

# SO-Bench: A Structural Output Evaluation of Multimodal LLM

## Supplementary Material

### A. Dataset

The original Vistext [43] eval dataset contains only single chart images that only have single data series, make the task less challenging. In our work, we first cluster charts into groups of size 2-6 using original chart titles bag-of-words as features and TF-IDF method to find similar charts. Then we render the chart group into one image treating each chart as a sub-plot. In total 383 charts are converted into 99 images, which are used in our data generation pipeline.

Figure 9 shows the top 100 most frequent features in the input JSON schemas of the SO-BENCH dataset, demonstrating the diversity of schema fields covered by our benchmark.

### B. Training Details

#### B.1. Training Dataset

This section provides additional details on the training dataset used in our study (Section 4.3). For supervised fine-tuning, we construct a training corpus sourced from the training sets of HierText [26], AriaUI [57], and COYO [7]. We balance both in-domain vs. out-of-domain data and real vs. synthetic JSON schemas. HierText serves as the in-domain dataset because its test split is used to build part of the SO-BENCH evaluation set. COYO and AriaUI, in contrast, provide out-of-domain training data. For the full HierText training set (10K images), we generate one version with synthetic schemas and another with real JSON schemas. For AriaUI, we downsample to 10K images, drawn from desktop, mobile, and web UI screens. Similar to HierText split, we generate one example with synthetic schema and another with real JSON schema for every image. Since COYO is extremely large (over 747M image-text pairs), we first categorize and then randomly downsample it to 95K images, consisting of: 20K infographics, 10K handwritten images, 40K scene-text images, 5K maps, and 20K tables. Among these, 80% of the images are paired with real JSON schemas, while 20% use synthetic schemas. Finally, we filter out noisy samples with low label quality, ending up with 114K training samples.

#### B.2. Training hyperparameters

In all our of SFT experiments, we train the model for around 3 epochs and we adjust the batch size (among 64, 128, 256) and training steps accordingly based on the training data size. We used learning rate of  $2e-5$  and max sequence length of 16k. For RL experiments, we used learning rate of  $3e-7$ , batch size of 256. For each sample, we generate 32 rollouts

and compute the advantage using the REINFORCE Leave One Out (RLOO) [3]. Again we train the model for around 3 epochs.

### C. Ablation Studies

Figure 11 studies the metrics correlations between SO-BENCH and other benchmarks from multiple open-sourced and proprietary models presented in Table 1. Their Pearson correlation coefficients are summarized in Table 2.

### D. Error Examples

Figure 12 shows an example output from Gemini-2.5-pro. We see that the model is already very good at interpreting chart data and its predicted data values are very close to the ground truth, which are considered to be correct with fuzzy match. However, there is no explicit unit annotation on the chart, so the model has to infer the unit based on the text. Although the predicted answer also make sense, the edit distance w.r.t the ground truth already exceeds the threshold, thus it could not score full match on this example. One potential to improve our evaluation metrics is to introduce VLM-as-a-judge for fields with fuzzy match.

As we discussed in subsection 4.3, due to overall shallower schema depth in AriaUI subset of training data, the model trained on it could not learn to generalize well to deep schemas in chart domain. Figure 13 shows such an error example from the model. In this case, the extracted values are correct, but the model just hallucinated a shallow structured that is not compliant with the input schema, leading to 0 scores on all metrics.

In Figure 14, Figure 15, Figure 16, we show 3 more error examples made by the SFT model trained on the entire training set. In Figure 14, we see that the model is partially correct on the generated outputs, but it mistakenly treated the title of the schema as one of the target fields, and produced an extra field which leads to 0 score on schema validation. This suggests that the model’s instruction following ability could be further improved. In Figure 15, we see that the model’s output is fully compliant with the schema, but all values are off compared to the ground truth. We see that the text in this image is very dense, and even for humans we have to zoom in on the first section of the page to see the details. This suggest that the visual perception ability of high-resolution images is still relatively weak. Finally, in Figure 16, we see that the model is correct in terms of schema validation and also get partial score on field match. However, it could not capture all available discounts on the page, because the information is actually scatter at differ-

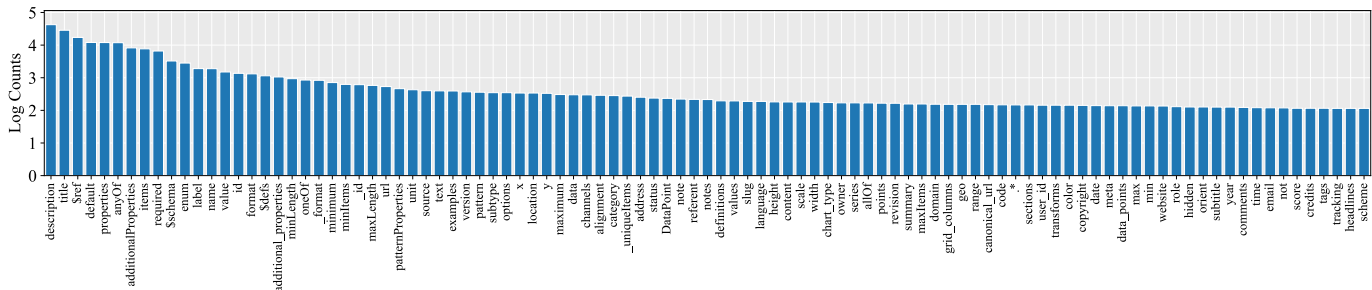


Figure 9. Top 100 frequent features

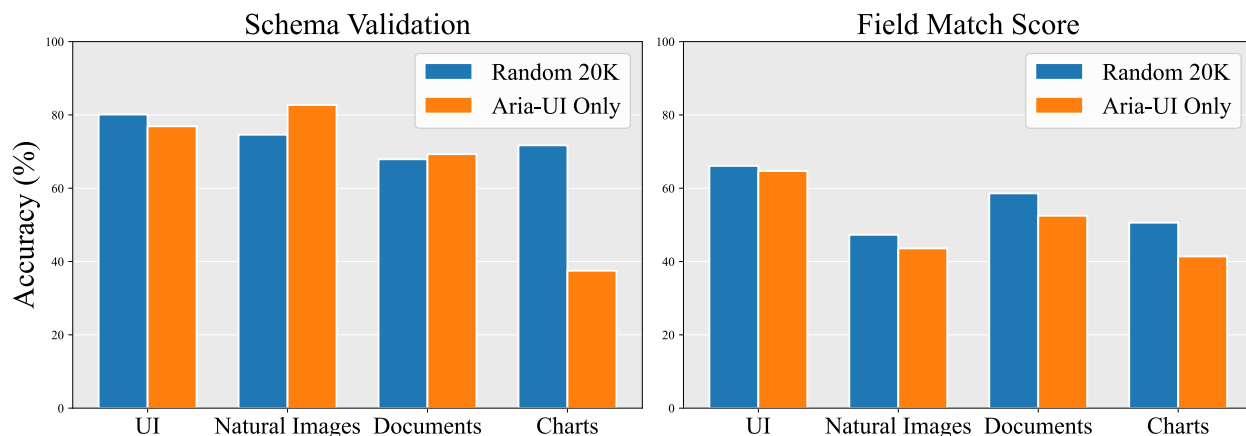


Figure 10. The results breakdown by the data categories for the model trained with 20K randomly sampled data and model trained with Aria-UI subset of the data. In the left figure, we show the schema validation accuracy and in the right figure, we show the field match accuracy (fuzzy).

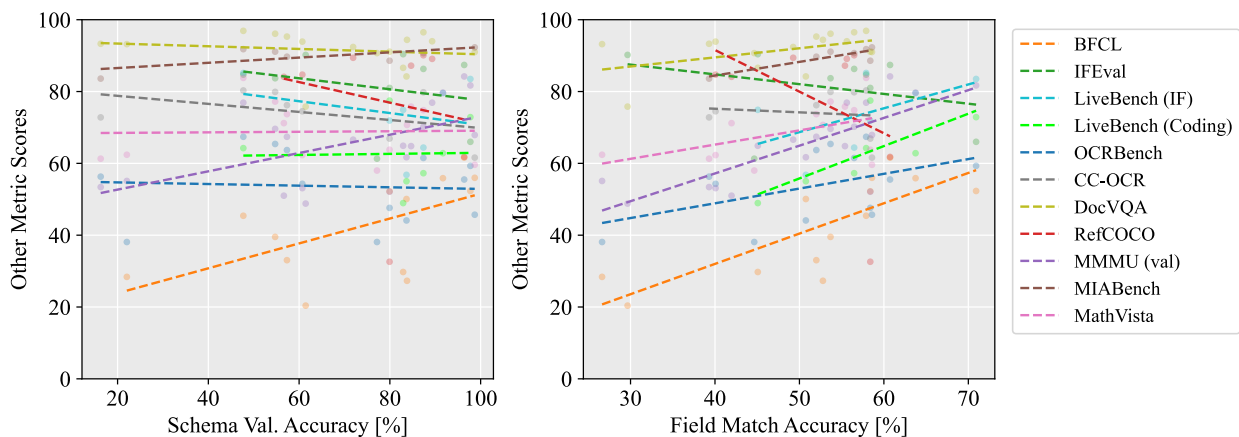


Figure 11. Metrics correlation analysis.

ent places. This hints that the general visual understanding ability that require reasoning over entire image is also a direction for improvement.

### E. Instruction Following Prompt

Table E presents the instruction-following prompt used for structured output generation. Notably, the prompt in-



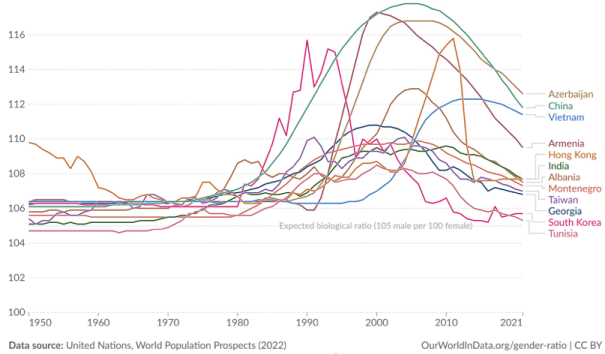
What were the 2010 sex ratios for China, India, and South Korea?

JSON Schema

```
{
  "$defs": {
    "Series": {
      "properties": {
        "name": {
          "description": "Series label",
          "title": "Name",
          "type": "string"
        },
        "values": {
          "description": "Values aligned to time_points",
          "items": {
            "anyOf": [
              {"type": "number"},
              {"type": "null"}
            ]
          },
          "title": "Values",
          "type": "array"
        },
        "unit": {
          "anyOf": [
            {"type": "string"},
            {"type": "null"}
          ],
          "default": "None",
          "description": "Unit for this series",
          "title": "Unit"
        }
      },
      "required": [
        "name",
        "values"
      ],
      "title": "Series",
      "type": "object"
    },
    ...
  },
  "title": "TimeSeriesChart",
  "type": "object"
}
```

Sex ratio at birth, 1950 to 2021

The sex ratio at birth is measured as the number of newborn boys for every 100 newborn girls. Higher values indicate a much higher number of newborn boys than girls.



```
{
  "series": [
    {
      "name": "China",
      "values": [117.5]
    },
    {
      "name": "India",
      "values": [111]
    },
    {
      "name": "South Korea",
      "values": [107]
    }
  ],
  "y_axis": {
    "label": "Sex ratio at birth",
    "unit": "newborn boys for every 100 newborn girls",
    "scale": "linear"
  },
  ...
}
```



```
{
  "series": [
    {
      "name": "China",
      "values": [118]
    },
    {
      "name": "India",
      "values": [109.5]
    },
    {
      "name": "South Korea",
      "values": [106.5]
    }
  ],
  "y_axis": {
    "label": "Sex ratio at birth",
    "unit": "boys per 100 girls",
    "scale": "linear"
  },
  ...
}
```

Figure 12. An example error (Chart image) from Gemini-2.5-Pro predictions. The box with Robot icon contains model predictions and the box with star icon contains ground truth. The blue font indicate that our fuzzy match metric is able to count those fields as correct, and the red font indicate mismatched field.

cludes explicit guidance on handling default values, particularly for fields labeled as required but not directly parsable from the input images. The same rules are consistently applied throughout both the data generation pipeline and the inference process, ensuring alignment between training and evaluation settings.

Get the "very important" percentages for being physically active, being outdoors, and being successful in your career.

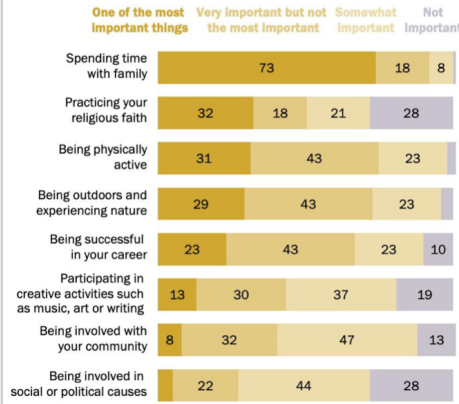
```

JSON Schema
{
  "ResponseOption": {
    "properties": {
      "key": {
        "description": "Stable key for programmatic reference.",
        "title": "Key",
        "type": "string"
      },
      "label": {
        "description": "Human-readable response label.",
        "title": "Label",
        "type": "string"
      },
      "order": {
        "anyOf": [
          { "type": "integer" },
          { "type": "null" }
        ],
        "title": "Order"
      },
      "required": ["key", "label"],
      "title": "Response Option",
      "type": "object"
    },
    "ResponseShare": {
      "properties": {
        "option_key": {
          "description": "References ResponseOption.key.",
          "title": "Option Key",
          "type": "string"
        },
        "percent": {
          "description": "Percentage value (0-100).",
          "title": "Percent",
          "type": "number"
        }
      },
      "required": ["option_key", "percent"],
      "title": "Response Share",
      "type": "object"
    }
  }
}

```

**Family time is of top importance for most Americans; majorities also prioritize physical activity, being outdoors and career success**

% who say each of the following is \_\_\_ to them personally, regardless of how much time they spend doing it



Source: Survey of U.S. adults conducted March 27-April 2, 2023. PEW RESEARCH CENTER

```

{
  "unit": "percent",
  "items": [
    {
      "label": "Being physically active",
      "shares": [
        {
          "option_key": "very_important",
          "percent": 43
        }
      ]
    },
    {
      "label": "Being outdoors and experiencing nature",
      "shares": [
        {
          "option_key": "very_important",
          "percent": 43
        }
      ]
    },
    {
      "label": "Being successful in your career",
      "shares": [
        {
          "option_key": "very_important",
          "percent": 43
        }
      ]
    }
  ]
}

```

```

{
  "very_important": {
    "being_physically_active": 43,
    "being_outdoors": 43,
    "being_successful_in_your_career": 43
  }
}

```

Figure 13. An example error (Chart image) from the model trained on AriaUI subset of data. Although the extracted values themselves are correct, the output does not follow the schema structure at all, leading to 0 scores on all metrics in this case.

Save this historic site as a place so I can look it up later for my notes.

```

JSON Schema
{
  "properties": {
    "address": {
      "$ref": "#/definitions/PostalAddress",
      "description": {
        "type": "string"
      },
      "geo": {
        "anyOf": [
          {
            "elevation": {
              "type": "number"
            },
            "latitude": {
              "type": "number"
            },
            "longitude": {
              "type": "number"
            }
          }
        ],
        "type": "object"
      },
      "image": {
        "format": "uri",
        "type": "string"
      },
      "name": {
        "type": "string"
      },
      "url": {
        "format": "uri",
        "type": "string"
      }
    },
    "required": [
      "name"
    ],
    "title": "Place",
    "type": "object"
  }
}

```



```

{
  "address": {
    "addressRegion": "Wisconsin",
    "description": "The Janesville Plank Road was one of 16 privately-owned toll roads authorized by Wisconsin's Territorial legislature in 1848. The road was specified to follow a 65 mile route from Milwaukee through East Troy to Janesville. Its roadbed was built on the 1838 Janesville Territorial Road, which followed an Indian trail extending southwestward from rice marshes near river mouths in the Milwaukee area to Janesville. (Continued on other side) Erected 2000.",
    "name": "Janesville Plank Road"
  }
}

```

```

{
  "address": {
    "addressLocality": "Janesville",
    "addressRegion": "Wisconsin",
    "description": "The Janesville Plank Road was one of 16 privately-owned toll roads authorized by Wisconsin's Territorial legislature in 1848. The road was specified to follow a 65 mile route from Milwaukee through East Troy to Janesville. Its roadbed was built on the 1838 Janesville Territorial Road, which followed an Indian trail extending southwestward from rice marshes near river mouths in the Milwaukee area to Janesville.",
    "name": "JANESVILLE PLANK ROAD",
    "title": "Place"
  }
}

```

Figure 14. An example error (Natural image) from the SFT model trained on full data. This example gets partial score on field match but is invalid w.r.t. the schema



I'm pricing software for our nonprofit—pull the pricing for the Indie tier along with any discounts I can apply for the plan on the left.

```

JSON Schema
{
  "properties": {
    "discounts": {
      "items": {
        "properties": {
          "code": {
            "description": "The discount code",
            "type": "string"
          },
          "percentage": {
            "description": "The discount percentage",
            "type": "number"
          }
        },
        "required": [
          "percentage",
          "code"
        ],
        "type": "object"
      },
      "type": "array"
    },
    "original_price": {
      "description": "The original price of the product",
      "type": "number"
    },
    "required": [
      "original_price",
      "discounts"
    ],
    "type": "object"
  }
}

```

```

{
  "discounts": [
    {
      "code": "5-9",
      "percentage": 10
    },
    {
      "code": "10+",
      "percentage": 20
    },
    {
      "code": "renewed
within 90
days",
      "percentage": 30
    }
  ],
  "original_price": 299
}

```

```

{
  "discounts": [
    {
      "code": "#",
      "percentage": 10
    }
  ],
  "original_price": 299
}

```

ServiceStack

What's New | Blog | Docs | Forums | Free | Total | Pricing | Sign In

Get started

## All ServiceStack

Unlimited usage of all ServiceStack Software, includes 12 months Software Maintenance & Commercial Support

Perpetual, Royalty-Free, Per-developer Pricing

**Indie**

For individuals, non-profits or organizations with 10 employees or less

**\$299** / developer

- Unlimited usage
- 12 months Software Maintenance
- Support Forums
- Issue Tracker
- Named License

ORDER NOW

**Business**

For large organizations with more than 10 total employees

**\$999** / developer

- Unlimited usage
- 12 months Software Maintenance
- Support Forums
- Issue Tracker
- Floating License
- 4 incidents / year

ORDER NOW

**Free**

ServiceStack is free for individuals and collaborative OSS projects or small apps that fit within the Free Usage Quotas. A 30 Day Trial License can be used to evaluate any ServiceStack Software unrestricted for 30 days.

Multiple License Discounts below are available for all purchases. 30% renewal discount is available for developer licenses renewed within 90 days. Renewing extends Software Updates and Commercial Support for an additional 12 months.

Qty	Discount
5-9	10%
10+	20%

Indie Renewal

**\$199** / developer

Business Renewal

**\$699** / developer

Figure 16. An example error (UI image) from the SFT model trained on full data. This example gets partial score on field match while also being valid w.r.t the schema

## Instruction Following Prompt for Structured Output

### SYSTEM PROMPT

**Visual Structured Information Extractor** You are an expert at extracting structured information from images according to provided schemas. Given the image and user intent, extract information that matches the provided JSON schema.

### Constraint-Aware Default Rules (for missing/unextractable required fields)

#### General precedence

- **P0 | Use explicit schema defaults:** If provided in schema, prefer these.
- **P1 | Respect enumerations/const:** Use the declared const or the first enum value (in stable order).
- **P2 | Apply type-specific rules while respecting schema constraints.**

#### Strings

- Default placeholder: #.
- Respect minLength/maxLength:
  - If minLength = m  $\geq$  1, return exactly m hash characters.
  - If minLength = maxLength = n, return n hash characters.
- If pattern forbids #, construct the shortest valid alternative using allowed characters ("A", "0", etc.).
- Only output null if "null" is explicitly allowed.

#### Integers & Numbers

- Default to 0.
- If minimum/exclusiveMinimum is present, choose the lowest valid value.
- If maximum/exclusiveMaximum is present and 0 violates it, choose the highest valid value under the bound.
- If multipleOf is present, adjust to the nearest valid multiple.

#### Booleans

- Default to false unless otherwise constrained.

#### Dates / Times

- Use RFC 3339-valid placeholders:
  - "1970-01-01" for date
  - "1970-01-01T00:00:00Z" for date-time
  - "00:00:00Z" for time
- If stricter patterns exist, choose the simplest valid match.

#### Arrays

- If minItems = k, return exactly k items, recursively filled with these same rules.
- Ensure uniqueness if uniqueItems:true.

#### Objects

- Populate all required fields recursively.
- Never add undeclared fields.

#### Edge-Case Notes

- Never hallucinate values. Required fields must be filled with constraint-compliant placeholders.
- Optional fields may be omitted or set to null if allowed.
- If constraints are contradictory, return the closest minimally violating placeholder.
- Stable tie-breaking: when multiple placeholder choices are possible, use the first in schema order.

#### Output Format

- Return only valid JSON, wrapped in:

```
```json
{ ... }
```
```

- All required fields must appear.
- Placeholders must respect schema constraints.

- No comments, no extra text, no extra fields.

**USER PROMPT**

JSON Schema:

```
```json  
{schema_json}  
```
```

User Intent: {user\_intent}

Please analyze the image and extract structured information according to the schema above, honoring the user intent.

Return only valid JSON in a fenced ```json code block.