

Twin-T & TwintVQA: A Reliable Structure–Detail Separating VLM and a Comprehensive Benchmark for Chart and Table Tasks

Jiahua Bao^{1,2,3}, Siyao Cheng^{1,2,3*}, Jiaxing Du¹, Qingtao Xia¹, Changjian He¹, Zeming Lang¹, Jie Liu^{1,2,3}

¹Harbin Institute of Technology, China

²National Key Laboratory of Smart Farming Technology and Systems, China

³China Mobile 5G Institute, China

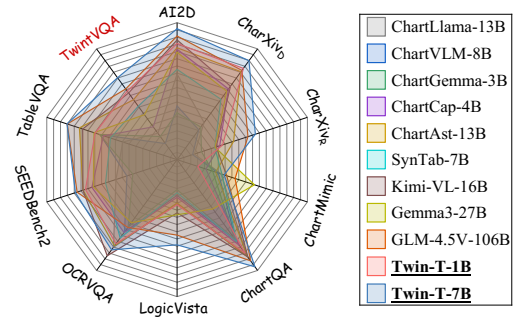
jhbao@stu.hit.edu.cn, csy@hit.edu.cn

Abstract

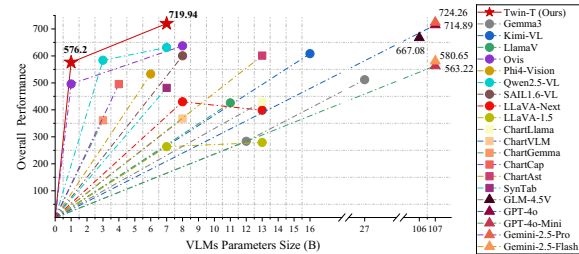
With the rapid development of Vision-Language Models (VLMs), there is a growing demand for automatic analysis of structured visual data. Charts and tables carry quantitative information through regular layouts, explicit numbers, and chart-specific reading patterns, yet current VLMs still underuse these properties, often causing value errors and unreliable analysis. To overcome these limitations, we propose **Twin-T**, a two-stage expert VLM for comprehensive chart–table tasks across Image, LaTeX, and Python. In stage 1, we propose a dual-head image encoder that can separate structural cues and fine details from input images. In stage 2, we propose MINT, a preference learning method that emphasizes numerical and keyword fidelity, as well as vision-text matching. Furthermore, we introduce a comprehensive TwintVQA benchmark with 17 chart types, 11 task types, 3 data formats, and short / medium / long QA settings. Our model narrows the gap between open-source and closed-source models on mainstream chart–table benchmarks, outperforming open-source models while even remaining competitive with GPT-4o and Gemini-2.5-Pro. More details are available at the <https://github.com/Samsara-1999/Twin-T-TwintVQA>.

1. Introduction

Vision–Language Models (VLMs) have shown steady gains on many visual downstream tasks [4, 6, 23, 36, 37, 49]. As practical use grows, the need to automatically analyze structured visuals such as charts and tables is rising [7, 46]. In real settings, charts and tables are primary carriers of quantitative information. Their understanding depends on two complementary types of signals: **structure**, such as axes, grids, headers, and layouts, and **details**, such as values, leg-



(a) Comparison across public chart–table benchmarks. Red labels mark our proposed **TwintVQA** benchmark, we compare chart–table expert VLMs and mainstream general-purpose VLMs.



(b) Overall chart–table performance vs. model size. “★” represents our Twin-T model; “●” represents general-purpose VLMs; “■” represents chart–table expert VLMs; “▲” represents API models.

Figure 1. Our **Twin-T (1B/7B)** VLM outperforms other models on chart–table tasks. It surpasses **GLM-4.5V-106B** and is even competitive with **GPT-4o** and **Gemini-2.5**. Zoom in for more details.

ends, colors, and local marks. In human analysis, structural cues often serve as a global scaffold for interpreting detailed evidence. Inspired by this functional role, these raise a central question: *how can we enable VLMs to jointly capture structure and details to generate reliable chart-table analysis like humans?*

However, existing general-purpose VLMs and chart–table expert models still show clear gaps [24, 28]. First,

*Corresponding author.

most methods rely on generic visual pathways and let a single encoder implicitly mix structural cues and fine-grained details. This design lacks an explicit chart-table inductive bias and makes it harder to align global layouts with local values, legends, and text regions. Existing methods add auxiliary tokens, linear layers, or routing modules [15, 41], while others insert fusion blocks across visual layers [29]. However, these designs still do not explicitly separate and control the interaction between chart-table structure and detail cues. Second, real-world chart-table data contain rich numerical information, but existing models are often not sufficiently sensitive to numbers. They mainly focus on visual content, yet often struggle to read values accurately, which reduces reliability in practical settings [8, 14, 50].

To overcome these limitations, we propose **Twin-T**, an expert VLM for charts and tables QA. We train Twin-T in two stages for more reliable chart-table understanding and generation. Stage 1 focuses on visual enhancement by introducing a dual-head image encoder: one head captures structural information, and the other captures fine details. We fuse the two image embeddings using a Schur-style update [32]. The fused visual tokens are then fed into the text decoder via supervised instruction tuning, which aligns the additional image encoder and original text decoder to chart and table distributions. Stage 2 further focuses on high-quality text generation, which employs **MINT** (Modality-Integrated Numeric and Text reinforcement for reliable generation), a preference learning method that improves numeric fidelity, encourages low-entropy distribution on output tokens logits, and increases text-vision matching level, leading to more reliable generation.

Furthermore, we build the TwintVQA benchmark to evaluate VLMs across multiple tasks and data formats. Our benchmark covers 11 tasks and 17 chart types, supports Image, Python, and LaTeX formats, and includes short, medium, and long QA settings. We release Twin-T-1B and Twin-T-7B to advance community research on VLMs across multiple parameter scales, as shown in Fig. 1.

Our contributions are as follows:

- We propose Twin-T, a two-stage trained chart-table VLM with a dual-head image encoder and MINT preference learning that separates structural and detailed information and generates reliable numbers and text.
- We release the TwintVQA benchmark that spans 11 tasks and 17 chart types, with 3 chart-table formats (Image, Python, LaTeX) and short / medium / long QA lengths.
- Twin-T shows leading performance on mainstream chart-table benchmarks, outperforming open-source VLMs and surpassing GLM-4.5V-106B, while remaining competitive with GPT-4o and Gemini-2.5. We release 1B and 7B models to advance community research on VLMs.

2. Related Works

General-Purpose VLMs: Most general-purpose VLMs connect a visual encoder to a text decoder through a projection connector and are typically trained with large image-text pairs followed by instruction tuning for conversational reasoning [5, 49]. LLaVA [21] established the standard recipe of visual instruction tuning, where GPT-4V [17] generates multimodal instructions that are used to align a CLIP [34] image encoder with an Large Language Model (LLM) for open-ended QA, enabling strong zero-shot performance across tasks. The Qwen-VL series [3] extends this architecture with a dedicated image receptor, and a three stage training pipeline on multilingual data. Ovis [26] focuses on structural alignment between visual and textual embeddings, adds preference-tuned reasoning, and supports large image inputs, reflecting a move toward compact but competitive VLMs. Despite competitive performance results across many domains and tasks, general-purpose VLMs can still fall short in specific domains compared with expert models [22], often due to limited domain knowledge and insufficient answer granularity [35].

Charts and Tables Expert VLMs: As the demand for automated data analysis grows, charts and tables have become central representations. Expert models for charts and tables are advancing rapidly [7]. ChartLLama [15] builds on LLaVA [21] and adds large-scale chart-table pretraining in captioning and structure tasks, which improves chart-table performance. ChartVLM [41] couples a versatile benchmark ChartX with a foundation model oriented to interpretable, chart-specific reasoning across multiple types and tasks, showing competitive results against both general and chart-specialized baselines. ChartAst [29] follows a two-stage recipe that pre-trains “chart to table” parsing to align visual patterns with symbolic structure, then performs multitask instruction tuning for zero-shot chart understanding. Despite these advances, expert models still show two clear gaps: (1) Most VLMs do not explicitly integrate chart-table specific structure into training, instead relying on “one size fits all” recipes with little task-aware supervision [13, 47]. (2) Existing benchmarks also cover too few task and chart types, and they emphasize short answers, so that cannot fully evaluate models performance [19, 27].

3. Methods

We propose Twin-T (1B/7B), a chart-table expert VLM built on Ovis2-1B and Qwen2.5-VL-7B with a dual-head image encoder. Our training has two stages: Stage 1 separates and fuses structural and detail embeddings, and aligns the fused visual embeddings with text decoder. Stage 2 applies our MINT, a novel preference learning method that focuses on generation precision.

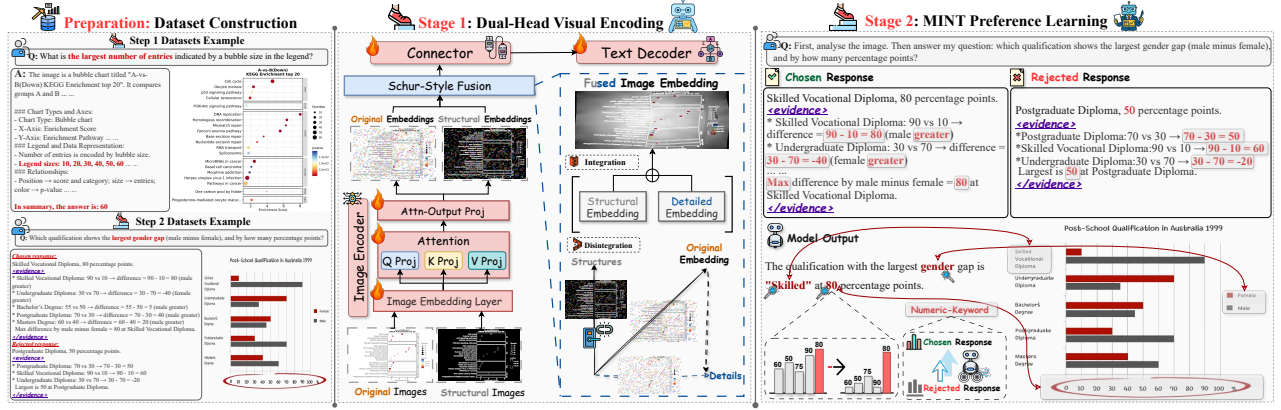


Figure 2. Workflow of our **Twin-T** VLM for chart-table QA. From left to right we show dataset examples used for Stage 1 and Stage 2 training. **Stage 1** builds a novel dual-head visual encoder: besides the original image, a structural image is passed through the image encoder; a parameter-free Schur-style module inserted between the image encoder and the connector, softly gates and removes structural directions to obtain a image details embedding, then fuses structural and detailed image embedding into fused visual embedding. This stage adapts the whole VLM to the novel dual-head pathway and fusion. **Stage 2** improves generation precision via preference learning on chosen vs. rejected responses: it upweights numeric and comparator tokens, encourages low-entropy logits on number tokens, and enlarges text-vision alignment (red curve). Zoom in for more details.

3.1. Stage 1: Dual-Head Visual Encoding

Our Motivation: We draw inspiration from how humans read charts and tables. For example, when viewing a pie chart, the first impression is the proportion among parts, only then do we focus on colors or specific numbers to complete tasks. We want VLMs to similarly separate and then integrate *structures* and *details* from the input images.

We propose a dual-head visual encoding and fusion method to capture structural information and details: edges, straight lines, and frames, tends to yield greater pixel magnitudes in the input image. To quickly capture structural (high-frequency) cues, we compute a structure image from the raw image employing the Canny method [10], then feed both the raw image and the structure image into the image encoder separately, using the shared and trainable image encoder to obtain a structural embedding $E_{\text{Stru}} \in \mathbb{R}^{B \times N_{\text{vis}} \times D_{\text{vis}}}$ and an image embedding $E_{\text{Img}} \in \mathbb{R}^{B \times N_{\text{vis}} \times D_{\text{vis}}}$, where B is the batch size, N_{vis} the number of visual tokens, and D_{vis} the token embedding dimension. Note that E_{Img} aggregates both image structures and details information. To separate E_{Stru} from E_{Img} and obtain a details embedding $E_{\text{Det}} \in \mathbb{R}^{B \times N_{\text{vis}} \times D_{\text{vis}}}$, we proceed as follows.

Structure-based soft gate. We use a soft gate because structural cues extracted from edge maps are not always reliable, and their contribution should be adjusted adaptively. For the t -th token in the b -th batch, we estimate structural strength by the norm of the structural embedding:

$$\|E_{\text{Stru}}[b, t, :]\|_2 = \sqrt{\sum_{d=1}^{D_{\text{vis}}} (E_{\text{Stru}}[b, t, d])^2} \quad (1)$$

This norm directly measures activation strength across embedding dimensions. However, the Canny derived structural image often carries spurious edges and background artifacts, Our Appendix has shown that this raw structural embedding can reduce performance. To further purify structural information from E_{Stru} , we smooth E_{Stru} and apply a structural soft gate to suppress noise and stabilize fusion:

$$w_{\text{Stru}}[b, t] = \sigma(\alpha(\|E_{\text{Stru}}[b, t, :]\|_2 - \tau)) \quad (2)$$

where $\alpha > 0$ is a temperature, τ is a structure threshold, and σ is the sigmoid [48] that avoids instability from hard thresholds. A larger embedding value yields $w_{\text{Stru}}[b, t] \rightarrow 1$ and the smaller value yields $w_{\text{Stru}}[b, t] \rightarrow 0$, thereby increasing the discriminability of structural information. We then define the detail gate as $w_{\text{Det}}[b, t] = 1 - w_{\text{Stru}}[b, t]$.

Schur-style fusion. We adopt Schur-style fusion because element-wise interaction can better preserve the consistent information shared by structure and detail features. We view E_{Img} as the sum of a structure embedding and a detail embedding. To reduce structural leakage on the E_{Img} , we first project away the structural direction, while preserving the detail direction, and then apply gated fusion. Image embeddings lie in a vector space, l_2 normalization removes magnitude while keeping direction. We therefore take the structural unit direction u and the retention factor γ :

$$u[b, t] = \frac{E_{\text{Stru}}[b, t]}{\|E_{\text{Stru}}[b, t]\|_2 + \varepsilon}, \quad \gamma[b, t] = \frac{\|E_{\text{Stru}}[b, t]\|_2^2}{\lambda + \|E_{\text{Stru}}[b, t]\|_2^2} \quad (3)$$

where ε prevents division by zero (we set $1e-6$) and λ controls structural preservation so that larger λ retains a smaller structural portion. We then define the projection onto the

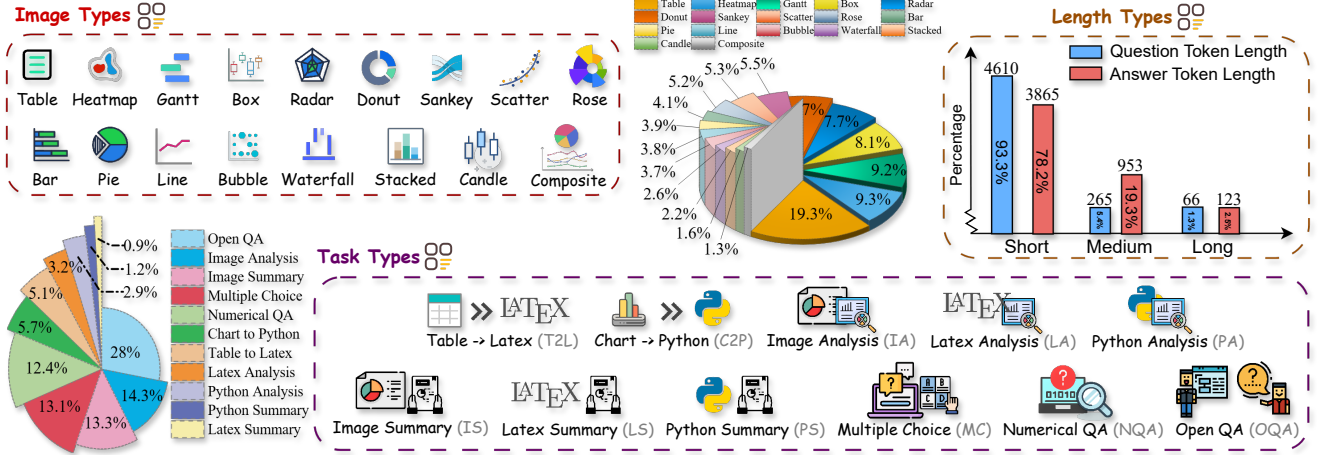


Figure 3. Overview of our TwintVQA benchmark. It covers **17 image types** (e.g., bar, pie, line, bubble, heatmap, radar, donut, sankey, scatter, rose, box, waterfall, stacked, candle, gantt, composite, table) and **11 task types** across 3 data formats (Image, LaTeX, Python), including Table $\rightarrow$ LaTeX (T2L), Chart $\rightarrow$ Python (C2P), Image / LaTeX / Python Analysis and Summary (IA / LA / PS / LS / PS), Multiple Choice (MC), Numerical QA (NQA), and Open QA (OQA). The right panel shows the **length distribution** for questions and answers with three bands: *Short* (0, 256], *Medium* (256, 1024], and *Long* (1024, 4096] tokens. Zoom in for more details.

structural direction $u[b, t]$ as $proj(x) = (x \cdot u[b, t]) u[b, t]$. We further apply a Schur-style fusion to remove the structural component from the E_{Img} , more details about Schur-style are provided in our Appendix:

$$E_{Det}[b, t] = E_{Img}[b, t] - \underbrace{\gamma[b, t] w_{Stru}[b, t]^2}_{\text{adaptive removal}} proj(E_{Img}[b, t]) \quad (4)$$

We use adaptive removal to reduce redundant structural information in the raw-image features, so that the fused representation can better preserve complementary details. Here $\gamma[b, t]$ is an adaptive scaling factor so that stronger structure implies stronger removal, and $w_{Stru}[b, t]^2$ further emphasizes highly structural positions while leaving weak-structure tokens nearly unchanged. Intuitively, when t -th token is structure-dominated, subtracting the projection of $E_{Img}[b, t]$ onto $u[b, t]$ suppresses structural leakage and shifts the representation toward fine details in E_{Det} .

$$E_{fuse}[b, t] = w_{Det}[b, t] E_{Det}[b, t] + w_{Stru}[b, t] E_{Stru}[b, t] \quad (5)$$

Compared with directly using E_{Img} , this design separates structural and detail cues before fusion. In Stage 1, the model is trained with cross-entropy to generate the target answer based on E_{fuse} . Finally, E_{fuse} is fed to the connector and text decoder.

3.2. Stage 2: MINT Preference Learning

Our Motivation: To further optimize the generated text, we build a preference dataset based on Stage 1 dataset and train with MINT. For each question, we provide two labels: a chosen response and a rejected response. We optimize Twin-T to prefer the chosen response during the

training process. MINT contains two components: a **Numeric-Keyword Preference** that strengthens accuracy on numbers and keywords, and a **Text-Vision Evidence Preference** that increases text-vision matching level. Building on Stage 1 alignment, Stage 2 trains only the text decoder.

3.2.1. Numeric-Keyword Preference

Most chart-table tasks revolve around numbers: reading values, comparing data, computing differences or ratios, and tracking trends. As shown in the ablations Tab.6, the Stage 1 model still errs on numeric tokens and comparator words. To address this, we introduce a token-level method that concentrates training on numeric tokens and comparator keywords (e.g., *greater*, *smaller*), encouraging the model to focus on these positions.

Let $M_{num}[b, t] \in \{0, 1\}$ and $M_{key}[b, t] \in \{0, 1\}$ be masks for numeric and comparator keywords in the chosen and rejected labels. We form a per-token weight $W_{[b, t]}$, normalized within each training sample to have mean ≈ 1 :

$$W[b, t] = \text{norm}\left(1 + M_{num}[b, t] + M_{key}[b, t]\right) \quad (6)$$

Let $\ell_{[b, t]}^{\text{ch}}$ and $\ell_{[b, t]}^{\text{rj}}$ denote the output logits of the “gold token” at $[b, t]$ under the chosen and rejected responses respectively. Here, “gold token” specifically refers to masked token (i.e., indices with $M_{num}[b, t] = 1$ or $M_{key}[b, t] = 1$). We want Twin-T to imitate the chosen response rather than the rejected one, especially on critical numeric and comparison tokens. We introduce a contrastive loss of numeric-keyword $\mathcal{L}_{NK}$. $\mathcal{L}_{NK}$ explicitly forces the chosen response to

score higher than the rejected one on those tokens.

$$\mathcal{L}_{\text{NK}} = \frac{1}{N} \sum_{t=0}^N W_{[b,t]} \left[-(\ell_{[b,t]}^{\text{ch}} - \ell_{[b,t]}^{\text{rj}}) \right]_+ \quad (7)$$

where N is the total number of tokens. Direct preference learning usually treats all generated tokens as equally important [44]. In chart-table QA, style tokens and filler phrases are often not critical, but numeric values, units, and comparative expressions are crucial for correctness. We therefore employ a weight W that up-weights those gold tokens, so that the gradients focus on factual accuracy rather than surface phrasing.

To further stabilize copying or computing numbers from the input image, we explicitly reduce logits entropy at numeric positions. Let the model output logits be $\text{logits}[b, t] \in \mathbb{R}^V$ with vocabulary size V , and $p[b, t](v) = \text{softmax}(\text{logits}[b, t])[v]$. Define entropy $H_{[b,t]}$:

$$H_{[b,t]} = - \sum_{v=1}^V p[b, t](v) \log p[b, t](v) \quad (8)$$

Let $\mathcal{P}$ be the index set of numeric positions in the chosen response. The low-entropy regularizer is $\mathcal{L}_{\text{Ent}} = \frac{1}{|\mathcal{P}|} \sum_{t \in \mathcal{P}} H_{[b,t]}$. The overall numeric-keyword preference loss combines the weighted hinge contrast and the entropy penalty: $\mathcal{L}_{\text{NKP}} = \mathcal{L}_{\text{NK}} + \mathcal{L}_{\text{Ent}}$.

3.2.2. Text-Vision Evidence Preference

To make answers more reliable, we constrain the chosen response to concentrate on visual evidence so that the text is supported by facts in the figure (e.g., data values, curves, and annotations). We wrap the evidence span in each response, for example, suppose that a chart shows a city’s population: 0.50M in 2024 and 0.52M in 2025. If the question asks for the relative growth from 2024 to 2025, We do not simply want the model to output a final value such as 4%. Instead, we want Twin-T to learn and expose the actual computation process, thus the chosen response includes “<evidence> (0.52 - 0.5)/0.5 = 4% </evidence>”. By contrast, the rejected response contains coarse or incorrect evidence, e.g. “<evidence> (0.52 - 0.5)/0.5 = 3% </evidence>”.

Let $\text{Mat}_{\text{txt-vis}} \in \mathbb{R}^{N_{\text{txt}} \times N_{\text{vis}}}$ be the cosine similarity matrix between the last-layer hidden states of text tokens and visual tokens. We aggregate only the response tokens within the evidence span. For batch b :

$$\mu(\text{Mat}_{\text{txt-vis}}[b]) = \frac{1}{|\mathcal{R}|} \sum_{t \in \mathcal{R}} \max_{1 \leq j \leq N_{\text{vis}}} \text{Mat}_{\text{txt-vis}}[b, t, j] \quad (9)$$

where $\mathcal{R}$ is the index set of tokens within the <evidence>...</evidence> span of the chosen and rejected response, and the row $\text{Mat}_{\text{txt-vis}}[b, t, :]$ gives the cosine similarities between the t -th evidence text token and all

visual tokens. We then define a text-vision matching loss $\mathcal{L}_{\text{TV}}$ on chosen (ch) versus rejected (rj) responses:

$$\mathcal{L}_{\text{TV}} = \frac{1}{B} \sum_{b=1}^B \left[-(\mu(\text{Mat}_{\text{txt-vis}}^{\text{ch}}[b]) - \mu(\text{Mat}_{\text{txt-vis}}^{\text{rj}}[b])) \right]_+ \quad (10)$$

When μ of the chosen response exceeds that of the rejected response, the loss is small, indicating stronger text-vision matching on the evidence span. The final MINT loss is $\mathcal{L}_{\text{MINT}} = \mathcal{L}_{\text{NKP}} + \mathcal{L}_{\text{TV}}$.

Overall, MINT preference learning steers Twin-T toward three behaviors: higher numeric accuracy, more stable number logits, and stronger text–vision matching, which together deliver stronger chart–table tasks performance.

4. Experiments and Analysis

We collected $\sim 40\text{K}$ chart-table images from public web sources. Using our question templates, we prompt GPT-4o to generate five QA pairs per image, yielding $\sim 200\text{K}$ QA pairs in total. For evaluation, TwintVQA’s images are collected from arXiv papers and web data and then manually screened, resulting in $\sim 5\text{K}$ charts and tables that closely resemble real-world cases. There is no overlap between the training and benchmark QA pairs. All experiments run on NVIDIA A800 GPUs and are repeated three times, we report the average performance.

4.1. Datasets Construction

The TwintVQA is a comprehensive benchmark to evaluate VLMs on chart-table tasks. In the evaluation process, each QA pair is scored by GPT-4o-Mini on a scale from 0 to 1, and we report the final score as the mean of all scores (converted to percentage). Our benchmark details are shown in Fig.3. In summary, our TwintVQA benchmark contains 4,941 questions, covers 17 chart types and 11 tasks, supports 3 data formats (Image, Python, LaTeX), and includes short, medium, and long QA lengths.

4.2. Benchmarks Evaluation

TwintVQA and Public Benchmarks. We evaluate Twin-T (1B/7B) on our TwintVQA and on other public chart-table benchmarks: AI2D [18], CharXiv [40], ChartMimic [45], ChartQA [27], LogicVista [42], OCRVQA [30], SEED-Bench2 [20] and TableVQA [19]. To provide broad comparisons, we include general-purpose VLMs with diverse sizes and architectures, specific experts models, as shown in Tab.1.

Twin-T-7B achieves the highest overall score (719.94) and sets the strongest open-source results across a broad range of chart-table benchmarks, including AI2D, CharXiv_R, LogicVista, SEEDBench2, TableVQA, and our TwintVQA. Twin-T-1B also performs competitively despite being much smaller, and surpasses many larger

Table 1. Performance results on public chart-table benchmarks. Best scores are highlighted in **blue**, and second-best scores in **green**; CharXiv_D and CharXiv_R denote chart description and chart reasoning respectively; “Overall” is the total score; “†” denotes small models ($\leq 4B$); “↑” indicates higher is better; “–” denotes a failure on that benchmark.

Models	AI2D↑	CharXiv _D ↑	CharXiv _R ↑	ChartMimic↑	ChartQA↑	LogicVista↑	OCRVQA↑	SEEDBench2↑	TableVQA↑	TwintVQA↑	Overall↑
General-Purpose Vision-Language Models											
Gemma3-12B[38]	45.66	25.32	13.50	19.74	28.13	20.58	39.12	29.38	35.73	26.43	283.59
Gemma3-27B	74.45	58.58	27.70	56.40	42.24	38.26	54.24	61.09	60.53	38.52	511.77
Kimi-VL-16B[39]	83.03	63.41	28.50	36.23	83.40	36.91	82.42	68.64	68.85	57.17	608.15
LlamaV-3.2-11B[1]	70.14	63.32	26.30	21.29	60.28	33.78	26.16	66.45	23.17	35.09	425.98
† Ovis2-1B[26]	76.52	58.80	27.60	10.83	74.92	25.50	70.18	61.48	52.87	37.17	495.87
Ovis2-8B	86.72	72.87	39.50	45.50	84.36	39.37	68.58	70.09	73.60	56.63	637.22
Phi4-Vision-6B[2]	83.19	60.72	31.10	–	83.08	36.24	60.35	68.69	67.12	42.33	532.82
† Qwen2.5-VL-3B[3]	81.31	61.62	27.80	32.65	83.96	37.36	70.35	68.47	69.98	50.31	583.81
Qwen2.5-VL-7B	80.83	76.32	38.30	37.40	86.32	46.09	70.90	69.61	74.28	51.36	631.09
SAIL1.6-VL-8B[11]	87.50	72.98	39.10	13.31	85.96	41.39	80.92	71.10	54.87	53.22	600.35
LLaVA-Next-8B[21]	73.28	44.90	15.60	19.52	69.36	25.50	61.36	56.83	42.48	21.09	429.92
LLaVA-Next-13B	70.27	38.35	14.10	13.71	57.16	26.62	64.84	54.46	39.78	19.30	398.59
LLaVA-1.5-7B[25]	52.75	38.25	8.50	–	17.80	24.83	48.08	41.63	18.43	13.34	263.61
LLaVA-1.5-13B	56.57	36.78	8.80	–	19.48	27.07	51.17	44.36	19.50	15.28	279.01
Chart-Table Expert Vision-Language Models											
ChartLlama-13B[15]	59.39	31.85	12.60	32.13	82.16	30.20	75.05	44.58	39.57	22.62	430.15
ChartVLM-8B[41]	37.27	25.55	11.30	30.11	69.64	24.83	61.95	36.58	56.14	13.80	367.64
† ChartGemma-3B[28]	32.51	27.93	17.80	25.95	68.06	20.81	57.74	32.32	56.51	19.85	361.11
† ChartCap-4B[24]	76.59	60.59	31.10	21.03	73.40	26.17	62.53	62.14	53.68	28.14	495.37
ChartAst-13B[29]	77.62	75.36	41.25	28.14	83.40	30.17	73.90	68.56	71.20	51.38	600.98
SynTab-7B[50]	62.63	51.73	27.47	32.72	70.24	22.93	67.12	52.16	48.52	45.83	481.35
API Vision-Language Models											
GLM-4.5V-106B[16]	85.72	76.53	51.70	42.31	86.76	52.35	58.11	72.90	80.08	60.62	667.08
GPT-4o[17]	81.44	85.08	47.80	61.83	93.12	57.05	70.57	73.12	77.82	67.06	714.89
GPT-4o-Mini	77.23	72.58	45.35	41.92	81.84	40.72	18.65	68.82	65.17	50.94	563.22
Gemini-2.5-Pro[9]	89.80	82.42	50.23	64.22	90.41	58.50	73.24	73.81	78.05	63.58	724.26
Gemini-2.5-Flash	72.36	69.25	39.25	50.80	75.68	45.02	42.33	50.94	76.02	59.64	580.65
Our Models											
† Twin-T-1B	80.57	78.25	33.20	15.16	80.16	31.32	72.17	66.80	59.78	58.79	576.20
Twin-T-7B	90.77	84.88	57.10	34.37	91.88	59.06	77.28	74.13	80.27	70.20	719.94

VLMs. Twin-T falls short mainly on OCR-heavy and code-reconstruction tasks. The gaps are driven by two factors: model size and data scale. Our models have fewer parameters and are trained on a smaller datasets than large commercial or API VLMs, which limits OCR coverage and code-reconstruction priors. Twin-T remains state-of-the-art among open-source models, surpasses GLM-4.5V-106B, and remains competitive with GPT-4o and Gemini-2.5-Pro.

TwintVQA Task Types and QA Lengths Analysis. To further evaluate the Twin-T model and other VLMs on diverse chart-table tasks performance, we conduct the grouping experiments on TwintVQA benchmark, grouping results by **task types** and by **QA lengths**, as shown in Tab.2.

Twin-T-7B achieves the strongest open-source results on TwintVQA. It scores 91.36 on Multiple Choice (MC), 91.69 on LaTeX Analysis (LA), 76.04 on Numerical QA (NQA), 66.49 on Open QA (OQA), 90.58 on Python Analysis (PA), and 83.57 on Table-to-LaTeX (T2L), and remains competitive on other tasks such as Image Summary (IS). Twin-T-1B also outperforms other small models, especially on NQA and OQA, showing that even at 1B scale it can extract numbers and make comparisons reliably. A remaining challenge is Chart-to-Python (C2P), where performance is lower for most models. Unlike LaTeX generation, C2P requires faithful program reconstruction and flexible code

synthesis, which likely needs code-aligned supervision and larger model capacity.

Across QA length buckets, Twin-T-7B is best on Long (61.16) among open-source models, and remains strong on Medium (69.70) and Short (70.91). Twin-T-1B also scores well relative to other models of similar size.

TwintVQA Chart and Table Types Analysis. To assess model performance across different **chart-table types**, and to validate the benchmark’s coverage, we conduct more evaluations on TwintVQA, as shown in Tab.3.

Across chart-table types, Twin-T-7B outperforms open-source models and GLM-4.5V-106B on many structure-heavy chart types (e.g., Bar, Box, Composite and Donut). It is even competitive with GPT-4o. Twin-T-1B also leads among small models ($\leq 4B$). Overall, on our TwintVQA benchmark the gains align with our design: Stage 1 separates structure from detail, and Stage 2 adds numeric preference learning. Together these steps help the model track axes, grids, and layout while keeping numbers and comparisons accurate.

4.3. Parameter-Sensitive Experiments

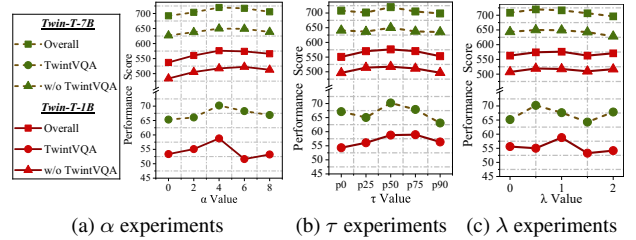
Parameters Analysis. We study three Stage 1 parameters, α , τ , and λ , and report their effects on performance in Fig. 4. All three parameters show clear, stable optima. For α , performance increases as α grows from 0 to about

Table 2. Performance of VLMs across task types and question-answering lengths on our TwintVQA benchmark.

Models	Task Types											Length Types		
	C2P↑	MC↑	IA↑	IS↑	LA↑	LS↑	NQA↑	OQA↑	PA↑	PS↑	T2L↑	Long↑	Medium↑	Short↑
General-Purpose Vision-Language Models														
Gemma3-12B	1.05	41.26	35.82	15.53	32.81	58.75	28.14	20.58	46.11	46.06	4.77	15.79	28.99	25.96
Gemma3-27B	4.72	53.08	56.88	24.13	35.18	45.95	38.75	34.25	47.36	40.26	43.70	38.05	43.38	36.78
Kimi-VL-16B	12.06	71.18	76.19	40.83	79.50	75.62	56.17	52.57	81.23	74.14	54.36	47.27	65.12	54.85
LlamaV-3.2-11B	2.69	60.84	31.63	15.09	76.84	46.24	36.43	32.05	61.69	61.45	31.38	27.65	35.20	35.48
[†] Ovis2-1B	4.75	52.23	49.09	27.42	57.01	48.50	33.59	35.82	50.87	54.26	17.10	25.83	38.67	37.29
Ovis2-8B	11.83	77.54	73.36	45.73	81.25	65.31	49.27	50.52	76.67	74.62	40.68	43.16	60.56	55.74
Phi4-Vision-6B	9.12	56.32	45.86	27.12	71.83	54.21	46.73	43.86	72.08	66.76	11.35	27.59	40.89	43.72
[†] Qwen2.5-VL-3B	8.57	65.02	66.73	33.28	77.44	65.40	46.83	46.51	75.08	67.79	49.74	43.99	57.38	48.10
Qwen2.5-VL-7B	13.80	65.23	73.51	32.64	82.03	67.95	45.88	43.54	80.63	68.57	58.74	49.46	64.02	46.86
SAI1.6-VL-8B	11.51	72.38	57.50	34.39	77.72	25.83	66.22	51.41	77.04	57.07	40.92	38.72	50.65	55.07
LLaVA-Next-8B	0.49	41.39	13.95	8.25	61.75	53.74	19.47	17.09	60.20	59.95	9.44	13.18	22.41	21.08
LLaVA-Next-13B	0.65	35.59	17.65	8.14	48.05	53.45	18.38	16.13	49.47	52.71	3.22	7.44	21.37	19.24
LLaVA-1.5-7B	0.35	30.39	3.30	3.07	56.08	51.83	11.10	10.31	43.58	46.66	2.96	12.17	13.55	13.33
LLaVA-1.5-13B	–	35.23	3.84	4.94	56.64	64.93	11.39	12.45	52.91	54.66	0.56	13.20	14.76	15.59
Chart-Table Expert Vision-Language Models														
ChartLlama-13B	3.33	41.54	11.16	2.15	55.94	65.11	39.47	19.34	30.56	55.31	–	15.09	19.61	24.31
ChartVLM-8B	–	29.23	11.37	1.52	49.69	96.25	10.70	6.45	51.67	22.50	4.55	10.53	19.01	11.79
[†] ChartGemma-3B	–	28.85	3.90	8.26	29.19	12.50	33.98	26.37	30.28	38.12	2.27	6.58	10.61	24.46
[†] ChartCap-4B	0.83	48.46	28.01	5.30	81.25	50.50	25.31	27.14	62.22	52.50	–	19.74	30.51	27.63
SynTab-7B	14.34	39.84	52.93	45.75	48.21	40.64	46.62	52.14	46.55	52.15	38.01	36.84	44.01	47.02
ChartAst-13B	14.06	59.99	53.02	46.13	62.79	52.26	60.18	48.93	60.11	56.57	58.91	45.23	49.89	52.28
API Vision-Language Models														
GLM-4.5V-106B	23.44	77.55	80.93	45.45	87.63	76.43	53.35	54.49	88.82	77.45	53.57	56.43	70.66	57.20
GPT-4o	31.13	74.62	94.59	59.89	87.50	91.25	55.56	61.18	82.61	87.81	44.09	62.89	77.37	62.97
GPT-4o-Mini	15.23	62.81	75.02	39.01	87.54	72.19	42.31	42.64	85.77	85.60	36.69	47.14	63.97	46.41
Gemini-2.5-Pro	10.33	80.01	93.96	46.55	85.12	85.85	50.25	61.34	72.50	74.88	54.55	47.11	73.77	60.24
Gemini-2.5-Flash	8.25	78.21	89.11	34.95	86.25	95.12	51.77	53.03	71.67	71.88	52.05	46.58	70.39	54.93
Our Models														
[†] Twin-T-1B	6.05	74.85	60.88	43.16	67.53	67.19	68.73	61.21	70.15	72.02	57.94	41.40	54.30	61.45
Twin-T-7B	18.61	91.36	73.70	54.38	91.69	71.88	76.04	66.49	90.58	82.07	83.57	61.16	69.70	70.91

4, then declines slightly. For both 1B and 7B, $\alpha \approx 4$ performs best: the gate is sharp enough to separate structure from detail, but not so hard that it amplifies local noise. For τ , we sweep ascending quantiles $p_0, p_{25}, p_{50}, p_{75}, p_{90}$ and find the 7B peak at p_{50} , while the 1B model peaks at p_{50} or p_{75} . Low τ retains background clutter and spurious strokes; very high τ removes thin axes and weak but informative contours. The two model scales differ in sensitivity: the 7B model is tolerant around the median, whereas the 1B model benefits from a slightly higher cutoff to suppress noise. For λ , the 7B model peaks near $\lambda \approx 1$, while the 1B model peaks around $\lambda \approx 0.5$, performance declines when λ moves away from these optima. A small λ leaves structural leakage in the E_{img} branch and blurs detail tokens, whereas a large λ subtracts too much and removes useful layout cues.

Structure Image Extraction. We test five structure extraction methods: Canny [10], Sobel [12], Scharr [33], Laplacian [31], and HED [43]. In brief, Canny yields the best speed–performance trade-off: it adds only ~ 0.008 s/image during inference, which is negligible compared with the ~ 0.02 s/output-token. Other extractors are more sensitive to background clutter and spurious details, tending to over-extract irrelevant high-frequency patterns, which weakens structure and detail separation.

Figure 4. We examine how α , τ , and λ affect the Twin-T’s performance. “w/o TwinVQA” denotes the aggregate score over all public benchmarks, excluding our TwintVQA.

4.4. Ablation Studies

We quantify the contribution of each module. First, we ablate the two training stages, as shown in Tab.4, where “✓” denotes inclusion and “✗” denotes exclusion. The results indicate that removing either stage leads to performance degradation. To further pinpoint where the performance gains come from, we ablate key components within each stage: in Stage 1, the dual-head encoder, the structure-aware gating, and the Schur-style fusion, as shown in Tab.5.

Removing the dual-head encoder leads to the largest

Table 3. Performance of different VLMs across chart-table types on our TwintVQA benchmark.

Models	Chart-Table Types																
	Bar↑	Box↑	Bubble↑	Candle↑	Composite↑	Donut↑	Gantt↑	Heatmap↑	Line↑	Pie↑	Radar↑	Rose↑	Sankey↑	Scatter↑	Stacked↑	Table↑	Waterfall↑
General-Purpose Vision-Language Models																	
Gemma3-12B	25.65	35.76	31.50	51.11	34.44	36.73	18.89	23.04	23.18	36.50	28.75	20.19	13.21	40.54	23.08	20.97	23.57
Gemma3-27B	38.49	39.83	34.75	38.38	35.68	50.72	28.91	33.20	40.01	46.36	38.82	35.50	34.45	44.88	36.32	40.27	37.25
Kimi-VL-16B	64.14	55.63	58.09	48.89	60.36	64.80	48.40	57.31	60.73	61.39	54.47	48.39	48.07	62.77	53.85	61.02	59.19
LlamaV-3.2-11B	36.53	36.84	37.25	33.98	31.29	43.18	22.02	34.33	40.22	39.32	29.89	27.40	26.65	42.45	38.05	40.12	35.11
Ovis2-1B	41.92	36.80	38.51	39.11	35.19	47.84	28.44	39.14	38.98	44.65	39.19	29.29	28.17	41.59	37.29	35.15	44.89
Ovis2-8B	67.10	55.07	66.89	54.11	45.56	64.54	44.93	52.04	68.77	69.15	55.25	37.04	46.82	51.96	48.46	61.03	75.04
Phi4-Vision-6B	49.39	42.72	47.11	42.38	48.50	50.41	32.92	41.46	48.56	53.10	41.06	39.21	33.38	50.62	39.50	39.75	43.23
† Qwen2.5-VL-3B	56.77	49.33	51.54	40.11	52.42	55.49	34.70	46.80	55.17	57.62	50.74	46.07	40.22	57.95	40.75	56.47	57.83
Qwen2.5-VL-7B	60.83	43.44	48.95	43.12	51.28	56.21	42.43	48.36	55.18	60.99	48.91	47.05	42.75	52.94	46.54	59.57	53.05
SAIL1.6-VL-8B	63.18	53.28	54.33	44.46	54.12	60.50	47.83	49.16	58.75	59.40	55.96	47.13	44.92	54.68	47.65	54.65	50.05
LLaVA-Next-8B	21.83	20.28	26.08	22.46	17.30	28.59	10.62	21.90	25.52	23.71	17.28	14.77	14.23	24.81	19.67	24.89	23.84
LLaVA-Next-13B	19.64	18.77	21.70	26.38	15.47	27.47	10.89	19.50	22.73	23.40	15.87	13.29	12.75	26.12	21.15	20.21	24.28
LLaVA-1.5-7B	9.54	12.70	15.80	13.52	4.84	16.71	5.94	16.15	13.17	17.87	11.17	11.18	9.05	12.24	13.22	17.81	11.39
LLaVA-1.5-13B	13.08	14.13	15.84	19.71	9.52	19.50	10.09	17.76	12.05	18.86	12.83	14.55	9.01	15.92	8.68	19.40	14.82
Chart-Table Expert Vision-Language Models																	
ChartLlama-13B	31.60	24.90	29.50	35.16	30.78	14.57	17.39	24.28	29.86	17.25	19.55	21.26	20.39	18.46	7.54	23.38	45.36
ChartVLM-8B	10.20	16.38	15.78	21.67	33.18	12.38	4.09	13.04	10.86	9.25	5.15	8.52	13.04	16.96	25.92	20.28	18.57
† ChartGemma-3B	32.55	10.17	23.83	44.44	27.56	30.08	18.46	14.72	26.50	29.90	11.12	10.93	8.61	20.71	23.46	17.19	42.79
† ChartCap-4B	27.55	31.90	35.28	26.11	40.04	34.92	18.48	22.09	30.05	29.75	30.22	27.78	25.23	41.50	29.62	23.43	39.57
SynTab-7B	45.27	47.03	46.30	45.05	49.90	45.58	45.33	45.31	47.27	45.96	47.35	44.35	43.77	48.57	47.08	44.88	46.09
ChartAst-13B	51.41	51.34	49.93	47.98	53.89	50.98	50.87	50.67	52.56	50.55	51.59	48.77	43.58	51.34	50.75	55.37	51.37
API Vision-Language Models																	
GLM-4.5V-106B	65.99	55.99	58.04	61.02	58.74	70.12	51.76	59.58	57.66	66.47	62.47	57.39	52.42	63.05	58.49	64.48	62.45
GPT-4o	78.13	59.24	74.44	82.78	69.44	71.49	45.07	63.24	77.50	81.75	68.48	61.11	57.36	80.61	55.38	68.82	82.86
GPT-4o-Mini	53.12	45.77	54.84	48.18	57.10	57.34	39.72	51.78	54.58	55.27	55.75	42.38	42.88	58.70	49.58	53.58	49.89
Gemini-2.5-Pro	67.21	49.48	69.72	79.78	60.05	65.90	48.74	52.91	70.68	79.10	61.62	58.52	58.79	71.96	56.15	71.19	77.50
Gemini-2.5-Flash	53.15	50.75	61.39	73.33	59.44	65.95	42.83	57.54	66.82	72.85	60.88	52.78	38.61	63.75	37.31	68.62	70.71
Our Models																	
† Twin-T-1B	62.62	59.19	60.87	59.12	58.91	67.21	50.65	61.00	63.81	63.51	61.75	49.08	50.14	58.56	57.04	59.71	56.02
Twin-T-7B	78.92	67.07	71.85	68.39	70.10	76.79	62.45	67.62	75.17	74.81	68.85	57.44	61.62	67.68	66.80	77.69	69.55

Table 4. Stage ablation experiments on Twin-T. “w/o TwintVQA”: aggregate over public benchmarks excluding TwintVQA.

Models	Stage1	Stage2	TwintVQA↑	w/o TwintVQA↑	Overall↑
Twin-T-1B	✓	✓	58.79	517.41	576.20
Twin-T-7B	✓	✓	70.20	649.74	719.94
Twin-T-1B	✓	✗	53.93 ▼4.86	475.33 ▼42.08	529.26 ▼46.94
Twin-T-7B	✓	✗	66.93 ▼3.27	608.12 ▼41.62	675.05 ▼44.89
Twin-T-1B	✗	✓	50.95 ▼7.84	461.87 ▼55.54	512.82 ▼63.38
Twin-T-7B	✗	✓	57.05 ▼13.15	615.47 ▼34.27	672.52 ▼47.42

Table 5. Ablation experiments of Stage 1 modules.

Models	TwintVQA↑	w/o TwintVQA↑	Overall↑
Twin-T-1B	53.93	475.33	529.26
↳w/o Dual-head encoder	49.21 ▼4.72	453.60 ▼21.73	502.81 ▼26.45
↳w/o Structure gating	51.12 ▼2.81	463.94 ▼11.39	515.06 ▼14.20
↳w/o Schur-style fusion	51.85 ▼2.08	465.67 ▼9.66	517.52 ▼11.74
Twin-T-7B	66.93	608.12	675.05
↳w/o Dual-head encoder	61.82 ▼5.11	580.46 ▼27.66	642.28 ▼32.77
↳w/o Structure gating	62.84 ▼4.09	594.14 ▼13.98	656.98 ▼18.07
↳w/o Schur-style fusion	61.71 ▼5.22	598.55 ▼9.57	660.26 ▼14.79

degradation, reducing Overall performance by about 5% on both the 1B and 7B models. Dropping the structure gating lowers performance by roughly 2.7% for both sizes, and removing the Schur-style fusion causes about a 2.2% drop. This shows that the dual-head design is the main driver of Stage 1 gains, while the gate and the Schur fusion each provide an additional 3% boost.

In Tab.6, **NK Acc** and **Match** are both scored by GPT-4o on a [0, 100] scale and averaged over all QA pairs. **NK Acc** shows numerical correctness, and **Match** reflects how well the answer is matched in the visual evidence. **Entropy** is the average logits entropy on numeric tokens, lower entropy

indicates more confident and stable numeric generation.

Table 6. Ablation experiments of Stage 2 modules.

Models	TwintVQA↑	w/o TwintVQA↑	Overall↑	NK Acc (%)↑	Entropy (%)↓	Match (%)↑
Twin-T-1B	58.79	517.41	576.20	84.20	16.80	90.60
↳w/o Num-Key	52.72 ▼6.07	490.44 ▼26.97	543.16 ▼33.04	81.50 ▼2.70	17.40 ▲0.60	88.30 ▼2.30
↳w/o Low-Entropy	55.91 ▼2.88	492.17 ▼25.24	548.08 ▼28.12	83.70 ▼0.50	24.90 ▲8.10	88.90 ▼1.70
↳w/o Txt-Vis	55.06 ▼3.73	503.11 ▼14.30	558.17 ▼18.03	84.00 ▼0.20	20.70 ▲3.90	82.40 ▼8.20
Twin-T-7B	70.20	649.74	719.94	90.60	13.40	95.80
↳w/o Num-Key	64.82 ▼5.38	627.68 ▼22.06	692.50 ▼27.44	84.70 ▼5.90	15.80 ▲2.40	94.10 ▼1.70
↳w/o Low-Entropy	66.79 ▼3.41	635.57 ▼14.17	702.36 ▼17.58	89.23 ▼1.37	17.20 ▲3.80	94.60 ▼1.20
↳w/o Txt-Vis	66.24 ▼3.96	629.41 ▼20.33	695.65 ▼24.29	88.75 ▼1.85	14.20 ▲0.80	89.80 ▼6.00

Num-Key preference drives the largest overall performance gains: removing it yields the biggest Overall drops and the sharpest decline in NK Acc (1B: −2.70%; 7B: −5.90%), confirming its role in numeric fidelity. Low-entropy regularization chiefly stabilizes number generation: ablating it spikes entropy markedly in Entropy metric (1B: +8.10%; 7B: +3.80%). Text-vision evidence preference most strongly affects matching level: removing it causes the largest drop in Txt-Vis Match (1B: −8.20%; 7B: −6.00%).

5. Conclusion

In this paper, we propose Twin-T (1B/7B), a two-stage trained VLM designed for chart-table tasks, which separates structures and details. We also introduce TwintVQA, a benchmark covering 17 chart types, 11 task types, and short/medium/long QA lengths. Across open-source VLMs and API models, Twin-T shows impressive performance on public chart-table benchmarks, with the 1B model achieving an overall score of 576.20, and the 7B model reaching 719.94. Among open-source models Twin-T achieves SOTA performance and even remains competitive with Gemini-2.5-Pro and GPT-4o.

Acknowledgments

This work is partly supported by the Project granted by the Ministry of Agriculture and Rural Affairs, the Key Research and Development Program of Heilongjiang Province under Grant Nos. 2022ZX01A22, 2021ZXJ05A03, the National Natural Science Foundation of China under Grant Nos. 62350710797, 61972114, 62106061, the National Science and Technology Major Project of China under Grant Nos. 2021ZD0110901, the collaborative innovation and promotion system of the modern agricultural industry technology for watermelon and melon in Heilongjiang Province.

References

- [1] Llama 3.2: Revolutionizing edge ai and vision with open, customizable models, 2024. 6
- [2] Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*, 2025. 6
- [3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 2, 6
- [4] Jiahua Bao, Siyao Cheng, Jiaying Du, Changjiang He, Zeming Lang, Hao Zhang, and Jie Liu. Bolt: Fewer tokens but more performance retention for efficient vision-language models inference. In *the 33rd ACM International Conference on Multimedia*, pages 4058–4067, 2025. 1
- [5] Jiahua Bao, Siyao Cheng, Jiaying Du, Ziqian Li, Changjiang He, and Jie Liu. History tracker: Retrieving historical image embeddings for efficient fine-grained reasoning in vision-language models. In *2025 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2025. 2
- [6] Jiahua Bao, Siyao Cheng, Jiaying Du, Yuhang Jia, Boyang Niu, Zeming Lang, Changjiang He, Hao Zhang, and Jie Liu. Stop mixing things up! biscuit teaches vision-language models to learn new concepts from images on the spot. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2408–2416, 2026. 1
- [7] Victor Soares Bursztyn, Jane Hoffswell, Eunyee Koh, and Shunan Guo. Representing charts as text for language models: An in-depth study of question answering for bar charts. In *2024 IEEE Visualization and Visual Analytics (VIS)*, pages 266–270. IEEE, 2024. 1, 2
- [8] Victor Carbune, Hassan Mansoor, Fangyu Liu, Rahul Aralikatte, Gilles Baechler, Jindong Chen, and Abhanshu Sharma. Chart-based reasoning: Transferring capabilities from llms to vlms. *arXiv preprint arXiv:2403.12596*, 2024. 2
- [9] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Naveen Sachdeva, Inderjit Dhillon, Marcel Blisstein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 6
- [10] Lijun Ding and Ardeshtir Goshtasby. On the canny edge detector. *Pattern recognition*, 34(3):721–725, 2001. 3, 7
- [11] Hongyuan Dong, Zijian Kang, Weijie Yin, Xiao Liang, Chao Feng, and Jiao Ran. Scalable vision language model training via high quality data curation. *arXiv preprint arXiv:2501.05952*, 2025. 6
- [12] Wenshuo Gao, Xiaoguang Zhang, Lei Yang, and Huizhong Liu. An improved sobel edge detection. In *2010 3rd International conference on computer science and information technology*, pages 67–71. IEEE, 2010. 7
- [13] Akash Ghosh, Arkadeep Acharya, Sriparna Saha, Vinija Jain, and Aman Chadha. Exploring the frontier of vision-language models: A survey of current methodologies and future directions. *arXiv preprint arXiv:2404.07214*, 2024. 2
- [14] Tanay Gupta, Tushar Goel, and Ishan Verma. Exploring multimodal language models for sustainability disclosure extraction: A comparative study. In *The Sixth Workshop on Insights from Negative Results in NLP*, pages 141–149, 2025. 2
- [15] Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, and Hanwang Zhang. Chartllama: A multimodal llm for chart understanding and generation. *arXiv preprint arXiv:2311.16483*, 2023. 2, 6
- [16] Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Li-hang Pan, et al. Glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv e-prints*, pages arXiv–2507, 2025. 6
- [17] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 2, 6
- [18] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pages 235–251. Springer, 2016. 5
- [19] Yoonsik Kim, Moonbin Yim, and Ka Yeon Song. Tablevqa-bench: A visual question answering benchmark on multiple table domains. *arXiv preprint arXiv:2404.19205*, 2024. 2, 5
- [20] Bohao Li, Yuying Ge, Yi Chen, Yixiao Ge, Ruimao Zhang, and Ying Shan. Seed-bench-2-plus: Benchmarking multimodal large language models with text-rich visual comprehension. *arXiv preprint arXiv:2404.16790*, 2024. 5
- [21] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024. 2, 6
- [22] Xinyao Li, Jingjing Li, Fengling Li, Lei Zhu, Yang Yang, and Heng Tao Shen. Generalizing vision-language models to novel domains: A comprehensive survey. *arXiv preprint arXiv:2506.18504*, 2025. 2
- [23] Zongxia Li, Xiyang Wu, Hongyang Du, Fuxiao Liu, Huy Nghiem, and Guangyao Shi. A survey of state of the art large vision language models: Benchmark evaluations and challenges. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1587–1606, 2025. 1

- [24] Junyoung Lim, Jaewoo Ahn, and Gunhee Kim. Chartcap: Mitigating hallucination of dense chart captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13171–13182, 2025. 1, 6
- [25] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024. 6
- [26] Shiyin Lu, Yang Li, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Han-Jia Ye. Ovis: Structural embedding alignment for multimodal large language model. *arXiv preprint arXiv:2405.20797*, 2024. 2, 6
- [27] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022. 2, 5
- [28] Ahmed Masry, Megh Thakkar, Aayush Bajaj, Aaryaman Kartha, Enamul Hoque, and Shafiq Joty. Chartgemma: Visual instruction-tuning for chart reasoning in the wild. *arXiv preprint arXiv:2407.04172*, 2024. 1, 6
- [29] Fanqing Meng, Wenqi Shao, Quanfeng Lu, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. Chartassistant: A universal chart multimodal language model via chart-to-table pre-training and multitask instruction tuning. *arXiv preprint arXiv:2401.02384*, 2024. 2, 6
- [30] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 947–952. IEEE, 2019. 5
- [31] Phillip A Mlsna and Jeffrey J Rodriguez. Gradient and laplacian edge detection. In *The essential guide to image processing*, pages 495–524. Elsevier, 2009. 7
- [32] Chris Paige and Charles Van Loan. A schur decomposition for hamiltonian matrices. *Linear Algebra and its applications*, 41:11–32, 1981. 2
- [33] Wenjun Qian. Edge detection in x-ray images of drill mast welds based on an improved scharr operator. *Welding International*, 38(8):574–581, 2024. 7
- [34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 2
- [35] Sheila Schoepp, Masoud Jafaripour, Yingyue Cao, Tianpei Yang, Fatemeh Abdollahi, Shadan Golestan, Zahin Sufiyan, Osmar R Zaiane, and Matthew E Taylor. The evolving landscape of llm-and vlm-integrated reinforcement learning. *arXiv preprint arXiv:2502.15214*, 2025. 2
- [36] Rui Shao, Wei Li, Lingsen Zhang, Renshan Zhang, Zhiyang Liu, Ran Chen, and Liqiang Nie. Large vlm-based vision-language-action models for robotic manipulation: A survey. *arXiv preprint arXiv:2508.13073*, 2025. 1
- [37] Mong Yuan Sim, Wei Emma Zhang, Xiang Dai, and Biaoyan Fang. Can vlms actually see and read? a survey on modality collapse in vision-language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24452–24470, 2025. 1
- [38] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. 6
- [39] Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chen-zhuang Du, Chu Wei, et al. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*, 2025. 6
- [40] Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. Chartxv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems*, 37:113569–113697, 2024. 5
- [41] Renqiu Xia, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Botian Shi, Junchi Yan, and Bo Zhang. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *IEEE Transactions on Image Processing*, 2025. 2, 6
- [42] Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. Logicvista: Multimodal llm logical reasoning benchmark in visual contexts. *arXiv preprint arXiv:2407.04973*, 2024. 5
- [43] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015. 7
- [44] Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*, 2024. 5
- [45] Cheng Yang, Chufan Shi, Yaxin Liu, Bo Shui, Junjie Wang, Mohan Jing, Linran Xu, Xinyu Zhu, Siheng Li, Yuxiang Zhang, et al. Chartmimic: Evaluating llm’s cross-modal reasoning capability via chart-to-code generation. *arXiv preprint arXiv:2406.09961*, 2024. 5
- [46] Lu Ying, Yun Wang, Haotian Li, Shuguang Dou, Haidong Zhang, Xinyang Jiang, Huamin Qu, and Yingcai Wu. Reviving static charts into live charts. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 1
- [47] Simon Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Peter Tong, et al. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. *Advances in neural information processing systems*, 37:110935–110971, 2024. 2
- [48] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023. 3
- [49] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5625–5644, 2024. 1, 2
- [50] 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 *the Computer Vision and Pattern Recognition Conference*, pages 24796–24806, 2025. 2, 6