# ViTaB-A: Evaluating Multimodal Large Language Models on Visual Table Attribution

Anonymous ACL submission

## Abstract

Multimodal Large Language Models (mLLMs) are often used to answer questions in structured data such as tables in Markdown, JSON, and images. While these models can often give correct answers, users also need to know where those answers come from. In this work, we study structured data attribution/citation, which is the ability of the models to point to the specific rows and columns that support an answer. We evaluate several mLLMs across different table formats and prompting strategies. Our results show a clear gap between question answering and evidence attribution. Although question answering accuracy remains moderate, attribution accuracy is much lower, near random for JSON inputs, across all models. We also find that models are more reliable at citing rows than columns, and struggle more with textual formats than images. Finally, we observe notable differences across model families. To better understand scaling within these model families, we introduce a Family Scaling Score (FSS), which measures how attribution performance changes with model size. Overall, our findings show that current mLLMs are unreliable at providing fine-grained, trustworthy attribution for structured data, which limits their usage in applications requiring transparency and traceability. The link for anonymous code repository.

## 1 Introduction

Multimodal Large Language Models (mLLMs) are increasingly used to answer questions over structured data. In practice, users rely on these models to read tables, extract values, compare entries, and summarize records across formats such as Markdown tables, JSON files, and document images. Prior work shows that mLLMs can often answer questions about structured inputs with reasonable accuracy, making them attractive as general-purpose data assistants (Fang et al., 2024; Liu et al., 2024).

However, answering a question correctly is often not enough. In many real-world scenarios, users also want to know where an answer came from. For example, if a model reports that a company's revenue increased in a given year, a natural follow-up is which row and which column in the table support this claim. In current systems, this step is frequently unreliable: models may produce a correct answer while failing to identify the specific part of the table that justifies it. We demonstrate this gap empirically, showing that question answering accuracy remains relatively high while attribution or citation accuracy is substantially lower across models and prompting strategies (Section 4.1).

In this paper, we study structured data attribution-the ability of mLLMs not only to generate correct answers, but to localize the rows and columns in the input data that support those answers. We evaluate attribution across three common table representations-Markdown, JSON, and images-using multiple model families and prompting strategies.

Our study is motivated by the observation that answer accuracy and attribution accuracy are distinct capabilities. Prior work shows that models can often arrive at correct answers without being fully grounded in the underlying evidence, particularly when partial cues or broadly relevant context are sufficient (Bohnet et al., 2022; Radevski et al., 2025). Our results provide concrete evidence of this disparity in structured data settings: across all evaluated models, question answering accuracy remains around 48–55%, while attribution accuracy is dramatically lower-often below 30%, and near

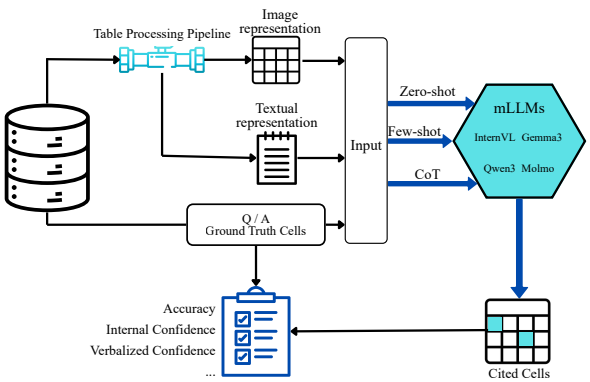


Figure 1: Overview of the ViTaB-A benchmarking.

random for JSON inputs (Section 4.1).

These findings align with broader evidence that attribution and citation remain challenging for language models. Prior work on hallucination shows that models often generate confident but ungrounded outputs, including incorrect or fabricated references (Huang et al., 2025). Even when explicitly prompted to cite sources, models frequently produce vague or incorrect attributions (Gao et al., 2023). Fine-grained structured attribution benchmarks-such as TabCite-have been introduced to assess models' ability to locate relevant table structures (e.g., rows and columns), highlighting that precise localization remains an open challenge (Mathur et al., 2024).

Across our experiments, we observe several consistent patterns. First, except in JSON settings, models are substantially better at identifying the correct row than the correct column, suggesting persistent difficulty with fine-grained field-level localization. Second, attribution is more reliable when tables are presented as images than when they are provided in textual formats such as Markdown or JSON. Finally, we observe notable differences across model families, indicating that architectural choices influence attribution behavior. To further analyze how attribution changes with model scale, we introduce the Family Scaling Score (FSS), a metric that summarizes how attribution performance improves as model size increases within a model family. FSS captures both the average scaling trend and the stability of this improvement across prompting strategy.

These limitations are particularly concerning in domains such as finance, healthcare, and law, where systems must support auditability and traceability. In such settings, it is not sufficient to provide a plausible answer; outputs must be traceable to specific data fields. Our results show that current mLLMs are unreliable at providing this level of fine-grained traceability, even when their answers appear correct (Section 4.1).

Finally, we summarize our contributions as follows:

- We propose **ViTaB-A**, an exhaustive benchmark for assessing mLLMs on Visual Table Attribution tasks across modalities (text, JSON, rendered images).
- To the best of our knowledge, we are the first to benchmark open-source mLLM families, not only on Table QA and Attribution performances, but also under confidence alignment and uncertainty calibration.
- We introduce the **Family Scaling Score (FSS)**, a family-level metric that quantifies how reliably attribution performance improves with model scale by jointly capturing average scaling gains, worst-case prompt behavior, and cross-prompt stability.
- Our findings reveal that mLLMs often struggle in spatial QA tasks compared to spatial attribution in a text-in-vision paradigm.

## 2 Related Works

**Question Answering over Structured Data:** Early neural methods such as TaPas (Herzig et al., 2020) and structure-aware transformers (Zhang et al., 2020) showed table reasoning without hand-crafted query programs. Datasets like HiTab (Cheng et al., 2022) and HiBench (Jiang et al., 2025) introduce hierarchical challenges, and several studies report moderate QA performance across text and image table formats (Fang et al., 2024; Deng et al., 2024; Zheng et al., 2024).

**Attribution and Grounding in LLMs:** Models often produce fluent but unsupported claims (Huang et al., 2025), and prompted citation generation still yields incorrect or unverifi-

able references (Gao et al., 2023). Evidence selection remains hard in practice (Bohnet et al., 2022; Vankov et al., 2025; Radevski et al., 2025), with recent efforts targeting structured table attribution directly (Mathur et al., 2024; Hu et al., 2025).

**Optimization Objectives and Confidence:** LLMs trained to reward helpful answers (Ouyang et al., 2022) are typically evaluated on final-answer correctness (Liang et al., 2022), with systems like WebGPT adding components to improve grounding (Nakano et al., 2021). Research on calibration reveals frequent misalignment between confidence and accuracy (Kumar et al., 2024a; Geng et al., 2024; Ye et al., 2024; Groot and Valdenegro Toro, 2024).

**Representation Effects:** Representation and layout affect model behavior, with prior work reporting comparisons of text vs. image table inputs and multimodal spatial challenges (Deng et al., 2024; Zheng et al., 2024; Wang et al., 2024; Liu et al., 2025).

## 3 Methodology

As prior works have demonstrated, mLLMs can produce correct answers, even with spurious calculations or hallucinated reasoning. We conduct a comprehensive study on visual table attribution to investigate trust and reliability in mLLMs. We focus on two major research questions:

1. *How accurately do mLLMs identify table cells that support a given answer?*
2. *Does a model's confidence score reliably reflect the correctness of its attribution?*

To address these questions, we benchmark mLLM attribution accuracy and uncertainty behavior across model family, input representations, and prompting strategies. Figure 1 details the workflow of our benchmarking approach.

In this section, we outline the overall workflow of our investigation, starting with the attribution task formulation (Sec. 3.1). We then describe the benchmark (Sec. 3.2) and model setup (Sec. 3.3) for all our experiments. Finally, we report the statistical attribution accuracy metrics for initial evaluation, followed by confidence score analysis to gauge alignment of internal and verbalized certainty during response (Sec. 3.4). Figure 1 details the workflow of our benchmarking approach.

### 3.1 Task Formulation

**V2: Reliability and Confidence Alignment**

Beyond *what* cells are cited, we assess whether the model can *reliably communicate* attribution correctness. We compare (i) *internal confidence* derived from token-level likelihoods of the citation output with (ii) *verbatim (verbalized) confidence* elicited as an explicit self-report; misalignment between the two is a known failure mode (Geng et al., 2024; Kumar et al., 2024a).

**V1: Representation Gap (Visual vs. Text)**

We vary the table representation

$$r \in \mathcal{R} \triangleq \{\mathsf{image}, \mathsf{markdown}, \mathsf{json}\}, \quad X_r = \mathrm{Enc}_r(T),$$

where $X_{\mathsf{image}}$ is a rendered table image and $X_{\mathsf{markdown}}, X_{\mathsf{json}}$ are structured text encodings. This contrasts attribution under true visual parsing vs. attribution under serialization, exposing whether grounding is genuinely multimodal (Deng et al., 2024). For image tables, we apply semantics-preserving perturbations

$$X'_{\mathsf{image}} = \pi(X_{\mathsf{image}}),$$

where $\pi$ changes appearance without changing cell content or layout (e.g., header/cell color changes, font/style changes). This allows us to assess the aggregate impact of superficial stylistic variations on attribution quality of rendered image data.

We analyze the *spatial intelligence* of an mLLM-family using structured tabular data attribution. Tables provide a controlled grounding substrate: evidence is discrete and compositional, and can be referenced unambiguously via row-column coordinates. This makes attribution a simple but revealing proxy for spatial competence: a model must align language with precise table structure (cells / rows / columns) rather than merely produce plausible text (Liu et al., 2025).

Each instance consists of a natural-language question $q$, a provided (correct) answer $a$, and a table $T$. The answer is *given* to the model to shift the burden from generation to grounding: the model is asked to identify *where* the support for $a$ resides in $T$. This isolates spatial grounding ability and enables controlled comparisons across model variants. Formally, let $\mathcal{I}(T) = \{(i, j) \mid i \in [m], j \in [n]\}$ denote the set of cell indices for an $m \times n$ table. Given $(q, a, T)$, the model outputs a set of cited cells $\hat{S} = f_\theta(q, a, X_r) \subseteq \mathcal{I}(T)$ returned as row–column indices (including header cells when they are part of the evidence) (Mathur et al., 2024). Here $X_r$ represents the table encoding.

Finally, to generalize across model scale and inference techniques, we assess the aforementioned properties on multiple model families with varying training parameters under standard in-context learning paradigms: zero-shot (Brown et al., 2020), few-shot (Brown et al., 2020), and CoT (Wei et al., 2022) prompting. All prompts used in this paper can be found in the Supplementary Material.

### 3.2 ViTaB-A Benchmark

Our experiments are conducted on **ViTaB-A**, which is constructed using the HiTab (Cheng et al., 2022) dataset, which contains question-answer pairs grounded in structured tables with annotated evidence cells. HiTab provides which ground truth reference table entries are required to support a correct answer, making it well-suited for attribution-centric evaluation.

We standardize attribution across representations by augmenting each table with explicit row and column labels. This enables unambiguous cell references (e.g., B3, E7) across all experimental conditions for accurate evaluation.

We present the tables to models using three different representations: **(1) JSON; (2) Markdown; (3) Rendered images**. Such setup allows us to study the attribution behavior under both structured textual and visual inputs. For image-based tables, we additionally introduce controlled visual perturbations that preserve the underlying tabular content while altering table appearance. These perturbations include variations in header color (red, blue, and green) and font style (Arial and Times New Roman). For each representation, we select 200 tables as the visual table attribution benchmark.

### 3.3 Model Setup

We evaluate a diverse set of mLLMs with varying architectures and parameter scales to investigate the capabilities of visual table attribution across model families. Specifically, we consider the **Gemma-3** family (4B, 12B, and 27B), **InternVL3.5** models (4B, 8B, 14B, and 38B), **Qwen3-VL** vision-language models (2B, 4B, 8B, and 32B), and the **Molmo2** family (4B and 8B) to assess evolution of attribution accuracy and uncertainty across model scale within a family as well as architectural differences across families.

For clarity and standardized comparison, we focus our discussion in Section 4 on the **4B-scale models** from each family. However, a comprehensive analysis covering all evaluated model sizes and configurations is provided in the Supplementary Material.

### 3.4 Attribution Metrics

We evaluate metrics that capture both the statistical accuracy of attribution and the alignment between model's internal and expressed confidence in attribution.

#### 3.4.1 Statistical Accuracy

We extract the row–column indices referenced by the model and compare them against the ground-truth attribution set. We compute cell-level accuracy measure to see how accurately the model returns the correct evidence cells. We also report row-wise and column-wise accuracy scores to gain insight into the model's localization ability. These metrics help understand if the inaccuracy stems from localization errors or failure to pinpoint exact cells. Additionally, we observe cell-wise, row-wise and column-wise precision, recall, and F1 scores which are included in the Supplementary Material.

Collectively, these metrics provide quantitative evidence of current mLLMs' ability to accurately retrieve and localize attribution references.

#### 3.4.2 Confidence-Accuracy Alignment

Confidence-Probability Alignment (Kumar et al., 2024b) refers to the correlation between

a model's internal confidence and its verbalized certainty. We derive *Internal Confidence* directly from answer-level token probabilities, reflecting how strongly the model internally prioritizes a selected attribution. *Verbalized Certainty* is the model's explicit expression of confidence through natural language, elicited via a Confidence Querying Prompt (CQP). High agreement between the two supports transparency and reliability in attribution.

**Internal Confidence:** For each output token $\mathcal{T}_i$, we apply softmax over logits to obtain $P_t(\mathcal{T}_i)$, and define the raw cell confidence $P(c)$ as the geometric mean of token probabilities across the tokens comprising cell $c$. We normalize to obtain

$$P_{IC}(c) = \frac{P(c)}{\sum_{c' \in \mathcal{C}} P(c')},$$

where $\mathcal{C}$ is the set of all table cells. For multi-cell citations, individual $P_{IC}$ scores are aggregated via mean pooling. All confidence values are reported on a 0–100 scale.

**Verbalized Certainty:** Inspired by (Kumar et al., 2024b), our CQP asks the model to select one of six ordinal certainty levels—*Very Certain*, *Fairly Certain*, *Moderately Certain*, *Somewhat Certain*, *Not Certain*, and *Very Uncertain*—mapped linearly to the interval $[0, 100]$ in increments of 20. Providing all candidate cells in the prompt allows the model to contextualize its response and make more informed confidence judgments.

**Confidence Gap:** To quantify alignment, we report the **Confidence Gap**:

$$\Delta = |\,\overline{V} - \overline{I}\,|,$$

where $\overline{V}$ and $\overline{I}$ are the mean verbalized and internal confidence scores across all instances for a given model and prompting strategy. A small $\Delta$ indicates that a model's expressed certainty closely tracks its internal belief; a large $\Delta$ signals miscalibration. The complete CQP formulation is provided in the Supplementary Material.

#### 3.4.3 Family Scaling Score

Per-model attribution accuracy and F1 measure performance at a fixed parameter scale, but they do not capture how reliably a model *family* improves with size. To quantify this, we introduce the **Family Scaling Score (FSS)**, a family-level metric that summarizes scaling behavior across parameter counts.

For each prompting strategy $p \in \mathcal{P}$, we fit a log-linear trend between attribution F1 and model size: $b_p =$ slope of $F_1 \sim \log_{10}(N)$ where $N$ is the number of parameters. A larger positive $b_p$ indicates stronger improvement with scale.

For each family and modality, we summarize prompt-wise slopes by their mean $B$, worst-case $W$, and standard deviation $S$ across prompts.

To penalize unstable scaling, we define

$$R = \frac{1}{1 + \dfrac{S}{|B| + \varepsilon}},$$

where $\varepsilon$ is a small constant. The final score is

$$\text{FSS} = 100\left(\alpha B + (1-\alpha)W\right) R,$$

with $\alpha = 0.7$ and $\varepsilon = 10^{-6}$ in our experiments.

A high FSS indicates that a model family scales well on average, avoids poor worst-case prompt behavior, and remains consistent across prompting strategies. FSS therefore complements fixed-scale attribution metrics by measuring *robust scaling quality* at the family level.

## 4 Results and Discussions

### 4.1 Subpar Attribution Despite Reasonable Question Answering

Our findings show that mLLMs are not inherently bad at question answering (QA) over structured data, but their performance decreases significantly for attribution tasks. As shown in Table 1, QA accuracy remains relatively stable across models and modalities, typically around 50-60%. In contrast, attribution accuracy reported in Table 1 is dramatically lower, ranging from near-random performance in JSON to around 33% in images.

This massive drop shows that poor attribution performance cannot be explained by weak reasoning or answer generation. Instead, models often identify the correct answer but fail to reliably point to the specific rows and columns that support it. Answer correctness and attribution quality therefore appear to be separate capabilities. As a result, models may appear reliable based on QA benchmarks while

Table 1: Model Accuracy in QA vs Attribution in % Across Prompting Strategies Across Open-source Models (for 4B Parameter); **Note**: green depicts overall best model, and red depicts worst.

| Strategy | Model | Markdown | | JSON | | Images | | Average | |
|---|---|---|---|---|---|---|---|---|---|
| | | QA | Attr. | QA | Attr. | QA | Attr. | QA | Attr. |
| Zero Shot | Qwen3-VL | 60.00 | 35.50 | 61.50 | 01.00 | 62.00 | 45.40 | 61.16 | 27.30 |
| | Gemma3 | 43.00 | 13.40 | 40.00 | 00.80 | 28.00 | 16.60 | 37.00 | 10.27 |
| | Molmo2 | 55.50 | 19.50 | 59.50 | 01.00 | 48.50 | 33.60 | 54.50 | 18.03 |
| | InternVL3.5 | 63.50 | 42.50 | 64.00 | 01.50 | 60.50 | 53.10 | 62.66 | 32.37 |
| Few Shot | Qwen3-VL | 62.00 | 21.50 | 61.50 | 01.50 | 60.50 | 34.90 | 61.33 | 19.30 |
| | Gemma3 | 39.50 | 6.20 | 36.00 | 00.60 | 25.00 | 14.20 | 34.00 | 07.00 |
| | Molmo2 | 58.00 | 18.00 | 56.50 | 00.50 | 45.00 | 20.20 | 53.00 | 12.90 |
| | InternVL3.5 | 64.00 | 36.50 | 64.00 | 01.00 | 58.50 | 52.60 | 62.16 | 30.03 |
| CoT | Qwen3-VL | 61.00 | 49.00 | 61.00 | 01.00 | 59.50 | 44.40 | 60.33 | 31.47 |
| | Gemma3 | 41.00 | 10.00 | 38.00 | 00.20 | 27.50 | 14.10 | 35.66 | 08.10 |
| | Molmo2 | 55.50 | 15.50 | 57.50 | 01.00 | 50.00 | 22.90 | 54.33 | 13.13 |
| | InternVL3.5 | 59.00 | 39.00 | 62.50 | 00.50 | 59.00 | 54.70 | 60.33 | 31.40 |
| **Average** | | 55.08 | 25.55 | **55.16** | 00.88 | 48.66 | **33.89** | – | – |

remaining unsuitable for applications that require traceability or auditability such as those in regulated industries like banking, healthcare and law. We discuss potential causes of this disconnect, including training objectives and evaluation practices in Section 5.

### 4.2 Attribution is Easier in Images than in Text

In Table 1, we see that attribution performance varies substantially across input modalities. Models perform best when tables are presented as images, followed by Markdown, with JSON being by far the most difficult format. Average attribution accuracy on JSON is below 1%, compared to over 30% for images.

One likely explanation is that images preserve the spatial & visual hierarchy of tables, allowing models to rely on layout-based cues such as row alignment, column boundaries, & visual grouping. Prior work has shown that multimodal models can effectively leverage spatial structure in document images for tasks such as table understanding and information extraction (Zheng et al., 2024; Wang et al., 2024). By contrast, text formats like JSON lack visual structure and encode hierarchy only through nested text, requiring models to try and understand structure from sequences - a setting that has been to be difficult both theoretically & empirically (Jiang et al., 2025; Hahn, 2020).

Interestingly, this trend reverses for QA accuracy. As shown in Table 1, models often perform better at answering questions in text formats than in images, despite performing worse at attribution in those same formats. One plausible explanation is that QA places weaker grounding requirements than attribution: models can often infer the correct answer from partial cues or broadly relevant context without needing to explicitly identify the supporting evidence (Bohnet et al., 2022; Radevski et al., 2025). Prior work has also suggested that reliably assessing attribution & context grounding is significantly more challenging than answer generation itself (Hu et al., 2025; Vankov et al., 2025). Together, these results suggest that while textual formats are often sufficient for producing correct answers, they pose a significantly greater challenge for precise and reliable attribution.

Overall, models attribute most reliably in image-based tables and struggle in textual formats, particularly JSON. While textual formats support accurate answer generation, they make citation significantly harder. This supports our hypothesis from earlier that question answering and attribution/citation are two distinct tasks.

### 4.3 Models are Better at Citing Rows than Columns

Across modalities (except JSON) and prompting strategies, models are substantially better at identifying the correct row rather than the correct column (Table 2). Averaged across

Table 2: Row vs Column Accuracy in % Across Modalities and Prompting Strategies, Across Open-source Models (for 4B Parameter); **Note**: green depicts overall best model, and red depicts worst.

| | | Markdown | | JSON | | Images | |
|---|---|---|---|---|---|---|---|
| **Strategy** | **Model** | **Row** | **Column** | **Row** | **Column** | **Row** | **Column** |
| Zero Shot | Qwen3-VL | 73.75 | 48.83 | 10.13 | 36.92 | 78.01 | 58.55 |
| | Gemma3 | 70.86 | 23.17 | 05.83 | 21.08 | 47.82 | 30.98 |
| | Molmo2 | 68.75 | 27.58 | 09.16 | 23.33 | 58.51 | 56.13 |
| | InternVL3.5 | 78.25 | 36.00 | 04.04 | 42.33 | 69.24 | 59.79 |
| Few Shot | Qwen3-VL | 77.00 | 29.83 | 10.53 | 11.75 | 79.05 | 46.65 |
| | Gemma3 | 55.37 | 16.33 | 01.83 | 11.83 | 42.51 | 31.28 |
| | Molmo2 | 72.50 | 24.08 | 09.3 | 16.33 | 40.96 | 35.16 |
| | InternVL3.5 | 64.80 | 18.08 | 03.08 | 18.25 | 59.43 | 16.43 |
| CoT | Qwen3-VL | 82.5 | 59.17 | 05.83 | 42.92 | 77.32 | 57.59 |
| | Gemma3 | 64.79 | 17.83 | 03.38 | 21.25 | 46.13 | 35.56 |
| | Molmo2 | 68.80 | 18.00 | 04.88 | 14.00 | 54.73 | 37.55 |
| | InternVL3.5 | 82.25 | 41.50 | 03.24 | 53.83 | 78.77 | 64.82 |
| **Average** | | **71.63** | 30.03 | 05.93 | **26.15** | **61.04** | 44.20 |

models and prompts, row accuracy is approximately 1.3-2× higher than column accuracy for Markdown and image-based tables.

One possible reason for this disparity is that rows and columns play very different roles in a table. Rows often represent complete, meaningful records, such as a single person, product, or transaction. Columns, on the other hand, represent abstract attributes or fields, such as dates, categories, or numerical properties. Prior work on table reasoning has shown that identifying and reasoning about the correct column – often referred to as schema linking – is especially difficult for language models, particularly when column headers are ambiguous or require implicit interpretation (Zhang et al., 2020; Herzig et al., 2020). In contrast, rows are easier to localize because they more closely match how entities and examples are described in natural language.

This consistent disparity suggests that fine-grained attribution – especially identifying the correct field within a record – remains a major unresolved challenge for mLLMs.

Table 3: Internal and Verbal Confidence Scores in % Across Modalities and Prompting Strategies. Difference is calculated as |`avg(verbal) - avg(internal)`| for each row.

| | | Markdown | | JSON | | Images | | Confidence |
|---|---|---|---|---|---|---|---|---|
| **Strategy** | **Model** | **Internal** | **Verbal** | **Internal** | **Verbal** | **Internal** | **Verbal** | **Gap** |
| Zero Shot | Qwen3-VL | 81.36 | 60.8 | 71.55 | 48.94 | 85.72 | 80.94 | 15.98 |
| | Gemma3 | 83.74 | 86.30 | 75.23 | 82.11 | 83.50 | 85.98 | **3.98** |
| | Molmo2 | 53.53 | 64.8 | 38.9 | 43.2 | 45.84 | 96.22 | 21.98 |
| | InternVL3.5 | 65.50 | 61.10 | 47.75 | 32.70 | 65.78 | 73.24 | 4.00 |
| Few Shot | Qwen3-VL | 77.4 | 52.6 | 60.2 | 25.58 | 83.58 | 77.02 | 21.99 |
| | Gemma3 | 88.32 | 83.23 | 64.97 | 100.00 | 83.39 | 85.92 | **10.83** |
| | Molmo2 | 60.31 | 60.6 | 37.94 | 32.6 | 36.75 | 92.94 | 17.04 |
| | InternVL3.5 | 67.79 | 48.54 | 43.49 | 26.47 | 69.54 | 48.86 | 18.98 |
| CoT | Qwen3-VL | 85.39 | 65.85 | 75.41 | 46.37 | 84.11 | 82.77 | 16.64 |
| | Gemma3 | 88.66 | 82.83 | 67.95 | 82.82 | 82.81 | 85.54 | **3.92** |
| | Molmo2 | 56.67 | 55.38 | 28.25 | 41.56 | 49.53 | 90.99 | 17.82 |
| | InternVL3.5 | 77.74 | 61.16 | 66.66 | 30.65 | 83.99 | 74.60 | 20.66 |
| **Average** | | **73.87** | 65.26 | **56.525** | 49.41 | 71.21 | **81.26** | – |

## 4.4 Persistent Miscalibration Between Internal and Verbalized Confidence

Table 3 reports internal and verbalized confidence scores across modalities and prompting strategies, along with the Confidence Gap $\Delta$ for each model–strategy pair. Several patterns emerge consistently. First, internal confidence scores are generally high across all models, often in the 60–88% range, regardless of actual attribution accuracy. This overconfidence mirrors findings from prior work showing that token-level probabilities are not inherently well-aligned with task accuracy (Geng et al., 2024; Groot and Valdenegro Toro, 2024).

Second, the Confidence Gap varies substantially across model families. Gemma3 exhibits the smallest gaps ($\Delta \approx$ 4–11%), indicating relatively consistent internal and verbalized signals, though this consistency does not translate into better attribution: Gemma3 ranks last among model families in attribution accuracy (Table 1). In contrast, Molmo2 and Qwen3-VL show large gaps ($\Delta \approx$ 17–22%), driven primarily by inflated verbal confidence in image settings, where Molmo2's verbalized scores reach up to 97%. InternVL3.5 lies in between, with gaps that fluctuate across prompting strategies.

Third, no modality consistently reduces miscalibration. JSON inputs, despite yielding near-zero attribution accuracy, do not produce uniformly smaller or larger gaps – models remain confidently wrong across the board. Together, these results reinforce that confidence scores, whether internal or verbalized, should not be treated as a reliable proxy for attribution quality (Ye et al., 2024; Kumar et al., 2024a).

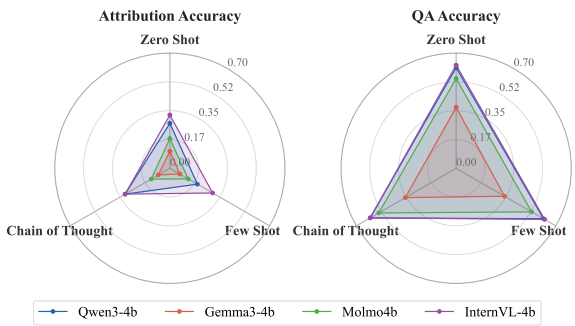


Figure 2: Radar charts comparing model families across attribution accuracy, QA accuracy, and confidence gap (1 − |verbal − internal|) under different prompting strategies.

Figure 2 compares model families across attribution accuracy, QA accuracy, and confidence gap under zero-shot, few-shot, and chain-of-thought prompting. Clear differences emerge across model families, showing that attribution performance depends heavily on the underlying model.

### 4.5 Scaling Behavior Differs Sharply Across Model Families

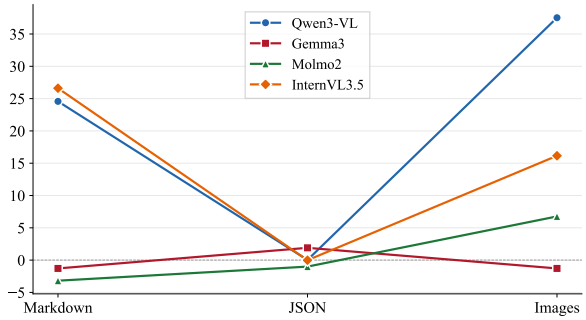


Figure 3: Family Scaling Score (FSS) across model families and table modalities. Higher values indicate stronger and more robust improvement in attribution F1 as model size increases within a family.

Figure 3 shows that attribution scaling depends strongly on both model family and input modality. The clearest pattern is that JSON yields near-zero FSS across all families, indicating that increasing parameter count offers little benefit when table structure is serialized into text. In contrast, image-based inputs produce the strongest scaling gains, with Qwen3-VL improving the most by a large margin. Markdown shows moderate but family-dependent gains.

The family-level differences are equally pronounced. Qwen3-VL exhibits the strongest overall scaling, especially on images. InternVL3.5 also scales well, but more evenly across Markdown and images, suggesting more stable cross-modality improvement. Molmo2 shows only modest gains, concentrated mainly in the image setting. Gemma3 remains nearly flat, and in some cases slightly negative, indicating limited attribution benefit from scale within this family.

These results show that larger models do not automatically yield better attribution. What matters is whether a model family can convert additional scale into robust grounding performance, particularly when spatial structure is preserved. FSS therefore complements fixed-scale evaluation by revealing which families scale *effectively*, not just which models are large.

## 5 Conclusion & Future Works

The gap between question answering and attribution quality reflects how current mLLMs are trained and evaluated. Most instruction-tuning and alignment pipelines optimize models to produce correct and helpful answers, but do not explicitly reward precise citation or faithful attribution to specific data fields (Ouyang et al., 2022). As a result, models can often answer questions correctly without reliably identifying the rows and columns that support those answers. Prior work, such as WebGPT, shows that accurate citation requires task-specific objectives, rather than emerging naturally from standard training pipelines (Nakano et al., 2021). Our proposed benchmark, ViTaB-A, facilitates affirming that QA accuracy and attribution accuracy are separate capabilities, and progress on one does not guarantee progress on the other.

This separation is reinforced by current evaluation practices. Benchmarks such as HELM focus primarily on answer accuracy, robustness, and calibration, while structured attribution and traceability receive little attention (Liang et al., 2022). Although prompting strategies can slightly affect attribution, they do not close the gap, suggesting that inference-time methods alone are insufficient. This creates a feedback loop where models are optimized and compared mainly on QA accuracy, even though attribution remains unreliable.

These findings point to several directions for future work. Attribution should be treated as a first-class objective, with training signals that directly optimize row and column localization. Finally, the high verbal confidence models express despite incorrect attribution raises concerns for user trust, especially in high-stakes domains. Overall, our results suggest that improving QA accuracy alone is not sufficient and that reliable structured attribution requires dedicated research attention.

## Limitations

While ViTaB-A provides a systematic evaluation of mLLM attribution across modalities, several directions remain open for future work. The benchmark is constructed from HiTab, which offers well-structured, annotated evidence cells well-suited for controlled attribution evaluation; extending this framework to noisier real-world tables is a natural next step. The near-random JSON attribution performance surfaces an interesting open question regarding whether serialization format, tokenization, or pretraining distribution is the primary contributing factor, which have been briefly discussed but we leave for future investigation to be analyzed in-depth. Our evaluation focuses on open-source model families, and applying ViTaB-A to proprietary models represents a straightforward extension of this framework. Finally, while our confidence alignment analysis reveals a consistent gap between expressed certainty and attribution correctness, developing methods to close this gap is an promising direction that this benchmark is designed to facilitate.

## Acknowledgments

The authors used AI assistants (Claude Code / Gemini / ChatGPT) to support code development and to refine the writing of this paper.

## References

Bernd Bohnet, Vinh Q Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Massimiliano Ciaramita, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, and 1 others. 2022. Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint arXiv:2212.08037*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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

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

Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, and Christos Faloutsos. 2024. Large Language Models (LLMs) on Tabular Data: Prediction, Generation, and Understanding–A Survey. *arXiv preprint arXiv:2402.17944*.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2024. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595, Mexico City, Mexico. Association for Computational Linguistics.

Tobias Groot and Matias Valdenegro Toro. 2024. Overconfidence is key: Verbalized uncertainty evaluation in large language and vision-language models. In *Proceedings of the 4th Workshop on Trustworthy Natural Language Processing (TrustNLP 2024)*, pages 145–171, Mexico City, Mexico. Association for Computational Linguistics.

Michael Hahn. 2020. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4320–4333.

Nan Hu, Jiaoyan Chen, Yike Wu, Guilin Qi, Hongru Wang, Sheng Bi, Yongrui Chen, Tongtong Wu, and Jeff Z Pan. 2025. Can LLMs Evaluate Complex Attribution in QA? Automatic Benchmarking using Knowledge Graphs. In *The 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025)*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Zhuohang Jiang, Pangjing Wu, Ziran Liang, Peter Q Chen, Xu Yuan, Ye Jia, Jiancheng Tu, Chen Li, Peter HF Ng, and Qing Li. 2025. Hibench: Benchmarking llms capability on hierarchical structure reasoning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 5505–5515.

Abhishek Kumar, Robert Morabito, Sanzhar Umbet, Jad Kabbara, and Ali Emami. 2024a. Confidence under the hood: An investigation into the confidence-probability alignment in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 315–334, Bangkok, Thailand. Association for Computational Linguistics.

Abhishek Kumar, Robert Morabito, Sanzhar Umbet, Jad Kabbara, and Ali Emami. 2024b. Confidence Under the Hood: An Investigation into the Confidence-Probability Alignment in Large Language Models. *Preprint*, arXiv:2405.16282.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, and 1 others. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Weichen Liu, Qiyao Xue, Haoming Wang, Xiangyu Yin, Boyuan Yang, and Wei Gao. 2025. Spatial Reasoning in Multimodal Large Language Models: A Survey of Tasks, Benchmarks and Methods. *Preprint*, arXiv:2511.15722.

Yu Liu, Duantengchuan Li, Kaili Wang, Zhuoran Xiong, Fobo Shi, Jian Wang, Bing Li, and Bo Hang. 2024. Are LLMs good at structured outputs? A benchmark for evaluating structured output capabilities in LLMs. *Information Processing & Management*, 61(5):103809.

Puneet Mathur, Alexa Siu, Nedim Lipka, and Tong Sun. 2024. Matsa: Multi-agent table structure attribution. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 250–258.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, and 1 others. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Gorjan Radevski, Kiril Gashteovski, Shahbaz Syed, Christopher Malon, Sebastien Nicolas, Chia-Chien Hung, Timo Sztyler, Verena Heußer, Wiem Ben Rim, Masafumi Enomoto, and 1 others. 2025. On Synthesizing Data for Context Attribution in Question Answering. *arXiv preprint arXiv:2504.05317*.

Ivan Vankov, Matyo Ivanov, Adriana Correia, and Victor Botev. 2025. ConSens: Assessing context grounding in open-book question answering. In *International Conference on Artificial Neural Networks*, pages 151–163. Springer.

Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, and Xiaomo Liu. 2024. DocLLM: A layout-aware generative language model for multimodal document understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8529–8548, Bangkok, Thailand. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Fanghua Ye, Mingming Yang, Jianhui Pang, Longyue Wang, Derek F. Wong, Emine Yilmaz, Shuming Shi, and Zhaopeng Tu. 2024. Benchmarking LLMs via Uncertainty Quantification. *Preprint*, arXiv:2401.12794.

Hongzhi Zhang, Yingyao Wang, Sirui Wang, Xuezhi Cao, Fuzheng Zhang, and Zhongyuan Wang. 2020. Table fact verification with structure-aware transformer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1624–1629.

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

# A Appendix

# B Attribution Task Prompts

### Prompt A: Zero-Shot Technique

```
You are a table analysis expert. Your task is to identify
which cell(s) in the table contain or support the given
answer to the question.

TABLE: {table}

QUESTION: {question}
ANSWER: {answer}

TASK: Identify the cell coordinate(s) that contain or
directly support this answer. Use Excel-style coordinates
where columns are letters (A, B, C, ...) and rows are
numbers (1, 2, 3, ...).

RESPONSE FORMAT: Return ONLY the cell coordinates in Excel
formula format.

Examples:
- Single cell: "=E7" or "=B3"
- Multiple cells: "=A2" or list them separately: "=A2, =B2,
=C2"
- If the answer involves a formula (sum, average, etc.),
you may use: "SUM(C3:C10)" or "=A1+B2"

IMPORTANT: Do NOT repeat the question, table, or
instructions. Output ONLY the cell coordinates.

ATTRIBUTED CELLS:
```

### Prompt B: Few-Shot Technique

```
You are a table analysis expert. Your task is to identify
which cell(s) in the table contain or support the given
answer to the question.

Here is an example:

EXAMPLE:
TABLE: {example table}
QUESTION: {example question}
ANSWER: {example answer}
ATTRIBUTED CELLS: {example cells}

Now analyze this table:

TABLE: {table}

QUESTION: {question}
ANSWER: {answer}

IMPORTANT: Do NOT repeat the example, question, table, or
instructions. Output ONLY the cell coordinates in formula
format.

ATTRIBUTED CELLS:
```

### Prompt C: Chain of Thought Technique

```
You are a table analysis expert. Your task is to identify
which cell(s) in the table contain or support the given
answer to the question.

TABLE: {table}

QUESTION: {question}
ANSWER: {answer}

Let's think step by step:

1. First, understand what the question is asking for.
2. Then, locate where the answer "\{answer\}" appears
or can be derived from in the table.
3. Identify the specific cell coordinate(s) using Excel-
style notation (columns as letters A, B, C... and rows as
numbers 1, 2, 3...).
4. If the answer is computed from multiple cells (e.g., a
sum), express it as a formula like "SUM(C3:C10)" or
"=A1+B2".
5. For simple cell references, use the format "=E7" or
"=B3".

IMPORTANT: Do NOT repeat the question or table in your
reasoning.

REASONING:

Based on the above reasoning, provide ONLY the final cell
coordinates in Excel formula format (e.g., "=E7",
"SUM(C3:C10)", or "=A1+B2").
Do NOT repeat your reasoning or the question.

ATTRIBUTED CELLS:
```

### Prompt D: Confidence Querying Prompt (CQP)

```
A language model was analyzing a table to identify cells
that support an answer.
TABLE: {table}
QUESTION: {question}
ANSWER: {answer}
All possible cell coordinates in this table:
{all cells list}
The model selected the following cell(s): {predicted cells}

Critically analyze whether the identified cells correctly
support the answer given all the available cells. Check
for missed cells or inclusion of irrelevant cells.
How certain are you that the model's cell selection is
completely correct?

a. Very Certain (No doubts at all)
b. Fairly Certain (Minor doubts)
c. Moderately Certain (Some doubts)
d. Somewhat Certain (Significant doubts)
e. Not Certain (Likely incorrect)
f. Very Uncertain (Definitely incorrect)

Answer with just the letter (a-f):
```

# C Confidence Scores:

## C.1 Conformal Prediction for Uncertainty Quantification

While attribution metrics evaluate whether a model's cited cell supports the answer, they do not quantify how uncertain the response is. Therefore, we study uncertainty quantification (UQ) as presented by (Ye et al., 2024). We employ split-conformal prediction, which converts the models per-cell confidence scores to a prediction set $C(x)$ which uses a user-controlled target error rate $\alpha$.

**Setup and Adaptations:** We utilize the standard split-conformal partition of calibration ($\mathcal{D}_{\text{cal}}$) and test ($\mathcal{D}_{\text{test}}$) sets. However, we introduce two specific adaptations to handle the nature of generative table attribution:

- **Multi-Cell Coverage:** Unlike standard classification where the label is a single token, a ground truth answer in our task may span a region of cells. We therefore define the coverage criterion as satisfied if *any* ground truth cell is present in the predicted set $C(x)$.
- **Open-Vocabulary Approximation:** Since our model operates over an open vocabulary, and not a fixed label set, computing the normalizing constant over all possible table coordinates is computationally complex. We instead adopt a sparse approximation where probability mass is estimated only for the tokens actively generated by the model, assigning zero probability to non-generated coordinates.

**Scoring Functions:** We implement two non-conformity scoring functions, adapted from Ye et al. (2024) to support our multi-cell coverage definition:

- **Least Ambiguous Class (LAC):** This method constructs prediction sets based on absolute probability thresholds. We define the non-conformity score $s_i$ as one minus the probability of the *most likely* correct cell:

$$s_i = 1 - \max_{y \in Y_{\text{true}}} \hat{p}(y \mid x) \tag{1}$$

  The prediction set is constructed by including all cells with probability $\hat{p}(y|x) \geq 1 - \hat{q}$, where $\hat{q}$ is the empirical quantile of scores over $\mathcal{D}_{\text{cal}}$.

- **Adaptive Prediction Sets (APS):** This method accumulates probability mass from the sorted predictions to account for the tail of the distribution. We define the non-conformity score $s_i$ as the minimum cumulative mass required to reach *any* valid ground truth cell:

$$s_i = \min_{y \in Y_{\text{true}}} A(y) \tag{2}$$

  where $A(y)$ is the cumulative probability mass of candidate cells sorted in descending order. The prediction set includes candidates until the cumulative mass exceeds the calibrated threshold $\hat{q}$.

**Results:** Table 5 reports uncertainty quantification results across representations and prompting strategies using prediction set (SS) and coverage rate (CR). Across all representations and prompting methods, coverage rates remain close to the target level, while prediction set sizes vary substantially by modality, with image based inputs consistently producing larger sets than markdown and JSON. This indicates higher attribution uncertainty under visual perturbations.

## C.2 Complete Attribution Metrics for All Model Families

Table 5, 6, 7, and 8 report the attribution metrics scores for the Gemma3, Qwen3-VL, Molmo2, and InternVL3.5 model families respectively.

Table 4: Model uncertainty quantification results for attribution across representations and prompting strategies. We report average prediction set size (SS, number of cells) and coverage rate (CR, %).

| | | **Markdown** | | **JSON** | | **Images** | |
|---|---|---|---|---|---|---|---|
| **Strategy** | **Model** | **SS** | **CR** | **SS** | **CR** | **SS** | **CR** |
| Zero Shot | Qwen3-VL | 184.96 | 83.50 | 186.58 | 79.50 | 187.22 | 87.50 |
| | Gemma3 | 165.61 | 80.32 | 183.73 | 80.72 | 186.49 | 83.70 |
| | Molmo2 | 184.11 | 83.50 | 187.80 | 79.00 | 187.53 | 90.10 |
| | InternVL3.5 | 183.09 | 80.50 | 185.54 | 81.50 | 177.33 | 79.00 |
| Few Shot | Qwen3-VL | 183.67 | 82.50 | 188.79 | 81.00 | 188.16 | 87.30 |
| | Gemma3 | 171.83 | 80.12 | 182.26 | 80.12 | 186.48 | 86.80 |
| | Molmo2 | 187.31 | 85.00 | 185.07 | 78.00 | 188.16 | 83.10 |
| | InternVL3.5 | 174.46 | 81.50 | 177.44 | 79.00 | 177.33 | 79.00 |
| Chain of Thought | Qwen3-VL | 173.60 | 84.00 | 179.17 | 80.50 | 184.89 | 88.80 |
| | Gemma3 | 183.27 | 79.32 | 180.59 | 78.51 | 185.97 | 83.70 |
| | Molmo2 | 181.05 | 81.50 | 188.55 | 80.50 | 184.55 | 83.50 |
| | InternVL3.5 | 185.15 | 77.00 | 187.21 | 78.50 | 186.40 | 77.00 |

Table 5: Attribution Metrics for Gemma Model Family

| | | **Markdown** | | **JSON** | | **Images** | |
|---|---|---|---|---|---|---|---|
| **Strategy** | **Model** | **F1** | **Accuracy** | **F1** | **Accuracy** | **F1** | **Accuracy** |
| Zero Shot | Gemma3-4b | 0.159 | 13.40 | 0.113 | 0.80 | 0.887 | 16.60 |
| | Gemma3-12b | 0.306 | 27.20 | 0.141 | 0.60 | 0.396 | 37.10 |
| | Gemma3-27b | 0.372 | 31.00 | 0.279 | 1.51 | 0.464 | 44.10 |
| Few Shot | Gemma3-4b | 0.713 | 6.20 | 0.006 | 0.60 | 0.160 | 14.20 |
| | Gemma3-12b | 0.247 | 22.80 | 0.004 | 0.40 | 0.396 | 36.90 |
| | Gemma3-27b | 0.279 | 26.50 | 0.002 | 0.00 | 0.393 | 37.13 |
| Chain of Thought | Gemma3-4b | 0.11 | 10.00 | 0.0036 | 0.20 | 0.16 | 14.10 |
| | Gemma3-12b | 0.26 | 24.20 | 0.0087 | 0.60 | 0.42 | 39.90 |
| | Gemma3-27b | 0.47 | 44.50 | 0.0125 | 1.00 | 0.47 | 45.90 |
| **Average** | | 0.253 | 22.87 | 0.01 | 0.63 | 0.340 | 31.77 |

Table 6: Attribution Metrics for Qwen3-VL Model Family

| | | Markdown | | JSON | | Images | |
|---|---|---|---|---|---|---|---|
| **Strategy** | **Model** | **F1** | **Accuracy** | **F1** | **Accuracy** | **F1** | **Accuracy** |
| Zero Shot | Qwen3-VL-2b | 0.10 | 8.50 | 0.005 | 0.50 | 0.15 | 14.10 |
| | Qwen3-VL-4b | 0.379 | 35.50 | 0.015 | 1.00 | 0.48 | 45.40 |
| | Qwen3-VL-8b | 0.261 | 23.50 | 0.004 | 0.00 | 0.40 | 37.50 |
| | Qwen3-VL-32b | 0.515 | 49.50 | 0.021 | 1.00 | 0.72 | 69.70 |
| Few Shot | Qwen3-VL-2b | 0.068 | 6.50 | 0.010 | 1.00 | 0.09 | 8.20 |
| | Qwen3-VL-4b | 0.240 | 21.50 | 0.019 | 1.50 | 0.38 | 34.90 |
| | Qwen3-VL-8b | 0.281 | 25.50 | 0.024 | 1.50 | 0.30 | 27.20 |
| | Qwen3-VL-32b | 0.390 | 36.00 | 0.003 | 0.00 | 0.69 | 68.10 |
| Chain of Thought | Qwen3-VL-2b | 0.220 | 20.00 | 0.003 | 0.00 | 0.32 | 29.80 |
| | Qwen3-VL-4b | 0.515 | 49.00 | 0.016 | 1.00 | 0.47 | 44.40 |
| | Qwen3-VL-8b | 0.570 | 52.50 | 0.001 | 0.00 | 0.68 | 64.80 |
| | Qwen3-VL-32b | 0.685 | 59.00 | 0.018 | 1.00 | 0.77 | 70.10 |
| **Average** | | 0.35 | 32.25 | 0.012 | 0.71 | 0.46 | 42.85 |

Table 7: Attribution Metrics for Molmo2 Model Family

| | | Markdown | | JSON | | Images | |
|---|---|---|---|---|---|---|---|
| **Strategy** | **Model** | **F1** | **Accuracy** | **F1** | **Accuracy** | **F1** | **Accuracy** |
| Zero Shot | Molmo2-4b | 0.218 | 19.5 | 0.013 | 1.00 | 0.355 | 33.6 |
| | Molmo2-8b | 0.222 | 20.00 | 0.015 | 0.05 | 0.369 | 34.80 |
| Few Shot | Molmo2-4b | 0.196 | 18.00 | 0.01 | 0.50 | 0.215 | 20.20 |
| | Molmo2-8b | 0.096 | 8.50 | 0.00 | 0.00 | 0.243 | 22.80 |
| Chain of Thought | Molmo2-4b | 0.162 | 15.50 | 0.019 | 1.00 | 0.243 | 22.90 |
| | Molmo2-8b | 0.205 | 19.50 | 0.013 | 0.50 | 0.3376 | 31.70 |
| **Average** | | 0.183 | 16.83 | 0.011 | 0.58 | 0.294 | 27.66 |

Table 8: Attribution Metrics for InternVL3.5 Model Family

| | | Markdown | | JSON | | Images | |
|---|---|---|---|---|---|---|---|
| **Strategy** | **Model** | **F1** | **Accuracy** | **F1** | **Accuracy** | **F1** | **Accuracy** |
| Zero Shot | InternVL3.5-4b | 0.273 | 42.50 | 0.013 | 1.50 | 0.441 | 53.10 |
| | InternVL3.5-8b | 0.193 | 17.50 | 0.016 | 1.00 | 0.254 | 22.50 |
| | InternVL3.5-14b | 0.44 | 46.50 | 0.026 | 1.50 | 0.557 | 56.30 |
| | InternVL3.5-38b | 0.595 | 56.50 | 0.023 | 1.50 | 0.616 | 59.40 |
| Few Shot | InternVL3.5-4b | 0.125 | 36.50 | 0.008 | 1.00 | 0.136 | 52.60 |
| | InternVL3.5-8b | 0.190 | 16.50 | 0.015 | 1.00 | 0.277 | 25.30 |
| | InternVL3.5-14b | 0.375 | 37.00 | 0.013 | 0.50 | 0.533 | 57.30 |
| | InternVL3.5-38b | 0.448 | 42.50 | 0.013 | 1.00 | 0.541 | 52.50 |
| Chain of Thought | InternVL3.5-4b | 0.364 | 39.00 | 0.01 | 0.50 | 0.534 | 54.70 |
| | InternVL3.5-8b | 0.379 | 36.00 | 0.017 | 1.00 | 0.49 | 45.80 |
| | InternVL3.5-14b | 0.447 | 40.50 | 0.018 | 1.00 | 0.582 | 55.40 |
| | InternVL3.5-38b | 0.607 | 58.00 | 0.003 | 0.00 | 0.646 | 59.5 |
| **Average** | | 0.369 | 39.08 | 0.014 | 0.95 | 0.467 | 49.53 |