ShowTable: Unlocking Creative Table Visualization with Collaborative Reflection and Refinement

Supplementary Material

Overview

A. More Related Works

A.1. Chart Generation with Agentic Tools

With rapid development of current MLLMs [2, 36, 39, 80], the strong tool calling ability provides a promising way to complete multi-modal tasks. Current approaches for chart generation predominantly rely on LLMs coupled with agentic tools, generally falling into three main categories. The first category [45, 53, 75] employs LLMs in an end-to-end manner to produce charts directly from text. These methods typically parse input to identify axes, map entities, and classify chart types, subsequently generating structured specifications for rendering engines like Vega-Lite. While automated, their expressiveness is strictly confined by the predefined grammatical rules of the underlying visualization language. The second category [22, 74, 76–78] focuses on generating executable plotting code (*e.g.*, via Matplotlib/Python). Recent works have incorporated multi-agent frameworks and reflection mechanisms to improve the syntactic correctness of the generated code [18]. Although these methods offer precise control, they heavily depend on external rendering engines and typically lack the capability to handle complex, artistic visual designs beyond standard plots. The third category [68] adopts a retrieval-editing pipeline that selects visual templates from an image corpus and adapts them to new data. While this benefits from template reuse, it is limited by the diversity of the database and often faces challenges in accurately aligning new data with retrieved visual structures.

Discussion. While existing methods excel in structural automation and factual plotting, they fundamentally struggle with **creativity** and **aesthetics**. Code-based and template-based approaches are bound by rigid rendering logic, making it difficult to produce visually striking infographics suit-

Table A1. The comparison between our ShowTable pipeline and code-based methods. For our ShowTable pipeline, we use Gemini-2.5-pro for rewriting, Wan2.5-T2I-Preview for generation, GPT-5 for Reflection, and Wan2.5-I2I-Preview for refinement here. For code-based methods, we use Gemini-2.5-pro to write python code with the matplotlib library.

Methods	DA	TR	RR	AA	AQ	Score
ShowTable Pipeline	69.6	84.9	67.6	68.4	4.9	67.9
Code (Gemini-2.5-pro)	83.8	89.2	85.3	97.7	4.1	79.4

Figure A1. The qualitative comparison between code-based methods and our ShowTable pipeline. The code-based methods often prioritize "correctness" but may fail at "presentation"

able for professional poster design, slide generation, or data-driven storytelling. They prioritize "correctness" but often fail at "presentation". To explicitly validate this, we conducted a quantitative comparison between a strong code-based baseline (e.g., using Gemini-2.5-Pro to generate executable plotting code) and our ShowTable pipeline, as shown in Tab. A1 and Fig. A1. As expected, code generation methods inherently achieve superior factual accuracy across structural metrics such as Data Accuracy and Relative Relationship. However, they fall significantly short in Aesthetic Quality (AQ: 4.1 vs. our 4.9) and frequently produce visual artifacts, such as overlapping text. This rigidity restricts their utility in design-centric, creative scenarios.

Importantly, our objective is not to replace traditional code-based rendering tools where absolute numerical precision is the sole priority. Instead, our work proposes the Creative Table Visualization task to explore the untapped potential of generative models and unified models in this domain. We frame our contribution as a technical advancement in multi-modal generation control: we aim to retain the unparalleled creative flexibility of generation models while actively mitigating their primary weakness, data fidelity, through our progressive self-correcting pipeline. We argue that generative models offer a significantly higher ceiling for flexibility and aesthetic quality, capable of seamlessly integrating data into artistic compositions. Furthermore, equipping text-to-image models with rigorous data fidelity is a vital frontier for achieving more general capabilities in visual synthesis (including scientific reporting), as seen in the trajectory of recent models like Wan2.6 [52] and Nano Banana [20] after the CVPR submission. By validating the feasibility of this approach, we aim to break the traditional boundaries of rule-based rendering and pave the way for more robust, unified, and creative visual synthesis systems.

A.2. Reinforcement Learning for Image Generation

Diffusion models have established themselves as the predominant framework for text-to-image (T2I) generation [5, 23, 30, 44, 46, 65] and text-to-video (T2V) generation [29, 61–63]. The integration of reinforcement learning (RL) into this paradigm began with works utilizing policy gradient optimization to guide the denoising process [4, 16, 67, 71]. The field subsequently expanded to include preference-based alignment methods, which achieve competitive performance without explicit reward modeling [57]. A significant recent development is the adoption of Group Relative Policy Optimization(GRPO) [21, 47], an efficient alternative that has inspired numerous adaptations for T2I generation. These include pioneering works [37, 72] which unified diffusion and flow matching under an SDE-based formulation. This line of inquiry further explores the design of specialized reward models and data curation strategies to enhance the framework's capability for producing high-quality, preference-aligned visual outputs [32, 33, 59].

B. More Method Details

B.1. Prompts in Pipeline

In our ShowTable pipeline, the MLLM acts as the central orchestrator, performing two key roles: rewriting and reflection. We detail the specific prompts used for these modules. The rewriting prompt, shown in Figure A2, instructs the MLLM to reason over the input table and translate its dense data into a detailed descriptive prompt suitable for the diffusion executor. The reflection prompt, shown in Figure A3, guides the MLLM to critically audit the generated image against the original table, identify inaccuracies, and formulate precise, actionable editing instructions for the refinement stage. To ensure fair comparison and consistency across experiments, we use the same system and user prompts for all models tested in these roles.

B.2. Rewriting Training Details

As discussed in Section 3.2, we fine-tune a specialized rewriting module based on Qwen3-8B to handle the critical task of reasoning and compositional planning. This module is trained on our 30K SFT data with both thinking and rewriting result, and we construct various kinds of instruction templates during training to ensure the diversity. For the implementation, we utilize the LLaMA-Factory [79] library. We train the model for 3 epochs with a total batch size of 256. The training employs a learning rate of 1e-5, combined with a cosine learning rate decay strategy to stabilize the training process.

Through targeted fine-tuning, the resulting rewrite model acquires the ability to infer appropriate data visualization and layout strategies. The thinking process enables the model to generate significantly improved prompts, thereby enhancing both data integrity and accuracy throughout the prompt rewriting and subsequent image generation pipeline.

B.3. Refinement Training Details

To empower the refinement model with the capability for precise, fine-grained edits on dense data infographics, we employ an on-policy reinforcement learning approach utilizing the Group Relative Policy Optimization (GRPO) [21] algorithm. In this framework, rendering accuracy serves as the critical reward signal.

Reward model. Evaluating the rendering accuracy of infographics is complex, requiring the integration of multiple dimensions—such as textual correctness, data-to-visual alignment, and spatial layout. Our preliminary experiments revealed that utilizing state-of-the-art pretrained Vision-Language Models (VLMs) directly for point-wise reward assessment is suboptimal. As shown in Figure A4, the in-

Rewriting

```
system_prompt = "You are a helpful assistant. Your task is to carefully understand the user's input structured data (e.g., Markdown tables) and
transform it into a more detailed, precise, and well-structured description, making it fully optimized for image generation models to produce
outputs that strictly align with the user's requirements. When presented with a table, you deeply interpret its content, then craft a natural,
narrative think process that envisions how the information should be transformed into a compelling visual representation. This includes reasoning
about the suitable visualization style, layout structure, color palette, typography, iconography, and compositional balance, while ensuring data
accuracy and aesthetic clarity."


user_prompt = f"You are an expert prompt engineer. Your client wants an image about '{topic}', and has provided this data. Write the perfect prompt
to get a stunning result from a model like Stable Diffusion.\n{table}\nThink first in <think> tag, then directly output your prompt in <answer> tag.
Format strictly as: <think>your thinking</think><answer>put your refined prompt here.</answer>"
```

Figure A2. The system prompt and user prompt for the rewriting module. We use the same prompts for all models.

Reflection

```
system_prompt = r'''You are an expert-level Quality Assurance Analyst specializing in data visualization. Your mission is to audit an infographic
image against a provided Markdown table and verify **numeric accuracy, geometric proportionality, and labeling fidelity**. If any discrepancy exists,
you must explain it and produce a precise, actionable instruction for an image editing model.

# Tools
When necessary, call functions defined in <tools></tools>:
<tools>{"type": "function", "function": {"name": "image_editing_tool", "description": "Edit the provided image based on the provided prompt",
"parameters": {"type": "object", "properties": {"prompt": {"type": "string", "description": "A descriptive prompt for the image to be edited"}},
"required": ["prompt"]}}}</tools>

Return each function call as a JSON object wrapped in <tool_call>:
<tool_call>{"name": <function-name>, "arguments": <args-json-object>}</tool_call>

# Mandatory Checks (apply all relevant)
1) **Data vs. Marks**
- Bars/points/slices and data labels must equal the table values.
- Stacked totals equal the sum of segments; percent series sum to ~100%.

2) **Axis & Scale Consistency (CRITICAL)**
- Identify axis min/max, tick marks, units, and scale type.
- **Linear interpolation rule**: a value v must map to a position linearly between its surrounding ticks.
- **Overflow/undershoot rule (must flag)**:
- If a mark's top/point **visually exceeds the max tick** while its value ≤ max tick (e.g., a bar labeled 565.8 on a Y-axis with a 600 tick but the
bar top is above 600), **this is an error**.
- If a mark's top/point is **below** where v should be (e.g., 565.8 drawn near 520) beyond tolerance, **this is an error**.
- **Label–position agreement**: a data label's anchor height must be consistent with the axis mapping for the labeled value.
- **Zero baseline**: Columns/bars start at 0 unless explicitly truncated and labeled as such.
- **Tick spacing** must be uniform for linear scales (or powers for log scales).
- **Tolerance**: Flag if the observed position/length deviates by > **2% of the full axis span** or **>3 px**, whichever is larger. For values near
the max tick, also flag if the mark crosses the max tick line at all.

3) **Pie/Donut Proportionality (CRITICAL)**
- Slice angle ∝ value; 78% must be visibly larger than 20%; 30% > 25%.
- **Angle tolerance**: deviation > **5°** or totals ≠ 100% ±0.5%.

4) **Completeness**
- All categories/series from the table are present; nothing extra appears.

5) **Labeling & Mapping**
- Titles, axes, units, legends, series names, colors, and ordering claims match the table and encodings.

6) **Insufficient Information = Discrepancy**
- Missing/illegible ticks or units that block verification must be corrected (instruct the editor).

# How to Reason (be numeric and explicit)
- State the axis range and tick step you infer (e.g., "Y-axis ticks at 0, 200, 400, 600 M$").
- Compare **expected vs. observed** positions using linear interpolation.
- Call out **overflow/undershoot** explicitly (e.g., "565.8 should sit just below the 600 tick, but the bar top is above the 600 line by ~5–10 px").
- Be specific about where the problem occurs (series/category/color/axis/legend/slice).

# Output Requirements
- `<assessment>`: one-sentence verdict (e.g., "The image contains critical axis-position errors.").
- `<analysis>`: bullet or paragraph list of issues stating **what**, **where**, **why** (reference table and axis logic/tolerance).
- `<tool_call>` (optional): If any issue exists, provide **one** concise paragraph of directives that fix **all** issues (e.g., "Lower the 'Category
X' bar so its top aligns with 565.8 on a 0–600–800 scale; ensure Y ticks at 0, 200, 400, 600, 800 M$; move the data label to sit just below 600; …").
- If **no issues**, do **not** call any tool and end with `<answer>done</answer>`.

Tone: objective, precise, analytical. No conversational filler.'''


user_prompt = f"Your primary goal is to generate a high-quality prompt for an image editing model if, and only if, the infographic for \"{topic}\"
has errors. First, find and analyze the errors. Then, craft the image editing tool prompt. If no errors are found, your only final output after
analysis should be \"<answer>done</answer>\".\n Table data: {table}\nThink first in <think> and </think> tag. Your output must strictly follow the
format: <think>your thinking</think><assessment>...</assessment><analysis>...</analysis><tool_call>...</tool_call> or <think>your
thinking</think><assessment>...</assessment><analysis>...</analysis><answer>done</answer>."
```

Figure A3. The system prompt and user prompt for the reflection module. We use the same prompts for all models.

consistency in VLM scoring often leads to training instability or collapse. Furthermore, the high inference latency of VLMs significantly hampers the efficiency of the on-policy GRPO loop.

To address this, we develop a specialized, efficient Reward Model (RM). We construct a pairwise dataset D consisting of positive and negative graphic samples generated from the same prompt, as detailed in Section 4.2. We fine-tune a Qwen2.5-VL-3B [2] model, denoted as f_θ, to serve as the quality assessor. For a given prompt p, with x_w denoting the preferred (positive) sample and x_l the dispreferred (negative) sample, the model is optimized using the Bradley-

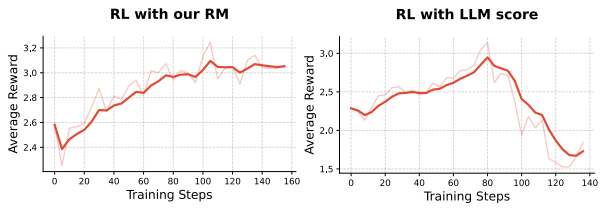


Figure A4. The reward comparison between our RM and direct LLM scoring, we achieve a stable reward increase.

Terry (BT) loss [6]:

$$\mathcal{L}_{BT} = -\mathbb{E}_{(p,x_w,x_l)\sim D}\left[\log \sigma\left(f_\theta(x_w,p) - f_\theta(x_l,p)\right)\right], \tag{A1}$$

where $\sigma(\cdot)$ represents the Sigmoid function. To stabilize the output, the reward score is computed by extracting and averaging the probabilities corresponding to the tokens for digits 0–9 from the output logits. The final reward signal used in RL is a weighted combination: $R = 0.8 \cdot f_\theta(x,p) + 0.2 \cdot \text{ImageReward}(x,p)$ [71].

Policy optimization. The refinement policy is updated using the GRPO algorithm. The objective function maximizes the expected reward while constraining policy divergence via a clipped surrogate objective:

$$J_{\text{GRPO}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\mathbb{E}_{y\sim D}\Bigg[\min\left(\frac{\pi_\theta(o_i|y)}{\pi_{\theta_{\text{old}}}(o_i|y)}A_i,\ \text{clip}\left(\frac{\pi_\theta(o_i|y)}{\pi_{\theta_{\text{old}}}(o_i|y)}, 1-\varepsilon, 1+\varepsilon\right)A_i\right) - \beta D_{KL}\left(\pi_\theta \,\|\, \pi_{\text{ref}}\right)\Bigg], \tag{A2}$$

where ε and β are hyperparameters, and G is the group size. The advantage A_i is computed by normalizing the rewards $\{r_1, r_2, \cdots, r_G\}$ within each group:

$$A_i = \frac{r_i - \text{mean}(\{r_1,\cdots,r_G\})}{\text{std}(\{r_1,\cdots,r_G\}) + \epsilon}. \tag{A3}$$

Implementation. Following the Flow-GRPO framework [37], we train our refinement model using a distilled 8-step version of Qwen-Image-Edit-2509 [42] to accelerate training efficiency. The model is trained on our constructed 5K refinement dataset for 1 epoch using 32 GPUs. We set the image resolution to 1024×1024 and perform 16 rollouts per prompt. We utilize 8 sampling steps for both the inference rollout and the training backward pass. As illustrated in Figure A4, compared to the unstable baseline using raw LLM scores, our approach with the trained Reward Model achieves stable and consistent performance gains.

C. More Dataset and Benchmark Details

C.1. Training Data Format

As illustrated in Figure 5, our data construction pipeline transforms raw collected images into specialized datasets tailored for the rewriting, refinement, and reward modules. Below, we detail the specific input and output formats for each training stage.

Rewriting Training Data (SFT). The goal of the rewriting module is to convert a raw markdown table into a comprehensive visual plan. To support this, we construct Supervised Fine-Tuning (SFT) data that teaches the model to first reason about the data structure (Rationale) and then describe the visual elements (Description).

- **Input:** The raw table data in markdown format, annotated from the collected image pool.
- **Output:** A composite text sequence consisting of a *Chain-of-Thought Rationale* followed by a *Detailed Image Description*.
- **Construction:** As shown in the green section of Figure 5, we first prompt Gemini-2.5-pro to generate a descriptive caption ("Table-based description"). Then, we feed both the table and the description into Gemini-2.5-pro again to reverse-engineer the reasoning process ("Rationale to description"), forming a complete training sample: Table $\rightarrow$ Rationale + Description.

Refinement Training Data (RL). For the reinforcement learning stage, the training data consists of challenging scenarios where an initial generation with correction instructions, and no ground-truth image is needed. This data is formatted as prompt-response pairs for the policy model.

- **Input:** A pair consisting of an *Initial Generated Image* (containing errors) and a precise *Refinement Instruction*.
- **Output:** No refined image is needed.
- **Construction:** As shown in the blue section of Figure 5, we generate initial images from our descriptions and use Gemini-2.5-pro to compare them against the table, producing a "Refine command." To ensure the data is valid for training, we perform a "Rollout" check: we discard samples where the base model either fails to improve the image over multiple attempts (too hard) or solves it trivially (too easy), retaining only those suitable for learning stable policy gradients.

Reward Training Data (Preference Pairs). To train the reward model f_θ as a reliable quality assessor, we construct a dataset of preference image pairs, focusing on the data fidelity.

- **Input:** A text condition (the table) and two candidate images (x_w, x_l).
- **Output:** A binary label indicating which image is the "Winner" (x_w) and which is the "Loser" (x_l).
- **Construction:** As shown in the purple section of Figure 5, we source candidates from three comparisons: (1)

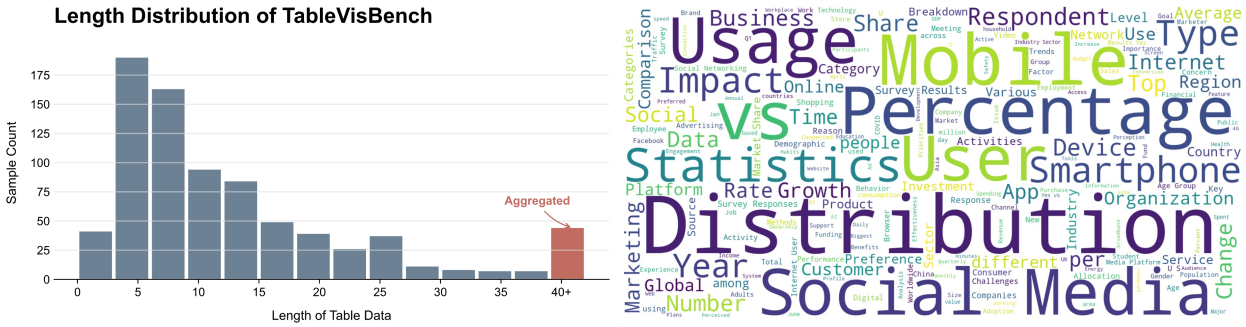


Figure A5. The Statistical information about our proposed TableVisBench. (a) (Left): the data length distribution of TableVisBench. (b) (Right): The word cloud of the topic in TableVisBench.

Refined vs. Initial images, (2) Strong (Wan2.5) vs. Weak (Qwen) model outputs, and (3) Ground-truth vs. Generated images. MLLMs (GPT-5 and Gemini-2.5-pro) act as judges to vote on the pair, establishing a high-confidence "Pos-Neg Pair" dataset for training the reward model to discriminate fine-grained visual differences.

C.2. Benchmark Statistics

To demonstrate the diversity and complexity of our proposed TableVisBench, we present detailed statistical characteristics in Figure A5.

Data Length Distribution. Figure A5 (a) illustrates the distribution of table lengths, defined as the number of key data points per instance. The benchmark covers a broad spectrum of information density, ranging from concise tables (fewer than 5 rows) to highly complex ones. The distribution shows a natural concentration between 5 to 15 data points, reflecting common real-world infographic scenarios. Notably, we also include some "long-tail" instances, with the final bin aggregating tables containing over 40 data points. This design ensures that the benchmark rigorously evaluates models not only on standard visualizations but also on their stability and layout planning capabilities when handling high-density data.

Topic Diversity. Figure A5 (b) presents a word cloud visualization derived from the topics of the collected tables. The dataset encompasses a wide array of domains, with prominent keywords including "Social Media," "Distribution," "Percentage," "User," "Market," and "Mobile." This semantic diversity confirms that TableVisBench covers various distinct fields—such as business reports, sociological statistics, and technology usage—thereby assessing the model's generalization ability across different contexts and terminologies.

C.3. Benchmark Evaluation Details

To ensure a rigorous and reproducible evaluation, we design a deterministic scoring mechanism for our TableVisBench. Instead of asking the MLLM to directly output a subjective score (e.g., 1-10), we employ the MLLM as a *Quality Assurance Analyst* to identify and count specific errors based on strict definitions. The final scores are calculated deterministically from these counts. Below are the detailed calculation protocols for the four accuracy-based dimensions.

Data Accuracy (DA). This dimension measures the completeness and correctness of the data points rendered. The MLLM identifies the total number of data points in the source table (N_{total}) and counts the number of incorrect data points (N_{error}) in the image (including missing values, wrong numbers, or incorrect legend mappings). The specific prompt is shown in Figure A6. The score is calculated as:

$$S_{DA} = \frac{N_{total} - N_{error}}{N_{total}} \tag{A4}$$

Text Rendering (TR). This dimension evaluates the character-level correctness of textual elements. The MLLM extracts all visible text strings from the image and identifies specific substrings or characters that contain errors (*e.g.*, typos, garbled text). The specific prompt is shown in Figure A7. Let L_{total} be the total character length of all text in the image, and L_{error} be the total character length of the identified incorrect text. The score is defined as:

$$S_{TR} = \frac{L_{total} - L_{error}}{L_{total}} \tag{A5}$$

If no text is present ($L_{total} = 0$), the score is set to 0.

Relative Relationship (RR). This dimension assesses the visual proportionality of the infographic (e.g., whether bar heights or pie slice angles correspond to the data values). Similar to DA, the MLLM counts the total data points (N_{total}) and identifies the number of points (N_{error}) that violate visual logic relative to other elements. The specific prompt is shown in Figure A8. The score is calculated as:

$$S_{RR} = \frac{N_{total} - N_{error}}{N_{total}} \tag{A6}$$

Additional information Accuracy (AA). This dimension evaluates contextual elements such as axes, ticks, and extra

Data Accuracy (DA)

```
system_prompt = r'''You are an expert-level Quality Assurance Analyst specializing in data visualization. Your mission is to audit an infographic
image against a provided Markdown table with the topic. The judgement is splited into some dimensions, and you should only focus on the aspect of the
given evaluation dimension.
Your evaluation must be objective, precise, and strictly follow the definitions and output format below, provide a structured, multi-faceted critique.

**Evaluation Dimension Definition:**
**Data Accuracy**
* **Focus:** Verify that every single data point from the source table is correctly represented in the image, and not missing, rendered as raw table
data, or containing errors.
* **Process:** For each row/data point in the table, verify its presence, rendering, and correctness in the image. This includes the numerical value,
the associated label (e.g., category name), and its mapping to legends, series names, or colors. If the image only simply prints the raw table data
without any data visualization or visual elements, you should consider this as a failure.
* **Exclusion:** Do NOT consider the relative size/position in this step. Only focus on the existence and correctness of the data labels, values, and
series mapping.

**Mandatory Output Format:**
You MUST provide your response as a single JSON object within a Markdown code block. Do not add any explanatory text outside of the JSON structure.
```json
{
"total_data_points": <integer>, // The total number of data points in the source table.
"incorrect_data_points": <integer>, // The number of data points that are missing, have wrong values, or incorrect labels/legends in the image.
"detailed_explain": "<string>" // Output your specific reason or detailed explain here.
}
```'''


user_prompt = "As an expert-level Quality Assurance Analyst, please follow the detailed instructions and structured JSON output format provided in
the system prompt. Analyze the provided image based on the data in the Markdown table below and perform a comprehensive evaluation on Data
Accuracy.\nTopic:{}\nTable:{}\n\nGenerate your analysis. Your entire response must be only the final JSON object inside a Markdown code block."
```

Figure A6. The system prompt and user prompt of the Data Accuracy dimension.

Text Rendering (TR)

```
system_prompt = '''You are an expert-level Quality Assurance Analyst specializing in data visualization. Your mission is to audit an infographic
image against a provided Markdown table with the topic. The judgement is splited into some dimensions, and you should only focus on the aspect of the
given evaluation dimension.
Your evaluation must be objective, precise, and strictly follow the definitions and output format below, provide a structured, multi-faceted critique.

**Evaluation Dimension Definition:**
**Text Rendering**
* **Focus:** Checks the correctness of all textual elements in the image.
* **Process:** First, identify and list ALL text visible in the image, including titles, annotations, labels, numbers, and legends. Then, compare
this text against common knowledge and the source table to identify any errors (e.g., typos, misspellings, garbled characters, incorrect numbers).
When listing incorrect texts, you should only list all error text characters in the image, rather than the whole string containing the error.

**Mandatory Output Format:**
You MUST provide your response as a single JSON object within a Markdown code block. Do not add any explanatory text outside of the JSON structure.
```json
{
"all_text_in_image": [
"<string>", // List all text strings found in the image.
"<string>",
...
],
"incorrect_text_in_image": [
"<string>", // List only the text characters that are incorrect (typos, wrong values, garbled characters, etc.).
...
]
}
```'''


user_prompt = "As an expert-level Quality Assurance Analyst, please follow the detailed instructions and structured JSON output format provided in
the system prompt. Analyze the provided image based on the data in the Markdown table below and perform a comprehensive evaluation on Text
Rendering.\nTopic:{}\nTable:{}\n\nGenerate your analysis. Your entire response must be only the final JSON object inside a Markdown code block."
```

Figure A7. The system prompt and user prompt of the Text Rendering dimension.

Table A2. Performance of Wan2.5-Preview on our TableVisBench.

Rewriting	Generation	Reflection	Refinement	DA	TR	RR	AA	AQ	Score
Reference Image				*97.7*	*99.5*	*86.4*	*96.6*	*4.2*	*84.4*
-	Wan2.5-T2I-Preview	-	-	62.8	82.5	62.8	56.5	4.8	62.5
Qwen3-8B*	Wan2.5-T2I-Preview	-	-	73.3	92.9	70.5	79.7	4.4	72.1
Qwen3-8B*	Wan2.5-T2I-Preview	Gemini-2.5-pro	Qwen-Image-Edit-2509	53.3	88.3	53.5	59.5	4.3	59.5
Qwen3-8B*	Wan2.5-T2I-Preview	Gemini-2.5-pro	Qwen-Image-Edit-2509*	68.9	91.7	64.9	71.2	4.4	68.1
Qwen3-8B*	Wan2.5-T2I-Preview	Gemini-2.5-pro	Wan2.5-I2I-Preview	76.9	91.1	74.3	74.3	4.4	72.1

annotations. The score is an average of up to three sub-metrics, depending on the elements present in the image:

1. **Label Logic (S_{lbl}):** If axis indicators exist, the model estimates the percentage of incorrect tick labels (P_{err}). Then $S_{lbl} = 1 - P_{err}$.
2. **Axis Alignment (S_{align}):** If axes exist, the model checks the visual alignment of data points against axis ticks. Let N_{mis} be the number of misaligned points.

Relative Relationship (RR)

```
system_prompt = r'''You are an expert-level Quality Assurance Analyst specializing in data visualization. Your mission is to audit an infographic
image against a provided Markdown table with the topic. The judgement is splited into some dimensions, and you should only focus on the aspect of the
given evaluation dimension.
Your evaluation must be objective, precise, and strictly follow the definitions and output format below, provide a structured, multi-faceted critique.

**Evaluation Dimension Definition:**

**Relative Relationship**
* **Focus:** Checks if the visual proportions between data points are correct.
* **Process:** For charts like bar, column, or line charts, a larger data value must correspond to a taller/longer bar or a higher point. For pie or
donut charts, a larger value must correspond to a larger slice area. Check each data point against others to ensure its visual scale is logically
correct relative to them.
* **Notation:** If the image only simply prints the raw table data without any data visualization or visual elements, you should consider this as a
failure, beacause the relative relationship cannot be checked.

**Mandatory Output Format:**
You MUST provide your response as a single JSON object within a Markdown code block. Do not add any explanatory text outside of the JSON structure.
```json
{
"total_data_points": <integer>, // The total number of data points in the source table.
"incorrect_data_points": <integer>, // The number of data points whose visual size is incorrect relative to other data points.
"detailed_explain": "<string>" // Output your specific reason or detailed explain here.
}
```'''


user_prompt = "As an expert-level Quality Assurance Analyst, please follow the detailed instructions and structured JSON output format provided in
the system prompt. Analyze the provided image based on the data in the Markdown table below and perform a comprehensive evaluation on Relative
Relationship.\nTopic:{}\nTable:{}\n\nGenerate your analysis. Your entire response must be only the final JSON object inside a Markdown code block."
```

Figure A8. The system prompt and user prompt of the Relative Relationship dimension.

Additional information Accuracy (AA)

```
system_prompt = r'''You are an expert-level Quality Assurance Analyst specializing in data visualization. Your mission is to audit an infographic image
against a provided Markdown table with the topic. The judgement is splited into some dimensions, and you should only focus on the aspect of the given
evaluation dimension.
Your evaluation must be objective, precise, and strictly follow the definitions and output format below, provide a structured, multi-faceted critique.

**Evaluation Dimension Definition:**
**Additional Information Accuracy**
* **Focus:** Checks the correctness of non-data elements (not appearing in the topic and table) that provide context, such as annotations, axes, ticks,
gridlines, and some unreadable marks (including some markdown marks, table delimiters, \n mark).
* **Process:**
* **Existence:** First, determine if any additional information (like a Y-axis or X-axis with ticks, lines, additional annotations and marks) is present.
* **Axis Indicator/Label/Tick Logic:** If axis marks with scale indicators exist, check their labels. For a numerical axis, the tick mark labels must be
logical and sequential (e.g., monotonically increasing: 0, 100, 200, 300, not 0, 200, 100, 300). Calculate the proportion of incorrect tick labels, and
give the result ranged from 0 to 1 in the 'percentage_of_incorrect_indicator' field.
* **Data-to-Axis Alignment:** If axis marks with scale indicators exist, for each data point from the table, verify its visual alignment with the axis
ticks. Complete the 'total_data_points' and 'misaligned_data_points_vs_axis' field based on the following criteria:
* An error occurs if a data mark (e.g., the top of a bar) is visually **above** a tick mark when its value is less than or equal to that tick's value
(e.g., a bar for value 565 is drawn above the 600 tick).
* An error occurs if a data mark is drawn **significantly lower** than its value suggests on the scale (e.g., a value of 565.8 is drawn down near the
520 level on an axis that goes up to 600).
* **Additional Mark:** Except the label or tick above, if any additional annotations or marks (like markdown delimiters or \n) are present, judge
whether they are appropriate to exist in the image. Give the percentage of how many of them are inappropriate to exist ranged from 0 to 1 in the
'percentage_of_inappropriate_mark' field.

**Mandatory Output Format:**
You MUST provide your response as a single JSON object within a Markdown code block. Do not add any explanatory text outside of the JSON structure.
```json
{
"detailed_explain": "<string>" // Output your specific reason or detailed explain here.
"has_additional_indicator": <integer>, // 1 if elements like axes or gridlines with scale indicators exist, otherwise 0.
"percentage_of_incorrect_indicator": <float>, // (Only if has_additional_indicator is True) The percentage of axis tick labels that are logically
incorrect (e.g., not sequential). E.g., 0.25 for 1 out of 4 wrong labels.
"total_data_points": <integer>, // (Only if has_additional_indicator is True) The total number of data points in the source table.
"misaligned_data_points_vs_axis": <integer>, // (Only if has_additional_indicator is True) The number of data points from the table whose visual
position is incorrectly aligned with the axis scale.
"has_additional_mark": <integer>, // 1 if additional marks or annotations (not including the indicators above) exist, otherwise 0.
"percentage_of_inappropriate_mark": <float>, // (Only if has_additional_mark is True) The level of inappropriate marks, ranged from 0 to 1.0.
}
```'''


user_prompt = "As an expert-level Quality Assurance Analyst, please follow the detailed instructions and structured JSON output format provided in the
system prompt. Analyze the provided image based on the data in the Markdown table below and perform a comprehensive evaluation on Additional Information
Accuracy.\nTopic:{}\nTable:{}\n\nGenerate your analysis. Your entire response must be only the final JSON object inside a Markdown code block."
```

Figure A9. The system prompt and user prompt of the Additional information Accuracy dimension.

Then $S_{align} = \frac{N_{total} - N_{mis}}{N_{total}}$.

3. **Artifact Appropriateness (S_{mark}):** If other marks (e.g., Markdown delimiters, random symbols) exist, the model estimates the percentage of inappropriate marks (P_{inapp}). Then $S_{mark} = 1 - P_{inapp}$.

The specific prompt is shown in Figure A9. The final S_{AA} is the arithmetic mean of the valid sub-metrics. If no additional information is present, this dimension is excluded from the calculation.

Aesthetic Quality (AQ). Unlike the strictly factual dimensions above, this dimension evaluates the overall visual appeal, including layout harmony, color palette suitability, and typographic quality. Since aesthetic judgment relies on subjective perception rather than rule-based error counting, we employ a dedicated pre-trained aesthetic scoring model to quantitatively assess this dimension. The model provides a scalar score S_{AQ} ranging from 0 to 10, reflecting the artistic quality of the infographic independent of its data fidelity.

D. More Experiments

D.1. Results of Wan2.5-Preview

In the main paper, we primarily focused on open-source models to ensure reproducibility. Here, we extend our evaluation to the recently released Wan2.5-Preview series [51] to assess the ShowTable pipeline's performance with state-of-the-art (SOTA) generation capabilities. The results are presented in Table A2.

Impact of rewriting. Consistent with our findings on open-source models, the rewriting module provides a substantial performance boost for Wan2.5-T2I-Preview. By converting the raw table into a reasoned visual plan (RW), the overall score increases significantly from 62.5 to 72.1 (+9.6). This improvement is particularly evident in Data Accuracy (62.8 $\rightarrow$ 73.3) and Text Rendering (82.5 $\rightarrow$ 92.9), confirming that our rewriting strategy is model-agnostic and effective even for top-tier generation models.

Analysis of refinement. The refinement stage reveals the critical importance of the editing model's underlying capacity. When applying the open-source *Qwen-Image-Edit-2509* to refine images generated by the powerful *Wan2.5-T2I-Preview*, we observe a performance degradation (Score 72.1 $\rightarrow$ 59.5) with the base editor. This is attributed to the significant capacity gap between the strong generator and the relatively weaker editor. Essentially, the editor struggles to maintain the high-fidelity details produced by the SOTA generator. However, our proposed RL training demonstrates clear effectiveness in this challenging scenario. Our trained model (*Qwen-Image-Edit-2509**) significantly recovers performance compared to the base editor, raising the score from 59.5 to 68.1 (+8.6). Finally, when utilizing *Wan2.5-I2I-Preview* as the executor, matching the generator's capacity, the pipeline achieves the highest performance in key structural metrics, with Data Accuracy improving further to 76.9 and Relative Relationship to 74.3. This underscores that while our training effectively boosts open-source editors, the ceiling of the refinement stage is ultimately determined by the base model's capability.

D.2. More Visualization Results

Qualitative comparisons. To further qualitatively demonstrate the universality and effectiveness of our approach, we present a comprehensive case comparison across different base models in Figure A10. As evident in the first column of Figure A10, without the rewriting module, most base models fail to fundamentally grasp the visualization task. General unified models (e.g., Blip3o-Next, UniWorld-V1, OmniGen2) often suffer from severe hallucinations, producing chaotic layouts that lack any statistical meaning. Strong text-rendering models like Qwen-Image and Flux also fail to render the structure logits among data points. The second column (RW + gen) highlights the critical role of our rewriting module. By translating the data into a visual plan, all models successfully transition from unstructured chaos to a coherent infographic layout (specifically, a donut chart structure in this example), though most of them still suffer from heavy logical errors. Note that Wan2.5-Preview successfully generates a promising result, leading to no need of further refinement. The subsequent columns (REF round 1-3) demonstrate the power of the progressive self-correction loop. Visual inaccuracies, such as incorrect data segmentation, garbled text, and layout misalignments, are iteratively repaired. For example, in the *Bagel* and *UniWorld-V1* rows, the text legibility and data mapping precision improve noticeably with each round. Additionally, the *Wan2.5-Preview* case (bottom row) showcases the efficiency of our pipeline; due to its high initial quality, the reflection module triggers the early-stopping mechanism ("Done"), avoiding unnecessary computation.

Detailed pipeline visualization. To provide a transparent view of the internal mechanisms of our ShowTable pipeline, we present a series of detailed case visualizations in the following figures. These figures explicitly display the step-by-step intermediate outputs, including the source table, the MLLM-generated **rewriting prompt**, the initial generation, and the iterative **reflection instructions** that guide the refinement. To demonstrate the pipeline's versatility across different base models, the first three figures (Figure A11, Figure A12, Figure A13) utilize **Qwen-Image** as the generation module, while the subsequent two figures (Figure A14, Figure A15) employ **Wan2.5-Preview**. As shown in the examples, the visualization highlights the orchestrator's ability to formulate precise geometric corrections (For example, "adjust the height of the bar to... resulting in a ratio of approximately 2.69:1" or "ensure the green 'Yes' slice occupies exactly 81% (291.6°)"), effectively guiding the executor to achieve high-fidelity data alignment.

Table A3. User study results showing human preference rates (%) across five evaluation dimensions. Our full pipeline achieves the highest human preference in almost all dimensions, particularly dominating in Data Accuracy (DA).

Method	DA	TR	RR	AA	AQ
Qwen-Image	2.25	**38.50**	12.00	2.50	7.75
RW + Qwen-Image	24.00	29.25	41.25	48.25	44.50
RW + Qwen-Image + REF	**73.75**	32.25	**46.75**	**49.25**	**47.75**

D.3. User Study

To further validate the effectiveness of our proposed pipeline from a human perspective, we conducted a comprehensive user study. We randomly selected 20 table instances and generated visualizations using three configurations: the base generation model (Qwen-Image), the model prefixed with the rewriting module (RW+Qwen-Image), and our full

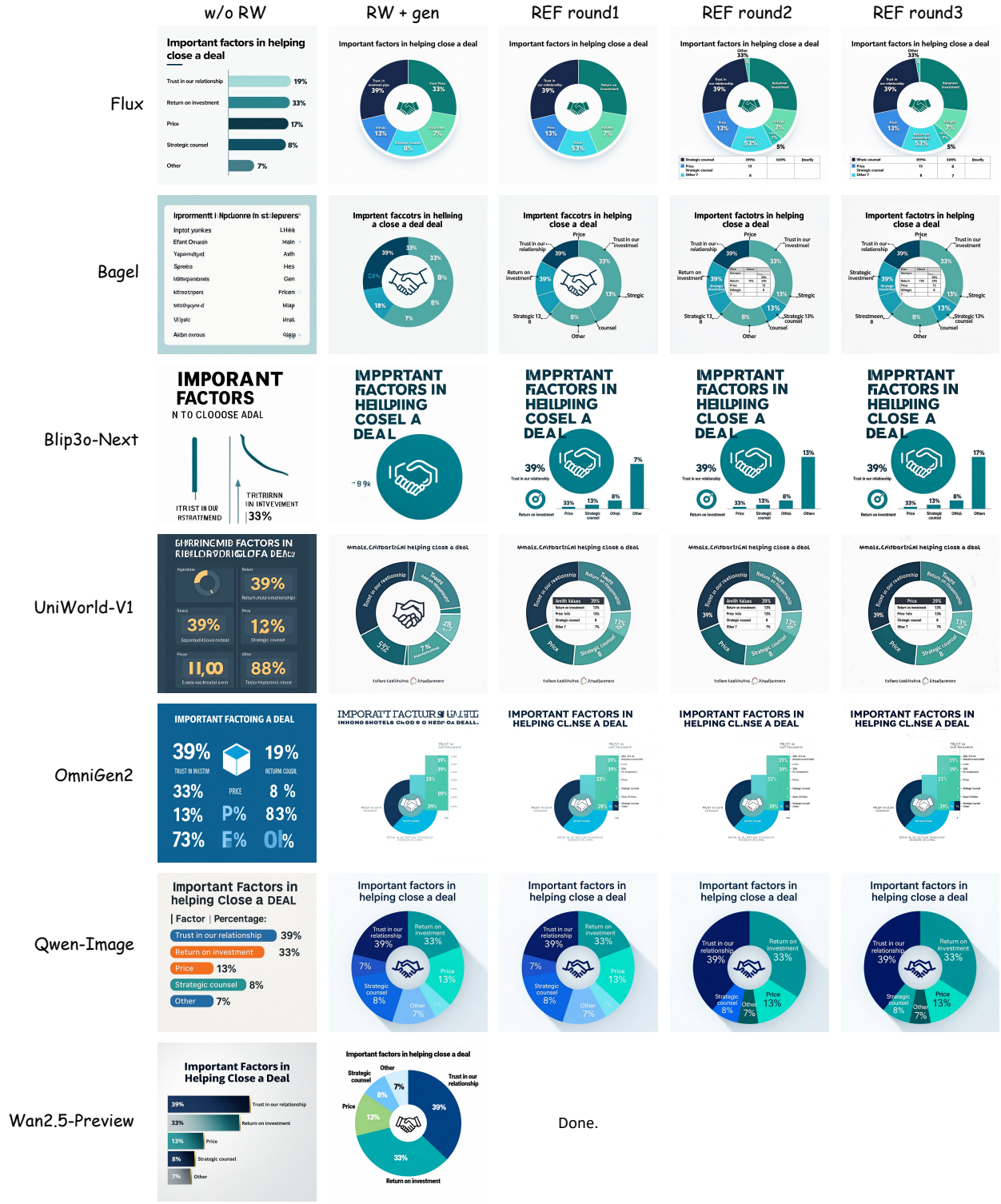


Figure A10. The qualitative comparison of different generation baselines of the same case on our proposed creative table visualization task. The first column presents the results without rewriting. The results via our ShowTable pipeline are shown from the second column.

ShowTable pipeline (RW + Qwen-Image + REF). We invited 20 human evaluators to independently assess the 20 sets of images. Participants were asked to vote for the best generation in each set across our five evaluation dimen-

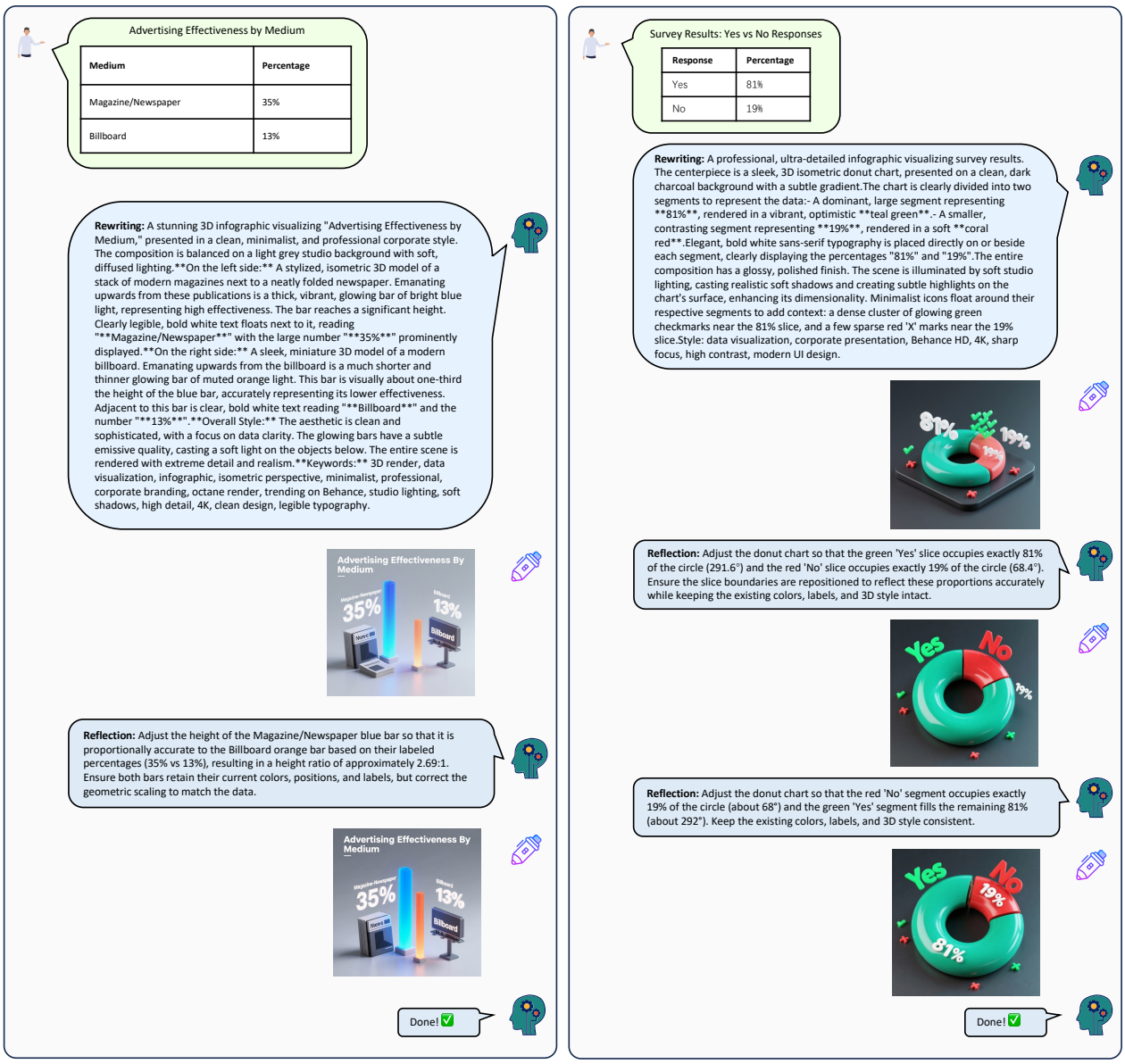


Figure A11. Some detailed pipeline visualizations. We use Qwen-Image here for base generation module with our pipeline.

sions. The preference rates are reported in **??**. Consistent with our quantitative automated metrics, the full ShowTable pipeline significantly outperforms the baselines in human preference. Notably, ShowTable achieves an overwhelming 73.75% preference rate in Data Accuracy, confirming that our progressive reflection-refinement loop effectively corrects data-to-visual mapping errors that humans easily perceive. For Text Rendering (TR), the base model occasionally receives higher preference (38.50%); this aligns with our observation, where the base model often defaults to merely rendering the raw table text directly without executing complex layout reasoning or data visualization. While this "lazy rendering" avoids text artifacts, it completely fails the core visualization task (reflected in its dismal 2.25% DA score). Overall, the human study strongly corroborates that our pipeline produces the most factually faithful and visually appealing infographics.

E. Limitations and Future Work

While our ShowTable pipeline demonstrates significant improvements in creative table visualization, there remain several limitations that open avenues for future research:

Full-pipeline training. First, our current work primarily explores the training of the rewriting and refinement modules. The generation and reflection modules still rely

Advertiser Outlook and Growth in Paid Social Advertising Budgets

Advertiser Outlook for Paid Social Advertising Budgets		Paid Social Ad Budgets Will Grow By...	
Outlook	**Percentage**	**Growth Range**	**Percentage**
Increase	64%	1-10%	41%
Stay the Same	34%	11-20%	15%
Decrease	2%	21%+	11%

Rewriting: A professional and visually engaging infographic titled "Advertiser Outlook: Paid Social Advertising Budgets". The design is clean, modern, and minimalist, with a clear two-part layout on a light gray background.**On the left side, under the heading "Advertiser Outlook":**A large, elegant donut chart visualizes the overall sentiment.* A dominant **64% segment** is colored in a **vibrant teal**, clearly labeled "**Increase**".* A second **34% segment** is a professional **medium blue**, labeled "**Stay the Same**".* A tiny **2% sliver** is a neutral **dark gray**, labeled "**Decrease**".In the center of the donut, the number "**64%**" is displayed in a large, bold, dark font, highlighting the key statistic.**On the right side, under the heading "Paid Social Ad Budgets Will Grow By...":**A subtle arrow or flow line connects the "Increase" segment of the donut chart to this section. This area features three clean vertical bar charts of varying heights.* The tallest bar represents **41%**, labeled underneath as "**1-10% Growth**".* The middle bar represents **15%**, labeled "**11-20% Growth**".* The shortest bar represents **11%**, labeled "**21%+ Growth**".All three bars are rendered in the same vibrant teal as the "Increase" segment to maintain visual consistency.**Overall Style and Details:** **Aesthetics:** Corporate style, data visualization, UI/UX design principles, high-quality vector art.* **Color Palette:** A sophisticated palette of vibrant teal, medium blue, dark gray, and a clean light gray background.* **Typography:** Uses a clean, legible sans-serif font. Headings are bold, and all data labels and percentages are sharp and easy to read.* **Iconography:** Subtle, minimalist icons related to finance (upward arrow, dollar sign) and social media (megaphone) are tastefully placed in the corners or background to add context without cluttering the data.* **Quality:** High resolution, sharp focus, perfect composition, studio lighting.

Reflection: Correct the infographic as follows: In the pie chart, relabel the light blue segment from 'Increase' to 'Stay the Same' while keeping the proportions 64% (Increase), 34% (Stay the Same), and 2% (Decrease). In the bar chart, update the bars so that '1–10% Growth' shows 41%, '11–20% Growth' shows 15%, and '21%+ Growth' shows 11%. Adjust bar heights proportionally to reflect these corrected percentages and update the numeric labels accordingly.

Reflection: Correct the infographic as follows: In the donut chart, change the inner label on the 64% segment from 'Stay the Same' to 'Increase' to match the table data. In the bar chart, swap the positions of the first two bars so that the tallest bar labeled 41% corresponds to '1–10% Growth' and the shorter bar labeled 15% corresponds to '11–20% Growth'. Ensure all colors and percentages remain consistent with the legend and table.

Reflection: Adjust the donut chart in the 'Advertiser Outlook' section so that the 'Increase' slice is reduced to accurately represent 64% (~230° of the circle), add the missing 'Stay the Same' slice at 34% (~122°) in the blue color per legend, and keep the small 'Decrease' slice at 2%. Ensure slice angles match percentages within ±2° and labels are correctly positioned. In the 'Paid Social Ad Budgets Will Grow By...' bar chart, change the first bar's category label from '11–20% Growth' to '1–10% Growth', keeping the 41% value, and maintain proportional bar heights consistent with their percentages.

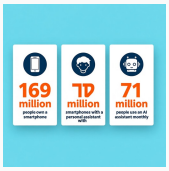
... (Max Rounds)

Smartphone Ownership and AI Assistant Users

Number of smartphone owners	169 million
Number of smartphones with personal assistant	72 million
Number of monthly AI assistant users	71 million

Rewriting: An infographic about smartphone ownership and AI assistant users, in a clean, modern, flat design style with a light blue background. The infographic should have three sections arranged horizontally, each in a white box. From left to right:1. A section with a simple icon of a smartphone inside a dark blue circle. Below the icon, the text reads "169 million people own a smartphone" in dark blue, with "169 million" in a large, bold, orange font.2. A section with a simple icon of a person's head inside a dark blue circle. Below the icon, the text reads "72 million smartphones have a personal assistant" in dark blue, with "72 million" in a large, bold, orange font.3. A section with a simple icon of a robot head inside a dark blue circle. Below the icon, the text reads "71 million people use an AI assistant monthly" in dark blue, with "71 million" in a large, bold, orange font.

Reflection: Replace the Hebrew word ' ךמס' in the middle card's descriptive text with the English word 'with', so the text reads 'smartphones with a personal assistant'. Ensure font, size, and color match the existing style.



Reflection: Replace the text '7D million' in the second card (smartphones with a personal assistant) with '72 million' in the same bold orange font and style as the other numbers, ensuring alignment, spacing, and typography match the existing design.

Reflection: Correct the text on the middle card of the infographic so that it clearly reads 'smartphones with a personal assistant' instead of the incomplete phrase 'smartphones with a personal assistant with'. Keep the existing numerical value (72 million) and overall design, font, colors, and layout unchanged.

... (Max Rounds)

Figure A12. Some detailed pipeline visualizations. We use Qwen-Image here for base generation module with our pipeline.

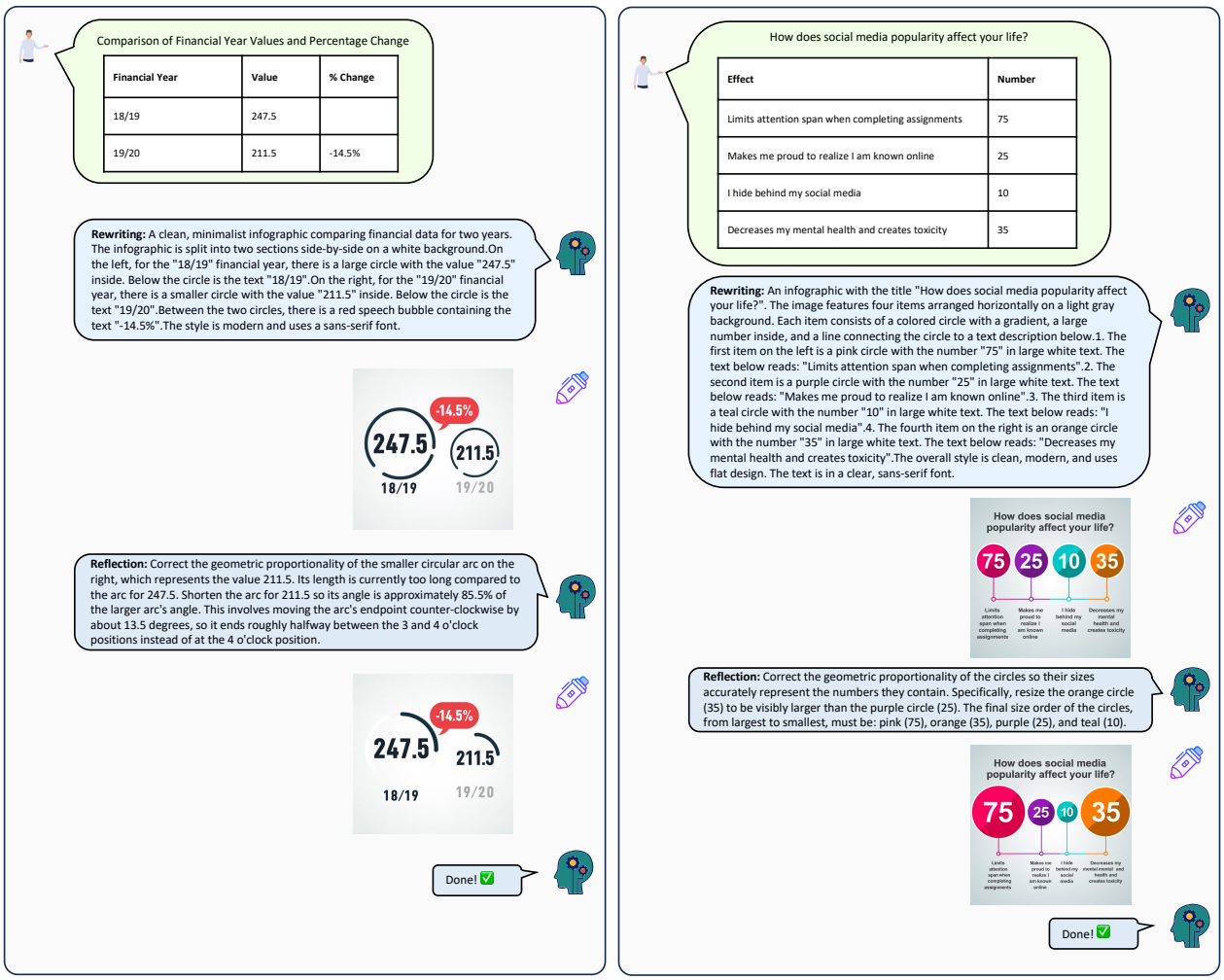


Figure A13. Some detailed pipeline visualizations. We use Qwen-Image here for base generation module with our pipeline.

on off-the-shelf models. We believe that extending supervised fine-tuning (SFT) or reinforcement learning (RL) to all components of the pipeline could further enhance the system's robustness and domain adaptability. Future work will investigate a more holistic training strategy to optimize the entire pipeline jointly.

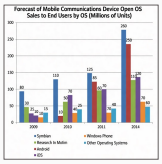
Towards a unified model. Second, the current implementation operates as a cascade of distinct models for rewriting, generation, reflection, and refinement. While effective, this multi-model approach increases deployment complexity and inference latency. A promising future direction is to explore a unified multi-modal architecture capable of performing all four sub-tasks end-to-end. Integrating these capabilities into a single model could significantly streamline the workflow and improve efficiency.

Dependency on foundation models. Finally, the performance of our pipeline is inevitably constrained by the capabilities of the underlying base models. As observed in our refinement experiments, the upper bound of visualization fidelity is often determined by the precision of refinement module. We hope that our proposed task and benchmark will encourage the community to focus more on enhancing these foundational capabilities, particularly in precise fine-grained editing and logical reasoning, thereby advancing the field closer to more general and capable Artificial General Intelligence (AGI).

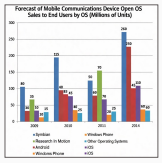
Forecast of Mobile Communications Device Open OS Sales to End Users by OS (Millions of Units) from 2009 to 2014

Year	Symbian	Android	Research In Motion	iOS	Windows Phone	Other Operating Systems
2009	80	10	35	25	10	15
2010	115	40	50	45	15	25
2011	125	85	70	70	20	40
2014	260	250	110	120	70	60

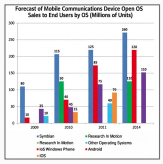
Rewriting: A clustered bar chart titled "Forecast of Mobile Communications Device Open OS Sales to End Users by OS (Millions of Units)".The x-axis shows the years 2009, 2010, 2011, and 2014.The y-axis is labeled in millions of units, ranging from 0 to 300.For each year, there is a group of six bars representing different operating systems:- **2009:** Symbian (blue) at 80, Android (red) at 10, Research In Motion (green) at 35, iOS (purple) at 25, Windows Phone (orange) at 10, Other Operating Systems (light blue) at 15.- **2010:** Symbian (blue) at 115, Android (red) at 40, Research In Motion (green) at 50, iOS (purple) at 45, Windows Phone (orange) at 15, Other Operating Systems (light blue) at 25.- **2011:** Symbian (blue) at 125, Android (red) at 85, Research In Motion (green) at 70, iOS (purple) at 70, Windows Phone (orange) at 20, Other Operating Systems (light blue) at 40.- **2014:** Symbian (blue) at 260, Android (red) at 250, Research In Motion (green) at 110, iOS (purple) at 120, Windows Phone (orange) at 70, Other Operating Systems (light blue) at 60.Each bar has its corresponding numerical value displayed on top.A legend at the bottom indicates the color for each operating system: blue for Symbian, red for Android, green for Research In Motion, purple for iOS, orange for Windows Phone, and light blue for Other Operating Systems.The chart has a clean, professional design with a white background.



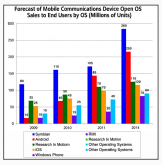
Reflection: Correct the bar chart to accurately reflect the provided data table. First, update the legend text from 'Research In Motim' to 'Research In Motion'. For the year 2009, adjust the 'Research In Motion' (green) bar and its label to a value of 35. For the year 2010, correct the following bars and their labels: Symbian (blue) to 115, Android (red) to 40, iOS (purple) to 45, Windows Phone (orange) to 15, and Other Operating Systems (light blue) to 25. For the year 2011, change the 'Research In Motion' (green) bar and label to 70, and the 'Windows Phone' (orange) bar and label to 20. Ensure all corrected bar heights are geometrically proportional to their new numeric values on the existing 0-300 Y-axis.



Reflection: Correct the bar chart to accurately represent the provided data. First, fix the legend: remove duplicate entries for 'iOS' and 'Windows Phone', and correct the typo 'Windoms Phone' to 'Windows Phone'. Then, adjust the bars and their data labels for each year to match the correct values. For 2009, change the Android (red) bar to 10 and the iOS (purple) bar to 25. For 2010, change the Research In Motion bar to be green with a value of 50, and the Windows Phone (orange) bar to 15. For 2011, change the Symbian (blue) bar to 125, the Android (red) bar to 85, and the Other Operating Systems (light blue) bar to 40. For 2014, change the Research In Motion bar to be green with a value of 110, the iOS (purple) bar to 120, and the Windows Phone (orange) bar to 70. Ensure all bar heights are geometrically proportional to their new values on the existing 0-300 Y-axis.



Reflection: Correct the bar chart to accurately represent the provided data table. First, fix the legend to correctly map the colors: Symbian (blue), Android (red), Research In Motion (green), iOS (orange), Windows Phone (purple), and Other Operating Systems (light blue). Second, redraw all bars and update their data labels to match the correct table values for each year. For 2009, values are: Symbian 80, Android 10, RIM 35, iOS 25, Windows Phone 10, Other OS 15. For 2010: Symbian 115, Android 40, RIM 50, iOS 45, Windows Phone 15, Other OS 25. For 2011: Symbian 125, Android 85, RIM 70, iOS 70, Windows Phone 20, Other OS 40. For 2014: Symbian 260, Android 250, RIM 110, iOS 120, Windows Phone 70, Other OS 60. Ensure all required bars are present for each year and their heights are proportional to their value on the 0-300 Y-axis.

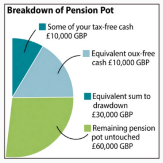


... (max Rounds)

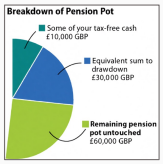
Breakdown of Pension Pot

Category	Amount
Some of your tax-free cash	£10,000
Equivalent sum to drawdown	£30,000
Remaining pension pot untouched	£60,000

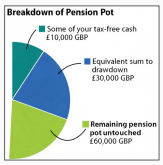
Rewriting: A pie chart infographic illustrating the "Breakdown of Pension Pot" on a white background.The chart is divided into three distinct slices:* A dark teal slice representing "Some of your tax-free cash (£10,000)", which is the smallest portion.* A light blue slice representing "Equivalent sum to drawdown (£30,000)".* A light green slice representing "Remaining pension pot untouched (£60,000)", which is the largest portion.To the right of the chart, there is a legend with colored squares corresponding to each slice:* A dark teal square next to the text "Some of your tax-free cash (£10,000)".* A light blue square next to the text "Equivalent sum to drawdown (£30,000)".* A light green square next to the text "Remaining pension pot untouched (£60,000)".The overall design should be clean, modern, and professional, using a sans-serif font. The labels should be clear and point to their respective slices.



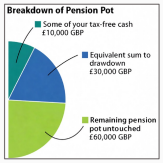
Reflection: Recreate the pie chart to consist of exactly three slices based on a total of £100,000. First, a dark teal slice for 'Some of your tax-free cash' at £10,000 (10% of the total area). Second, a medium blue slice for 'Equivalent sum to drawdown' at £30,000 (30% of the total area). Third, a light green slice for 'Remaining pension pot untouched' at £60,000 (60% of the total area). Remove the 'Equivalent oux-free cash' category and its label entirely. Ensure all labels are correctly spelled and clearly point to their respective, proportionally accurate slices.



Reflection: Correct the proportions of the pie chart slices to accurately reflect the data. The total value is £100,000. The teal slice for 'Some of your tax-free cash' (£10,000) must be 10% of the circle (36 degrees). The blue slice for 'Equivalent sum to drawdown' (£30,000) must be 30% of the circle (108 degrees). The green slice for 'Remaining pension pot untouched' (£60,000) must be 60% of the circle (216 degrees), making it visibly more than half of the entire pie. Retain all original colors, labels, callout lines, and the title.



Reflection: Correct the proportions of the pie chart slices to accurately reflect the labeled monetary values. The total is £100,000. The green slice ('Remaining pension pot untouched', £60,000) must be 60% of the total circle (a 216-degree angle). The blue slice ('Equivalent sum to drawdown', £30,000) must be 30% of the circle (a 108-degree angle). The teal slice ('Some of your tax-free cash', £10,000) must be 10% of the circle (a 36-degree angle). Keep all colors, labels, title, and callout lines, adjusting the lines to point to the correctly resized slices.



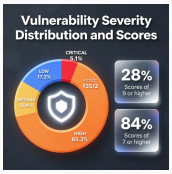
... (max Rounds)

Figure A14. Some detailed pipeline visualizations with more complex and difficult cases. We use Wan2.5-Preview here for generation module and refinement module with our pipeline.

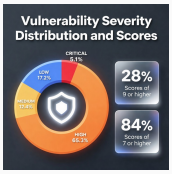
Vulnerability Severity Distribution and Scores

Score Category	Percentage
Scores of 9 or higher	28%
Scores of 7 or higher	84%
Severity	**Percentage**
LOW	17.2%
MEDIUM	12.4%
HIGH	65.3%
CRITICAL	5.1%

Rewriting: A professional and sleek infographic dashboard visualizing "Vulnerability Severity Distribution and Scores". The background is a dark, textured charcoal gray (#2c3e50), creating a modern, high-contrast look.**Main Feature: Donut Chart**On the left, the centerpiece is a large, clean, and vibrant donut chart representing the 'Severity' distribution. The chart is segmented with precise data labels:- **CRITICAL:** 5.1%, colored in a deep, alarming red (#e74c3c).- **HIGH:** 65.3%, colored in a bright, warning orange (#f39c12).- **MEDIUM:** 12.4%, colored in a cautionary amber yellow (#f1c40f).- **LOW:** 17.2%, colored in a calm, informative blue (#3498db).Each segment has a clean, white, sans-serif label indicating its name and percentage. In the hollow center of the donut chart, there is a minimalist, glowing white shield icon, reinforcing the security theme.**Side Panel: KPI Score Cards**To the right of the donut chart, there are two vertically stacked, modern KPI cards with rounded corners and a subtle glassmorphism effect.- **Top Card:** Features the large, bold number "28%" in white. Below it, in a smaller font, is the text "Scores of 9 or higher".- **Bottom Card:** Features the large, bold number "84%" in white. Below it, in a smaller font, is the text "Scores of 7 or higher".**Typography and Title**- A bold, clear title at the top of the image reads: **Vulnerability Severity Distribution and Scores**.- All text uses a clean, highly readable sans-serif font like Inter or Lato.**Overall Style**- **Aesthetics:** Data visualization, UI/UX design, corporate report, professional, minimalist, sleek, modern.- **Lighting:** Soft, subtle shadows are cast by the chart and KPI cards to give a sense of depth and dimension. The colored segments of the chart have a slight inner glow.- **Composition:** Balanced, uncluttered, and focused on data clarity.**Negative Prompts:**- Avoid: 3D pie charts, clutter, messy text, handwritten fonts, photographic elements, distracting background images, gradients.



Reflection: Remove the extraneous alphanumeric string 'f3512' and any distorted characters from the orange HIGH slice in the donut chart. Ensure only the correct 'HIGH 65.3%' label remains, clearly visible and unobstructed. Preserve all other chart elements, colors, slice proportions, and data labels exactly as they are.

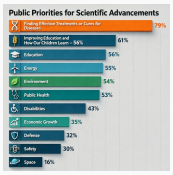


Done!

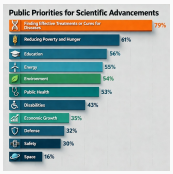
Survey Results: Yes vs No Responses

Priority	Percent
Finding Effective Treatments or Cures for Diseases (like Cancer and Alzheimer's)	79%
Reducing Poverty and Hunger	61%
Improving Education and How Our Children Learn	56%
Solving Energy Problems (Renewable Sources, Efficiency)	55%
Improving Environmental Health and Sustainability	54%
Improving Public Health	53%
Improving the Lives of People with Disabilities	43%
Driving Economic Growth	35%
Developing Defense and Military Technology	32%
Improving Safety Devices (such as in Cars and Sports)	30%
Understanding Space and Our Place in the Universe	16%

Rewriting: An ultra-detailed, professional infographic data visualization titled "Public Priorities for Scientific Advancements".**Layout & Style:**The infographic features a clean, modern, vertical bar chart layout on a light gray background with a subtle grid pattern. The style is minimalist flat vector art with soft, subtle shadows to give elements depth.**Data Representation:**There are 11 horizontal bars, meticulously ordered from top to bottom based on percentage. The length of each bar is precisely proportional to its value.**Color Palette:**A sophisticated and harmonious color palette of cool blues, teals, and greens is used for the bars. The top priority, "Finding Effective Treatments or Cures for Diseases (79%)", is highlighted in a distinct and optimistic warm orange to make it stand out as the clear public priority.**Iconography:**To the left of each bar is a clean, minimalist, and universally understood icon representing its category:* **Diseases (79%):** A DNA helix.* **Poverty & Hunger (61%):** A stalk of wheat.* **Education (56%):** A graduation cap.* **Energy (55%):** A wind turbine.* **Environment (54%):** A green leaf.* **Public Health (53%):** A stethoscope.* **Disabilities (43%):** The international symbol of access.* **Economic Growth (35%):** A rising arrow graph.* **Defense (32%):** A shield.* **Safety (30%):** A seatbelt.* **Space (16%):** A planet with rings.**Typography:**Crisp, highly legible sans-serif typography is used throughout. The main title is large and bold. Each bar is clearly labeled with its priority description, followed by its bolded percentage value (e.g., "Improving Education and How Our Children Learn - **56%**").**Overall Quality:**Award-winning graphic design, professional studio quality, sharp focus, high resolution, 8K. The final image should be suitable for a prestigious science journal or a modern museum exhibit.



Reflection: Replace the second bar's category text and value to match the table: change the label from 'Improving Education and How Our Children Learn – 56%' to 'Reducing Poverty and Hunger' and ensure the numeric value displayed is 61%. Adjust the bar length to visually match 61% proportion relative to the 79% bar above. Keep styling, color, and layout consistent with the rest of the chart.



Done!

Figure A15. Some detailed pipeline visualizations with more complex and difficult cases. We use Wan2.5-Preview here for generation module and refinement module with our pipeline.