, Alfio Gliozzo, Feifei Pan, Jaydeep Sen, Karthik Sankaranarayanan, et al. Ait-qa: Question answering dataset over complex tables in the airline industry. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pp. 305–314, 2022.

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 148076–148090, 2026a.

Kelvin Li, Chuyi Shang, Leonid Karlinsky, Rogerio Feris, Trevor Darrell, and Roei Herzig. Latent implicit visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33457–33466, 2026b.

Chaohu Liu, Haoyu Cao, YongXiang Hua, and Linli Xu. Multimodal table understanding with difficulty-aware reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 755–763, 2026a.

Chaohu Liu, Shida Wang, Yubo Wang, and Linli Xu. Tablemix: Enhancing multimodal table reasoning in mllms from a data-centric perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 33553–33565, June 2026b.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=DHyHRBwJUTN.

Nafise Sadat Moosavi, Andreas Rücklé, Dan Roth, and Iryna Gurevych. Learning to reason for text generation from scientific tables. *arXiv preprint arXiv:2104.08296*, 2021.

Van-Quang Nguyen and Takayuki Okatani. CoReTab: Improving multimodal table understanding with code-driven reasoning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6498–6523, March 2026.

Panupong Pasupat and Percy Liang. 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)*, pp. 1470–1480, July 2015.

Brandon Smock, Rohith Pesala, and Robin Abraham. Pubtables-1m: Towards comprehensive table extraction from unstructured documents. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4624–4632. IEEE, 2022.

Prasham Yatinkumar Titiya, Jainil Trivedi, Chitta Baral, and Vivek Gupta. MMTabReal: Real-world benchmark for multimodal table understanding. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 41156–41176, July 2026.

Haolan Wang, Zhenghao Liu, Xinze Li, Xiaocui Yang, Yu Gu, Yukun Yan, Qi Shi, Fangfang Li, Chong Chen, and Ge Yu. Hippo: Enhancing the table understanding capability of llms through hybrid-modal preference optimization. In *International Conference on Database Systems for Advanced Applications*, pp. 203–218. Springer, 2026a.

Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in latent visual space beyond image and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12030–12040, 2026b.

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

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33510–33520, 2026.

Kun-Yang Yu, Zhi Zhou, Shi-Yu Tian, Xiao-Wen Yang, Zi-Yi Jia, Ming Yang, Zi-Jian Cheng, Lan-Zhe Guo, and Yu-Feng Li. Thinking with tables: Enhancing multi-modal tabular understanding via neuro-symbolic reasoning. *arXiv preprint arXiv:2603.24004*, 2026.

Weichao Zhao, Hao Feng, Qi Liu, Jingqun Tang, Shu Wei, Binghong Wu, Lei Liao, Yongjie Ye, Hao Liu, Wengang Zhou, et al. Tabpedia: Towards comprehensive visual table understanding with concept synergy. *Advances in Neural Information Processing Systems*, 37:7185–7212, 2024.

Mingyu Zheng, Xinwei Feng, Qingyi Si, Qiaoqiao She, Zheng Lin, Wenbin Jiang, and Weiping Wang. Multimodal table understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9102–9124, August 2024.

Bangbang Zhou, Zuan Gao, Zixiao Wang, Boqiang Zhang, Yuxin Wang, Zhineng Chen, and Hongtao Xie. Syntab-llava: Enhancing multimodal table understanding with decoupled synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24796–24806, June 2025.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. 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)*, pp. 3277–3287, 2021.

Yingjie Zhu, Xuefeng Bai, Kehai Chen, Yang Xiang, Youcheng Pan, Xiaoqiang Zhou, and Min Zhang. Decoupling skeleton and flesh: Efficient multimodal table reasoning with disentangled alignment and structure-aware guidance. In *Forty-third International Conference on Machine Learning*, 2026.

A Structure-Guided Supervision Construction

A.1 Source Tables and Image Rendering

We construct our training corpus from the training and development splits of InfoTabs (Gupta et al., 2020), TabFact (Chen et al., 2020), TabMWP (Lu et al., 2023), TAT-QA (Zhu et al., 2021), and WikiTableQuestions (WTQ) (Pasupat & Liang, 2015). Their original test splits are excluded from data construction and reserved for evaluation through MMTab. Since these datasets primarily provide structured or serialized tables, we render each table as a PNG image while preserving its textual content, row and column organization, headers, and merged cells. To increase visual diversity, we randomly sample color palettes and use the resulting colors to fill table cells during rendering. We additionally retain the normalized table structure and cell bounding boxes for subsequent evidence alignment.

A.2 Question and Subtable Generation

We use GPT-5.4-mini as a vision-language teacher to synthesize reasoning examples directly from the rendered table images. The decoding temperature is set to zero. For each table, the teacher generates samples spanning six types of table understanding and reasoning tasks, including cell lookup, conditional selection, comparison, arithmetic reasoning, and ranking or extreme-value identification and fact verification.

Each generated example contains a question, a short deterministic answer, and a minimal key subtable that is sufficient to derive the answer. The key subtable is serialized using vertical bars to separate columns and line breaks to separate rows. The prompt requires every question to be answerable solely from the table image and instructs the teacher to return null when a valid example cannot be generated. The complete generation prompts and output schema are as follows:

Prompt for Training Sample Generation

```
You are generating multimodal table reasoning data from a table image.

Generate exactly 6 examples, one for each type:
1. cell_lookup
2. conditional_selection
3. comparison
4. arithmetic_reasoning
5. ranking_extreme
6. fact_verification

For every valid example:
- The question must be answerable only from the table image.
- Prefer questions involving multiple rows and multiple columns when possible.
- The key_subtable must be minimal but sufficient to answer the question.
- The key_subtable may include the table title if needed.
- Sub-table format: columns separated by "|"; rows separated by "\\n".
- Avoid outside knowledge, vague questions, and visual-style questions.
- Use short deterministic answers.
- The answer must always be a list of strings, e.g. ["120"] or ["entailment"].
- For fact_verification, the question field must contain only the statement to verify, with no
     instruction suffix.
- For fact_verification, generate exactly one entailment, one contradiction, and one neutral
     example.
- Use neutral only when the table does not contain enough information.

If the table is not suitable for one type, return null for question, answer, and key_subtable of
     that type.
If the table is not suitable for a fact_verification label, still include that slot but return
     null for question, answer, and key_subtable.

Output valid JSON only, using this schema:
{
  "examples": [
    {
      "type": "cell_lookup",
      "question": "..." or null,
      "answer": ["..."] or null,
      "key_subtable": "Title if needed\\nColumn A|Column B\\nvalue A|value B" or null
    },
    {
      "type": "conditional_selection",
      "question": "..." or null,
      "answer": ["..."] or null,
      "key_subtable": "..." or null
```

```
    },
    {
      "type": "comparison",
      "question": "..." or null,
      "answer": ["..."] or null,
      "key_subtable": "..." or null
    },
    {
      "type": "arithmetic_reasoning",
      "question": "..." or null,
      "answer": ["..."] or null,
      "key_subtable": "..." or null
    },
    {
      "type": "ranking_extreme",
      "question": "..." or null,
      "answer": ["..."] or null,
      "key_subtable": "..." or null
    },
    {
      "type": "fact_verification",
      "question": "entailment statement only" or null,
      "answer": ["entailment"] or null,
      "key_subtable": "..." or null
    },
    {
      "type": "fact_verification",
      "question": "contradiction statement only" or null,
      "answer": ["contradiction"] or null,
      "key_subtable": "..." or null
    },
    {
      "type": "fact_verification",
      "question": "neutral statement only" or null,
      "answer": ["neutral"] or null,
      "key_subtable": "..." or null
    }
  ]
}
```

A.3 Stage-Wise Evidence Construction

The generated key subtable is aligned with the underlying structured table to identify the supporting row set $\mathcal{R}$ and column set $\mathcal{C}$. Based on these supports, we construct three complementary evidence views. The row-oriented view highlights the question-relevant rows together with their associated headers, while the column-oriented view highlights the relevant columns and their header associations. The subtable view retains the compact intersection of the selected rows and columns, containing the evidence required to derive the answer.

For each evidence view, we produce both an image $\mathbf{X}_m$ and a serialization $\mathbf{C}_m$ of its retained cells. The image provides the visual reconstruction target, whereas the serialization provides explicit content supervision. Each example is randomly assigned either a row-first or column-first route. The corresponding row and column views are placed in the first two stages according to this assignment, while the compact subtable is always used as the third stage.

A.4 Filtering and Statistics

We discard outputs with missing questions, answers, or key subtables, as well as examples whose key subtables cannot be aligned with the source table. Neutral fact-verification examples are also removed because their answers are not grounded in explicit supporting cells. Answer formats are normalized across source datasets before conversion to the final SFT format.

The resulting corpus contains 41,567 examples, including 20,810 row-first and 20,757 column-first trajectories. Table 4 reports the joint distribution across source datasets and reasoning tasks.

A.5 LI-GRPO Data

We randomly sample 10,000 examples from the SFT corpus for LI-GRPO training. For this stage, the intermediate evidence images and serialized contents are removed, leaving the original table image, question, answer, and route metadata. Consequently, LI-GRPO does not access the stage-specific supervision used during SFT and optimizes the generated latent trajectories solely through outcome rewards and LCR.

Table 4: Distribution of the constructed SFT corpus across source datasets and reasoning tasks. Select., Arith., Rank., Ver.+, and Ver.− denote conditional selection, arithmetic reasoning, ranking or extreme-value identification, positive verification, and negative verification, respectively.

Source	Lookup	Select.	Compare	Arith.	Rank.	Ver.+	Ver.−	Total
InfoTabs	1,882	1,578	1,707	1,674	1,489	945	874	10,149
TabFact	1,860	1,510	1,737	1,652	1,497	937	906	10,099
TabMWP	1,614	1,571	1,567	1,551	1,564	809	782	9,458
TAT-QA	325	263	340	359	302	202	192	1,983
WikiTableQuestions	1,825	1,462	1,677	1,658	1,448	940	868	9,878
Total	7,506	6,384	7,028	6,894	6,300	3,833	3,622	41,567

Table 5: The statistics of the seven evaluation benchmarks.

Statistic	Question Answering				Fact Verification			Total
	WTQ	HiTab	TAT-QA	AIT-QA	TabFact	InfoTabs	PubHealthTab	
Samples	4,344	1,576	772	511	6,845	5,400	1,942	21,390
Tables	421	535	231	111	1,045	600	362	3,305

B Experimental Setup

B.1 Datasets

Table 5 reports the number of evaluation samples in the seven MMTab benchmarks used in our experiments.

B.2 Baselines

We train two LVR variants (Li et al., 2026a) on the 41,567 examples in TableLVR-41K using Qwen2.5-VL-7B. Both use the original table image and question as input and jointly optimize answer generation and latent visual reconstruction during SFT. They differ in the reconstruction target, where *LVR-Random* uses visual features associated with a randomly sampled bounding box in the input table, and *LVR-Subtable* uses the visual representation of the complete third-stage subtable image. Each SFT model is subsequently optimized with GRPO on the same 10,000-example subset used for DLVR, using answer-correctness and format rewards without the Latent Contribution Reward.

B.3 Detailed Implementation Settings

We implement DLVR using Qwen2.5-VL-7B-Instruct and Qwen3-VL-8B-Instruct. For both backbones, we freeze the vision encoder and multimodal projector and apply LoRA to the linear layers of the language model. During SFT, the token embeddings and language modeling head are also optimized to learn the newly introduced control tokens. Each structural or content block contains $K = 4$ continuous latent states, allocated by repeating its component control token K times. Table 7 maps the notation introduced in Section 3.1 to the corresponding tokens used in our implementation. We use AdamW with a cosine learning-rate schedule, a warmup ratio of 0.03, weight decay of 0.1, and a maximum gradient norm of 1.0. Training uses bfloat16 precision, gradient checkpointing, and DeepSpeed ZeRO-2.

Table 6 summarizes the model-specific settings. Both SFT models are trained for one epoch on the complete TableLVR-41K corpus. LI-GRPO is initialized from the corresponding merged SFT checkpoint and trained for one epoch on the 10K subset. We use correctness and format reward weights of 1.0 and 0.1, respectively, and disable the other auxiliary rewards. LCR is computed at every update using control-token embedding interventions. The KL coefficient is set to zero in the final experiments. During LI-GRPO and inference, the decoder is allowed to realize either of the two valid reasoning orders according to the next-token scores of the corresponding stage-control tokens,

Table 6: Detailed training configurations for DLVR.

Configuration	**Qwen2.5-VL-7B-Instruct**		**Qwen3-VL-8B-Instruct**	
Stage	**SFT**	**LI-GRPO**	**SFT**	**LI-GRPO**
Training instances	41,567	10,000	41,567	10,000
Epochs	1	1	1	1
Learning rate	5×10^{-5}	5×10^{-6}	5×10^{-5}	5×10^{-6}
Batch size per device	1	8	1	8
Gradient accumulation	8	1	8	1
LoRA rank	8	8	8	16
LoRA scaling factor	16	16	16	32
LoRA dropout	0	0	0	0
Latent states per block	4	4	4	4
λ_{con}	0.01	–	0.1	–
λ_{fus}	0.1	–	0.5	–
λ_{orth}	0.01	–	0.01	–
Number of rollouts	–	4	–	4
Sampling temperature	–	0.9	–	2.8
Maximum prompt length	–	8,192	–	8,192
Maximum completion length	–	512	–	128
LCR weight	–	0.2	–	0.1
KL coefficient	–	0	–	0

Table 7: Control tokens used in DLVR and their corresponding mathematical notation.

Role	Notation	Implementation
Trajectory start	τ_{start}	<|lvr_start|>
Trajectory end	τ_{end}	<|lvr_end|>
Row structure	$\tau_{\text{row}}^{\text{str}}$	<|row_struct_lvr|>
Row content	$\tau_{\text{row}}^{\text{con}}$	<|row_content_lvr|>
Column structure	$\tau_{\text{column}}^{\text{str}}$	<|col_struct_lvr|>
Column content	$\tau_{\text{column}}^{\text{con}}$	<|col_content_lvr|>
Subtable structure	$\tau_{\text{sub}}^{\text{str}}$	<|sub_struct_lvr|>
Subtable content	$\tau_{\text{sub}}^{\text{con}}$	<|sub_content_lvr|>
Row termination	$\tau_{\text{row}}^{\text{end}}$	<|row_lvr_latent_end|>
Column termination	$\tau_{\text{column}}^{\text{end}}$	<|col_lvr_latent_end|>
Subtable termination	$\tau_{\text{sub}}^{\text{end}}$	<|sub_lvr_latent_end|>

while the row, column, and subtable boundaries are enforced through their respective control tokens. This order-flexible decoding is used to remain consistent with the order augmentation introduced during SFT and is not interpreted as task-specific route optimization. The maximum number of visual tokens is set to 5,120. No stage-specific evidence image, serialized content, or external image operation is used at inference time. All experiments are conducted using 8 NVIDIA A100 GPUs with 80G memory.

C Ablation Study

Table 8 examines the contribution of each component of DLVR. Removing LI-GRPO and retaining only decoupled latent SFT (i.e., *w/o LI-GRPO*) reduces the average score from 65.15 to 64.32, showing that reinforcement learning further improves the learned latent reasoning process. Removing LCR while retaining GRPO (i.e., *w/o LCR*) also causes a consistent average degradation, indicating that directly rewarding the predictive contribution of latent states is more effective than outcome-based optimization alone. The structure-content decomposition is particularly important. Removing it (i.e., *w/o Decoupling*) produces the largest overall decline, with pronounced degradation on AIT-QA, InfoTabs, and PubHealthTab. Removing the content alignment loss (i.e., *w/o* $\mathcal{L}_{\text{con}}$) results in a comparable drop, especially on the two fact-verification datasets, demonstrating the importance of explicitly grounding content states in relevant cell semantics. The orthogonality loss

Table 8: Complete ablation results on Qwen2.5-VL-7B. The best result in each column is bolded.

	Question Answering				Fact Verification		
Method	WTQ	TAT-QA	HiTab	AIT-QA	TabFact	InfoTabs	PubHealthTab
DLVR (ours)	**59.94**	36.40	**43.15**	**89.04**	**78.35**	**72.30**	76.88
w/o LI-GRPO	56.93	**37.56**	42.64	87.28	78.12	72.22	75.49
w/o Decoupling	56.84	36.79	42.39	85.32	78.22	66.67	68.90
w/o $\mathcal{L}_{\text{orth}}$	59.00	36.53	42.07	86.50	77.82	69.63	**79.20**
w/o $\mathcal{L}_{\text{con}}$	59.07	36.27	42.77	87.08	77.82	66.44	66.74
w/o LCR	59.69	37.18	42.89	88.65	77.93	71.93	75.54

Table 9: Robustness of DLVR to different reasoning orders. Default order-flexible decoding is compared with forced row-first and column-first execution.

	Question Answering				Fact Verification			Avg.
Route	WTQ	TAT-QA	HiTab	AIT-QA	TabFact	InfoTabs	PubHealthTab	
Order-flexible	59.94	36.40	43.15	89.04	**78.35**	**72.30**	**76.88**	65.15
Row-first	59.92	36.40	43.15	89.04	78.16	**72.30**	76.73	65.10
Column-first	**59.97**	36.40	**43.34**	89.04	78.25	72.26	76.83	**65.16**

yields a smaller but consistent overall benefit, suggesting that reducing redundancy between structural and content states improves their complementarity. Although individual ablations occasionally perform better on TAT-QA or PubHealthTab, the complete model achieves the highest average score and the strongest performance on five of the seven datasets, validating the overall design of DLVR

D Robustness to Reasoning Order

Table 9 evaluates the robustness of DLVR to different execution orders. We compare the default order-flexible decoding with two settings in which the row-first or column-first order is explicitly enforced. All three settings achieve nearly identical performance, with average scores differing by at most 0.06 points. This result indicates that randomizing the reasoning order during training prevents the model from becoming dependent on a particular sequence of row and column processing. Since both orders ultimately encode the same complementary evidence before subtable integration, DLVR remains stable regardless of which stage is executed first. The default decoding therefore provides order flexibility rather than attempting to identify an optimal question-specific route.

E Sensitivity to Latent Block Length

Table 10: Performance under different latent block lengths K. Each reasoning stage contains K structural and K content states, resulting in $2K$ latent states per stage. The best result in each column is bolded.

	Question Answering				Fact Verification			Avg.
K	WTQ	TAT-QA	HiTab	AIT-QA	TabFact	InfoTabs	PubHealthTab	
2	59.44	**36.92**	42.83	87.28	78.17	69.20	72.30	63.73
4	**59.94**	36.40	**43.15**	**89.04**	**78.35**	**72.30**	76.88	**65.15**
6	59.02	36.66	42.70	87.87	78.33	**72.30**	76.83	64.82
8	59.32	36.66	42.89	88.26	78.01	72.02	75.25	64.63
10	58.45	36.01	42.20	88.26	78.00	71.43	**77.14**	64.50

Table 10 reports the complete results under different latent block lengths. Increasing K from 2 to 4 improves the average score by 1.42 points, with gains on six of the seven benchmarks. The $K = 4$ configuration achieves the best average performance and the strongest results on most individual

datasets, indicating that a moderate latent capacity is sufficient to encode the required structural and semantic evidence. Increasing K further does not produce consistent improvements. Although $K = 6$ matches the best InfoTabs result and $K = 10$ performs best on PubHealthTab, performance on several other benchmarks gradually declines. Longer blocks also increase recurrent computation linearly and may introduce redundant latent states or greater optimization difficulty. We therefore adopt $K = 4$, yielding eight latent states per stage, as the default configuration.