# Supplementary Materials: NovaChart: A Large-scale Dataset towards Chart Understanding and Generation of Multimodal Large Language Models

## 1 Details of NovaChart

### 1.1 Chart Types

We present all 18 types of charts and their descriptions in Table 3. Additionally, we present an example every chart type in Figure 1.

### 1.2 Tasks Definition

We list all 15 chart understanding and generation tasks ,and their corresponding definitions and examples in Table 4.

### 1.3 Source Data Selection

The suitable attributes combinations for every chart types mentioned in data curation are listed in Table 1.

## 2 Training Details

The model architectures of the three models we fine-tune are presented in Table 2. We fine-tune these models on the training split of NovaChart for 2 epochs (1670 steps) with a global batch size of 1024 and utilize LoRA[2] on their LLM Branch for parameter-efficient tuning. The whole training process is conducted on 8 NVIDIA A800 GPUs. Deepspeed ZerO is applied for parallelized training. For LLaVA-v1.5, we set the learning rate to 2e-5 and the warm-up ratio to 0.03. For LoRA settings, the $\alpha$ and $r$ are set to 64 and 64. For InternLM-XComposer, we set the learning rate to 5e-5 and the warm-up ratio to 0.01. $\alpha$ and $r$ of LoRA are 64 and 64, respectively. For Qwen-VL-Chat, we set the learning rate to 1e-5 and the warm-up ratio to 0.03. We set $\alpha$ of LoRA at 16, and $r$ at 64.

## 3 Evaluation Metrics

In our evaluation, chart data understanding and chart visual understanding tasks are assessed through simple question-answer formats.

For chart data understanding and chart visual understanding tasks, responses from the model include numeric, strings, sequences, and mappings. We define numeric and string responses as atomic responses, and sequence and mapping responses are composite responses since they are composed of atomic ones. For Chart summarization and analysis, and Chart generation tasks, model response are generally open-ended, involving chain of thoughts, free-form answers, and python code. During evaluation, we include necessary output guidance in the prompts (e.g., "Please respond with a short answer without any preambles or analysis. ") to ensure right output format. To accurately and efficiently evaluate model performance on NovaChart, we design the following evaluation scheme for atomic, composite and open-ended responses. In Table 5, we present the response format for different tasks along with their corresponding evaluation methods. In subsequent sections, we assume the parsed model output is $y$ and the ground truth is $\hat{y}$.

**Table 1: Attribute combinations of different chart types use in data curation. U, N, C, E are abbreviation for Unique-Numeric, Numeric, Categorical and Enumerable attributes, respectively. The notation $X \times k$ indicates that we select $k$ distinct attributes of the $X$ attribute in this category of chart. (n) is ranged from 3 to 6.**

| Chart Type | Attribute Combinations |
|---|---|
| Single-class Line Plot | $U \times 1 + N \times 1$ |
| Multi-class Line Plot | $U \times 1 + N \times 1 + C \times 1$ |
| Single-class Scatter Plot | $N \times 2$ |
| Multi-class Scatter Plot | $N \times 2 + C \times 1$ |
| Single-class Bar Plot | $E \times 1$ |
| Multi-class Bar Plot | $E \times 1 + C \times 1$ |
| Univariate Histogram | $N \times 1$ |
| Bivariate Histogram | $N \times 2$ |
| Correlation Heatmap | $N \times (n)$ |
| Pie Chart | $E \times 1$ |
| Ring Chart | $E \times 1$ |
| Rose Chart | $E \times 1$ |
| Radar Chart | $N \times (n)$ |
| Box Plot | $N \times 1$ |
| Sankey Chart | $E \times 2$ |

**Table 2: Architectures of baseline MLLMs.**

| Models | Parameters | LLM Branch | Visual Branch |
|---|---|---|---|
| LLaVA-v1.5 | 13.4B | Vicuna-13B | CLIP ViT-L/14@336px |
| InternLM-XComposer | 8.2B | InternLM-Chat-7B | EVA-CLIP-G/14 |
| Qwen-VL-Chat | 9.6B | Qwen-7B | OpenCLIP ViT-G/14 |

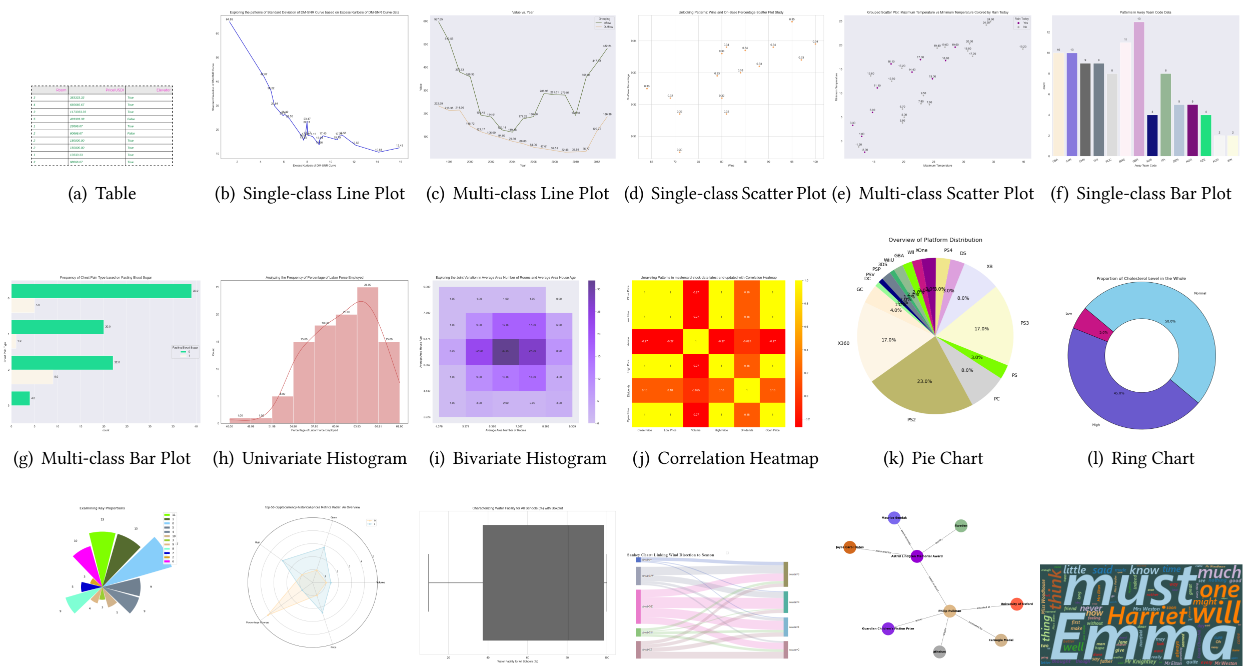
### 3.1 Atomic Responses

Atomic responses include string and numeric responses. For string outputs, we adopt the Exact Match (EM) metric, where the model's output $y$ is considered accurate only if it matches $\hat{y}$ exactly. Formally, this can be expressed as:

$$\text{score}_{\text{string}}(y, \hat{y}) = [y = \hat{y}]$$

where $[\cdot]$ is the indicator function, and $[a = b]$ indicates that strings $a$ and $b$ are directly identical.

For numeric outputs, inspired by discussions in DePlot [4] about RNSS, we apply a smoothing adjustment to the relative distance to accommodate cases where $\hat{y}$ is close to zero. Formally, the smoothed



(a) Table (b) Single-class Line Plot (c) Multi-class Line Plot (d) Single-class Scatter Plot (e) Multi-class Scatter Plot (f) Single-class Bar Plot

(g) Multi-class Bar Plot (h) Univariate Histogram (i) Bivariate Histogram (j) Correlation Heatmap (k) Pie Chart (l) Ring Chart

(m) Rose Chart (n) Radar Chart (o) Box Plot (p) Sankey Chart (q) Knowledge Graph (r) Word Cloud

**Figure 1: The examples of chart types covered by NovaChart.**

relative distance and the corresponding similarity score are expressed as:

$$\mathrm{D}(y, \hat{y}) = \frac{|y - \hat{y}|}{|\hat{y}| + \delta}$$

$$\mathrm{score}_{\mathrm{numeric}}(y, \hat{y}) = \max(1 - \mathrm{D}(y, \hat{y}), 0)$$

where $\delta = 10^{-2}$ is a smoothing factor to prevent division by zero when $\hat{y} \approx 0$; $|\cdot|$ denotes the absolute value.

## 3.2 Composite Responses

Composite responses include sequence responses (e.g., a list of data points extracted from a line chart) and mapping responses (e.g., a mapping between each class and their respective pie chart sector colors). Note that elements in a sequence and values in a mapping may not necessarily be atomic but could be another composite one. Therefore, we need to design a recursive scoring standard to cover all possible cases.

For sequences, we refer to the Levenshtein distance [3]: legitimate operations on a sequence include adding an element, deleting an element, or replacing an element. Unlike the classic Levenshtein distance, although the cost of adding or deleting an element is a fixed value of 1, the cost of replacing an element is modified according to the similarity of the elements before and after replacement. We name this modified Levenshtein distance the Adaptive Levenshtein Distance (ALD). Formally, ALD and the corresponding sequence similarity score can be recursively expressed as:

$$\mathrm{ALD}_{y,\hat{y}}(i,j) = \begin{cases} 0, & \text{if } i = j = 0 \\ i, & \text{if } i \neq 0 \text{ and } j = 0 \\ j, & \text{if } i = 0 \text{ and } j \neq 0 \\ \min \begin{cases} \mathrm{ALD}_{y,\hat{y}}(i-1, j) + 1 \\ \mathrm{ALD}_{y,\hat{y}}(i, j-1) + 1 \\ \mathrm{ALD}_{y,\hat{y}}(i-1, j-1) + 1 - \mathrm{score}(y_i, \hat{y}_j) \end{cases} & \text{else} \end{cases}$$

$$\mathrm{score}_{\mathrm{sequence}}(y, \hat{y}) = 1 - \frac{\mathrm{ALD}_{y,\hat{y}}(\|y\|, \|\hat{y}\|)}{\|y\| + \|\hat{y}\|}$$

where $\mathrm{ALD}_{y,\hat{y}}(i,j)$ represents the ALD value for the first $i$ elements of $y$ and the first $j$ elements of $\hat{y}$; $\|\cdot\|$ denotes the length of a sequence; $\mathrm{score}(y_i, \hat{y}_j)$ represents the similarity score between the $i$-th element of $y$ and the $j$-th element of $\hat{y}$, which is recursively calculated according to the types of them (it is a fixed value of 0 in case of different types).

For mappings, the keys are fixed as strings, while the values can be a number, string, or another composite type. Since we can consider a mapping as a set of key-value pairs, referring to the SCRM metric in StructChart [5] , we define the Adaptive Jaccard Distance (AJD), which can be expressed as follows:

First, we define $K_y$ and $K_{\hat{y}}$ as the key sets of $y$ and $\hat{y}$, respectively, and $y_k$ and $\hat{y}_k$ represent the values mapped to by $y$ and $\hat{y}$ under key $k$. Next, we calculate the intersection and symmetric difference of $K_y$ and $K_{\hat{y}}$. Each key $k$ in the symmetric difference contributes 1 to the AJD, while each key $k$ in the intersection contributes 1 minus the similarity score between $y_k$ and $\hat{y}_k$. Formally, AJD and the corresponding mapping similarity score can be expressed as:

$$\text{AJD}(y, \hat{y}) = \|K_y \oplus K_{\hat{y}}\| + \sum_{k \in K_y \cap K_{\hat{y}}} (1 - \text{score}(y_k, \hat{y}_k))$$

$$\text{score}_{\text{mapping}} = 1 - \frac{\text{AJD}(y, \hat{y})}{\|K_y \cup K_{\hat{y}}\|}$$

where $\oplus$ indicates the symmetric difference of two sets, $\|\cdot\|$ denotes the size of a set, and $\text{score}(y_k, \hat{y}_k)$ represents the similarity score between the values mapped to by $y$ and $\hat{y}$ under key $k$.

## 3.3 Open-ended Responses

Evaluating open-ended responses, such as lengthy textual replies or code generation, poses significant challenges in terms of accuracy. Therefore, following ChartLlama [1], we utilize GPT-4 as the evaluation model to assess the quality of model responses from multiple dimensions. Furthermore, we adopt a single-blind testing methodology, where GPT-4 serves as the evaluator but remains unaware of the true identities of the models being assessed. Instead, the models are anatomized and referred to only as Model A and Model B (see Figure 2 and 3). This approach ensures that GPT-4 does not develop biases based on model identity. Furthermore, to prevent any positional bias, the order of Model A and Model B is randomized in each evaluation instance.

For evaluations involving long text responses, such as chart analysis and summarization, GPT-4 is required to assess the responses based on two dimensions: correctness and meaningfulness. These dimensions are weighted in a 3:1 ratio, and then re-scaled to the range of 0 to 1, emphasizing correctness as the more critical factor in determining the quality of the model outputs, while still considering meaningfulness to avoid overly conservative but correct responses.

In the realm of code generation tasks, such as generating charts, GPT-4 similarly evaluates the outputs based on correctness and quality, following the same weighting rationale. Detailed scoring criteria and the complete GPT-4 evaluation prompt templates for both types of tasks are illustrated in Figure 2 and 3, where $ serves as a placeholder to be replaced by the actual content during the evaluation process.

## 4 Supplementary Experimental Results

We display detailed experimental results across different tasks in Tables 6, 7, 8, and 9, and for different chart types in Table 10. Note that to calculate the average score of different chart types in Table 10, we first determined the average score for each task within that chart type and then computed the overall average to mitigate the impact of varying task counts. Some column names are abbreviated. The full task names and chart type names can be found in 3 and 4.

**Table 3: Chart types and their descriptions.**

| Chart Type | Description |
|---|---|
| Table | A grid that displays information in rows and columns, often used for numerical and categorical data comparison. |
| Single-class Line Plot | A chart that displays a series of data points connected by straight line segments, representing one variable over a continuous variable. |
| Multi-class Line Plot | Similar to a single-class line plot, but compares multiple series on the same chart, each representing a different category. |
| Single-class Scatter Plot | A chart that uses dots to represent individual pieces of data in two dimensions, typically used to observe and show relationships between two numeric variables. |
| Multi-class Scatter Plot | A scatter plot that includes data points from multiple categories, often color-coded to show distinctions between categories. |
| Single-class Bar Plot | A chart that represents data with rectangular bars with lengths proportional to the values they represent, featuring a single category. |
| Multi-class Bar Plot | A bar plot that displays multiple groups of data side by side, with each group representing a different category. |
| Univariate Histogram | A type of bar chart that represents the distribution of a single variable by dividing the data into bins and counting the number of observations in each bin. |
| Bivariate Histogram | A histogram that displays the distribution of two variables simultaneously, using a grid of bins and color or shading to represent counts. |
| Correlation Heatmap | A graphical representation of data where individual values contained in a matrix are represented as colors, used to show correlation between variables. |
| Pie Chart | A circular chart divided into sectors, each representing a proportion of the total. |
| Ring Chart | Similar to a pie chart, but with a central hole, focusing on comparing the parts of a whole without a central point. |
| Rose Chart | A circular chart with categorical data points extending outward from the center, with the length of each 'petal' proportional to the magnitude. |
| Radar Chart | A graphical method of displaying multivariate data in the form of a two-dimensional chart of three or more quantitative variables represented on axes starting from the same point. |
| Box Plot | A standardized way of displaying the distribution of data based on a five-number summary: minimum, first quartile, median, third quartile, and maximum. |
| Sankey Chart | A flow diagram in which the width of the arrows is proportional to the flow rate, showing how quantities flow from one set of values to another. |
| Knowledge Graph | A visual representation of complex information using nodes and links to display how different entities and concepts are interconnected. |
| Word Cloud | A visual representation of text data where the size of each word indicates its frequency or importance in the document. |

**Table 4: All chart understanding and generation tasks with definitions and examples included in NovaChart.**

| Task | Description | Example |
|---|---|---|
| Data Identification | Determine if the given statement about a data point in the chart is correct. | The value of instance_id=[2] and feature=[NO. OF Internet Plans] is 3.6154, yes or no? Choose the appropriate option for the above question from the given choices: [Yes, No]. |
| Data Comparison | Compare numerical values for two data points in the chart. | For feature=[High Explosives Weight (Tons)], The value of instance_id=[2] is greater than the value of instance_id=[3], yes or no? Choose the appropriate option for the above question from the given choices: [Yes, No]. |
| Data Extraction with Condition | Extract numerical data with a specified condition from the chart and present it in a certain format. | Extract the numerical data points from the chart. You need to extract the data points between x=[71.23] and x=[102.79]. Your answer should be formatted as a list of the same length as the number of data units, where each item contains two numbers for the x- and y-coordinates of the data unit. The list should be sorted by x-coordinate. |
| Data Referring | Identify and report a specific value according to the textual referring. | Determine the numerical value of lower bound from the boxplot. Answer with a single number. |
| Color Recognition | Identify and report the color(s) used in the chart. | Extract and present the color information for each class depicted in the chart. Your answer should be formatted as a dict: for each key-value pair, the key should be the label of class, and the value is the respective color of this class. |
| Style Detection | Determine the preset style used in the chart. | Determine the likely seaborn style used in this image. |
| Chart Classification | Identify the type of chart based on its visual characteristics and structure. | What kind of chart does this visual data representation belong to? Select the correct solution: [multi-hue bar plot, word cloud, pie chart, bivariate histogram, multi-class line plot]. |
| Visual Elements Retrieval | Identify specific visual elements or features within the chart. | Does this histogram exhibit a KDE curve? Select the appropriate answer from the provided options for the question above: A. Yes B. No |
| Text Extraction | Extract and report specific text from the chart. | What is the designated label for the y-coordinate? Your answer should not include any leading preamble words or other content. |
| Chart Pattern Recognition | Analyze and describe the pattern or trend evident in the chart. | Identify the pattern that best suits the image displayed. Offer a step-by-step, detailed explanation in your response. |
| Chart Analysis | Interpret and explain the information conveyed by the chart, detailing the insights it provides. | What information or knowledge does this image convey? Provide a comprehensive answer, detailing each step thoroughly. |
| Chart Summarization | Provide a concise summary of the key points and information presented in the chart. | Please offer a brief summary of the image in one paragraph. |
| Table to Chart Generation | Write code to create a chart from the table using specific chart types. | Can you guide me in extracting table data and generating a code to visualize it using single-class line plot? |
| Chart Type Conversion | Write code to convert an existing chart into a different chart type. | Compose a code that transforms the chart into multi-class bar plot. |
| Chart Blueprint | Design a function that creates the chart based on given chart image. | Design a function with [title, style, color] as parameters to generate a chart of the same type. |

**Table 5: Response format and evaluation metrics of tasks included in NovaChart.**

| Task | Response Type(s) | Evaluation Metric |
|---|---|---|
| Data Identification | string | EM |
| Data Comparison | string | EM |
| Data Extraction with Condition | sequence/mapping | ALD/AJD Similarity |
| Data Referring | numeric | Smoothing Relative Similarity |
| Color Recognition | string/mapping | AJD Similarity |
| Style Detection | string | EM |
| Chart Classification | string | EM |
| Visual Elements Retrieval | string | EM |
| Text Extraction | string | EM |
| Chart Pattern Recognition | string/open-ended text | EM (only string responses without CoT are evaluated) |
| Chart Analysis | open-ended text | GPT-Score |
| Chart Summarization | open-ended text | GPT-Score |
| Table to Chart Generation | open-ended code | GPT-Score |
| Chart Type Conversion | open-ended code | GPT-Score |
| Chart Blueprint | open-ended code | GPT-Score |

**Table 6: Supplementary experimental results in chart data understanding tasks.**

| Model | Data Identification (EM) | Data Comparison (EM) | Data Extraction with Condition (ALD/AJD Similarity) | Data Referring (Smoothing-Relative) |
|---|---|---|---|---|
| Qwen-VL-chat | 0.344 | 0.374 | 0.125 | 0.435 |
| Qwen-VL-chat + NovaChart | **0.746** | **0.673** | **0.491** | **0.550** |
| LLaVA-v1.5 | 0.246 | 0.262 | 0.234 | 0.260 |
| LLaVA-v1.5 + NovaChart | **0.698** | **0.704** | **0.524** | **0.625** |
| InternLM-XComposer | 0.380 | 0.568 | 0.358 | 0.539 |
| InternLM-XComposer + NovaChart | **0.896** | **0.855** | **0.623** | **0.837** |

**Table 7: Supplementary experimental results in chart visual understanding tasks.**

| Model | Color Recog. (AJD Similarity) | Style Detection (EM) | Chart Classifi. (EM) | V. Elements Identifi. (EM) | Text Extraction (EM) |
|---|---|---|---|---|---|
| Qwen-VL-chat | 0.159 | 0.109 | 0.423 | 0.188 | 0.690 |
| Qwen-VL-chat + NovaChart | **0.400** | **0.418** | **0.649** | **0.523** | **0.720** |
| LLaVA-v1.5 | 0.120 | 0.078 | 0.338 | 0.178 | 0.369 |
| LLaVA-v1.5 + NovaChart | **0.587** | **0.907** | **0.979** | **0.704** | **0.869** |
| InternLM-XComposer | 0.261 | 0.163 | 0.442 | 0.244 | 0.537 |
| InternLM-XComposer + NovaChart | **0.826** | **0.993** | **1.000** | **0.885** | **0.963** |

**Table 8: Supplementary experimental results in chart analysis and summarization tasks.**

| Model | Chart Pattern Recognition (EM) | Chart Analysis (GPT-Score) | Chart Summarization (GPT-Score) |
|---|---|---|---|
| Qwen-VL-chat | 0.123 | 0.402 | 0.362 |
| Qwen-VL-chat + NovaChart | **0.392** | **0.525** | **0.486** |
| LLaVA-v1.5 | 0.105 | 0.428 | 0.337 |
| LLaVA-v1.5 + NovaChart | **0.647** | **0.445** | **0.347** |
| InternLM-XComposer | 0.154 | 0.384 | 0.264 |
| InternLM-XComposer + NovaChart | **0.661** | **0.660** | **0.764** |

## Multi-modal Chart Understanding Evaluation

**Evaluation Description:**
You are asked with evaluating the outputs generated by two models (Model A and Model B) on a multi-modal chart image understanding task. You are required to provide evaluations based on the correctness and meaningfulness of the textual outputs generated by these models. You will be provided with the following information for each model:
• Type of chart
• Logical representation of the chart
• The task given to both models
• Textual output generated by each model
• Evaluation criteria
• Your response format (as a json object)
Your evaluation should focus on assessing the accuracy of the information presented in the outputs (correctness) as well as the depth and relevance of the insights provided (meaningfulness). Please assign scores from 0 to 5 for each aspect of the evaluation, with higher scores indicating better performance.

**Chart Type:** $chart_type

**Logical Representation of Data:**
• Description of the chart and logical representation: $description
• Context information and semantic mapping of the chart: $semantic
• Logical representation: $data_representation

**Task Given to Models:** $task

**Model Outputs:**
• **Model A Output:** $model_a_output
• **Model B Output:** $model_b_output

**Evaluation Criteria:**
**Correctness:**
– 0: Output contains significant factual errors or misinterpretations of data.
– 1: Output contains several factual errors or misinterpretations of data.
– 2: Output contains some factual errors or misinterpretations of data.
– 3: Output is mostly accurate but may contain minor errors or ambiguities.
– 4: Output is largely accurate with few errors or ambiguities.
– 5: Output is completely accurate with no factual errors or ambiguities.
**Meaningfulness:**
– 0: Output lacks meaningful insights or fails to provide relevant analysis.
– 1: Output provides very limited or superficial insights.
– 2: Output provides some insights but lacks depth or relevance.
– 3: Output provides relevant insights with reasonable depth.
– 4: Output provides insightful analysis with good depth and relevance.
– 5: Output provides highly insightful analysis with deep relevance and clarity.

**Response Format of the Evaluation**
The output should be formatted as a JSON instance that conforms to the JSON schema below.
As an example, for the schema {"properties": {"foo": {"title": "Foo", "description": "a list of strings", "type": "array", "items": {"type": "string"}}}, "required": ["foo"]}the object {"foo": ["bar", "baz"]} is a well-formatted instance of the schema. The object {"properties": {"foo": ["bar", "baz"]}} is not well-formatted.
Here is the output schema:

```
{"properties": {"Model_A_Evaluation": {"description": "A brief comment on model A's output. ",
"title": "Model A Evaluation", "type": "string"}, "Model_A_Correctness": {"description": "The
correctness score (0~5) of model A's output. ", "title": "Model A Correctness", "type":
"string"}, "Model_A_Meaningfulness": {"description": "The meaningfulness score (0~5) of model
A's output. ", "title": "Model A Meaningfulness", "type": "string"}, "Model_B_Evaluation":
{"description": "A brief comment on model B's output. ", "title": "Model B Evaluation", "type":
"string"}, "Model_B_Correctness": {"description": "The correctness score (0~5) of model B's
output. ", "title": "Model B Correctness", "type": "string"}, "Model_B_Meaningfulness":
{"description": "The meaningfulness score (0~5) of model B's output. ", "title": "Model B
Meaningfulness", "type": "string"}}, "required": ["Model_A_Evaluation", "Model_A_Correctness",
"Model_A_Meaningfulness", "Model_B_Evaluation", "Model_B_Correctness",
"Model_B_Meaningfulness"]}
```

**Figure 2: The evaluation prompt template for chart analysis and summarization tasks.**

## Multi-modal Chart Generation Evaluation

**Evaluation Description:**
You are asked with evaluating the outputs generated by two models (Model A and Model B) on a multi-modal chart understanding and coding task. You are required to provide evaluations based on the correctness and quality of the textual outputs (containing the code) generated by these models. You will be provided with the following information:
• Type of chart
• Logical representation of the chart
• The task given to both models
• Textual output generated by each model
• Evaluation criteria
• Your response format (as a json object)
Your evaluation should focus on assessing the accuracy of the code presented in the outputs (correctness) as well as the quality of the code generated (quality). Please assign scores from 0 to 5 for each aspect of the evaluation, with higher scores indicating better performance.

**Chart Type:** $chart_type

**Logical Representation of Data:**
• Description of the chart and logical representation: $description
• Context information and semantic mapping of the chart: $semantic
• Logical representation: $data_representation

**Task Given to Models:** $task

**Model Outputs:**
• **Model A Output:** $model_a_output
• **Model B Output:** $model_b_output

**Evaluation Criteria:**
**Correctness:**
– 0: Code does not meet any of the requirements and produces completely irrelevant or endless repetition results.
– 1: Code contains major errors or misinterpretations or attempt to evade most of task requirements.
– 2: Code meets some requirements of the task but contains notable errors.
– 3: Code mostly meets the requirements of the task with minor errors or a general rather than detailed code.
– 4: Code largely meets the requirements of the task with few errors or ambiguities.
– 5: Code perfectly meets all requirements of the task.
**Quality**:
– 0: Code lacks any discernible structure or organization and is unreadable.
– 1: Code has minimal structure and inconsistent formatting, making it difficult to follow.
– 2: Code has basic structure but lacks consistency in formatting and naming conventions.
– 3: Code has a clear structure and follows consistent formatting and naming conventions, but improvements can be made.
– 4: Code has a well-defined structure, consistent formatting, and clear variable naming, contributing to good readability.
– 5: Code has an excellent structure, impeccable formatting, and meaningful variable naming, making it highly readable and maintainable.

**Response Format of the Evaluation**
The output should be formatted as a JSON instance that conforms to the JSON schema below.
As an example, for the schema {"properties": {"foo": {"title": "Foo", "description": "a list of strings", "type": "array", "items": {"type": "string"}}}, "required": ["foo"]}the object {"foo": ["bar", "baz"]} is a well-formatted instance of the schema. The object {"properties": {"foo": ["bar", "baz"]}} is not well-formatted.
Here is the output schema:

```
{"properties": {"Model_A_Evaluation": {"description": "A brief comment on model A's output. ",
"title": "Model A Evaluation", "type": "string"}, "Model_A_Correctness": {"description": "The
correctness score (0~5) of model A's output. ", "title": "Model A Correctness", "type":
"string"}, "Model_A_Quality": {"description": "The quality score (0~5) of model A's output. ",
"title": "Model A Quality", "type": "string"}, "Model_B_Evaluation": {"description": "A brief
comment on model B's output. ", "title": "Model B Evaluation", "type": "string"},
"Model_B_Correctness": {"description": "The correctness score (0~5) of model B's output. ",
"title": "Model B Correctness", "type": "string"}, "Model_B_Quality": {"description": "The
quality score (0~5) of model B's output. ", "title": "Model B Quality", "type": "string"}},
"required": ["Model_A_Evaluation", "Model_A_Correctness", "Model_A_Quality",
"Model_B_Evaluation", "Model_B_Correctness", "Model_B_Quality"]}
```

**Figure 3: The evaluation prompt template for chart generation tasks.**

**Table 9: Supplementary experimental results in chart generation tasks.**

| Model | Table to Chart Genetation (GPT-Score) | Chart Conversion (GPT-Score) | Chart Blueprint (GPT-Score) |
|---|---|---|---|
| Qwen-VL-chat | 0.115 | 0.170 | 0.206 |
| Qwen-VL-chat + NovaChart | **0.434** | **0.347** | **0.263** |
| LLaVA-v1.5 | 0.252 | 0.358 | 0.360 |
| LLaVA-v1.5 + NovaChart | **0.282** | **0.497** | **0.440** |
| InternLM-XComposer | 0.268 | 0.376 | 0.264 |
| InternLM-XComposer + NovaChart | **0.770** | **0.738** | **0.833** |

**Table 10: Supplementary experimental results on different chart types (averaged over tasks)**

| Model | Bi. Hist. | Box | Heatmap | K.G. | Multi. Bar | Multi. Line | Multi. Scatter | Pie | Radar |
|---|---|---|---|---|---|---|---|---|---|
| Qwen-VL-chat | 0.274 | 0.337 | 0.353 | 0.212 | 0.307 | 0.337 | 0.294 | 0.387 | 0.317 |
| Qwen-VL-chat + NovaChart | **0.581** | **0.513** | **0.626** | **0.438** | **0.532** | **0.505** | **0.450** | **0.624** | **0.586** |
| LLaVA-v1.5 | 0.212 | 0.291 | 0.259 | 0.264 | 0.227 | 0.250 | 0.208 | 0.333 | 0.291 |
| LLaVA-v1.5 + NovaChart | **0.753** | **0.630** | **0.670** | **0.419** | **0.662** | **0.657** | **0.554** | **0.788** | **0.621** |
| InternLM-XComposer | 0.255 | 0.305 | 0.228 | 0.274 | 0.358 | 0.411 | 0.299 | 0.460 | 0.351 |
| InternLM-XComposer + NovaChart | **0.906** | **0.825** | **0.855** | **0.723** | **0.897** | **0.864** | **0.701** | **0.878** | **0.864** |

| Model | Ring | Rose | Sankey | Bar | Line | Scatter | Table | Uni. Hist. | Word. |
|---|---|---|---|---|---|---|---|---|---|
| Qwen-VL-chat | 0.343 | 0.310 | 0.186 | 0.364 | 0.302 | 0.302 | 0.115 | 0.288 | 0.418 |
| Qwen-VL-chat + NovaChart | **0.624** | **0.603** | **0.361** | **0.602** | **0.445** | **0.427** | **0.434** | **0.605** | **0.429** |
| LLaVA-v1.5 | 0.294 | 0.272 | 0.242 | 0.229 | 0.296 | 0.257 | 0.252 | 0.264 | **0.504** |
| LLaVA-v1.5 + NovaChart | **0.765** | **0.757** | **0.491** | **0.699** | **0.574** | **0.551** | **0.282** | **0.749** | 0.301 |
| InternLM-XComposer | 0.463 | 0.436 | 0.116 | 0.460 | 0.320 | 0.369 | 0.268 | 0.368 | 0.413 |
| InternLM-XComposer + NovaChart | **0.870** | **0.913** | **0.730** | **0.882** | **0.726** | **0.689** | **0.770** | **0.901** | **0.815** |

## References

[1] Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, and Hanwang Zhang. 2023. ChartLlama: A Multimodal LLM for Chart Understanding and Generation. arXiv:2311.16483 [cs.CV]

[2] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. https://openreview.net/forum?id=nZeVKeeFYf9

[3] Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. Soviet Union, 707–710.

[4] Fangyu Liu, Julian Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. 2023. DePlot: One-shot visual language reasoning by plot-to-table translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 10381–10399. https://doi.org/10.18653/v1/2023.findings-acl.660

[5] Renqiu Xia, Bo Zhang, Haoyang Peng, Hancheng Ye, Xiangchao Yan, Peng Ye, Botian Shi, Yu Qiao, and Junchi Yan. 2024. StructChart: Perception, Structuring, Reasoning for Visual Chart Understanding. arXiv:2309.11268 [cs.CV]