

M3Grounder: Mask-Based Multi-Span and Multi-Granular Grounding for Document QA

Venkata Kesav Venna^{1,*} Sai Madhusudan Gunda^{2,*} Jyothi Swaroopa Jinka^{2,†}
 Hrithik Sagar Rachakonda^{2,†} Anirudh Srinivasan¹ Ravi Kiran Sarvadevabhatla^{1,2}
¹ BharatGen ² IIIT Hyderabad
 {venkat.kesav, anirudh.srinivasan}@titiitb.org ravi.kiran@iiit.ac.in
 {sai.gunda, jinka.swaroopa, hrithik.rachakonda}@research.iiit.ac.in
 * Equal contribution † Equal contribution

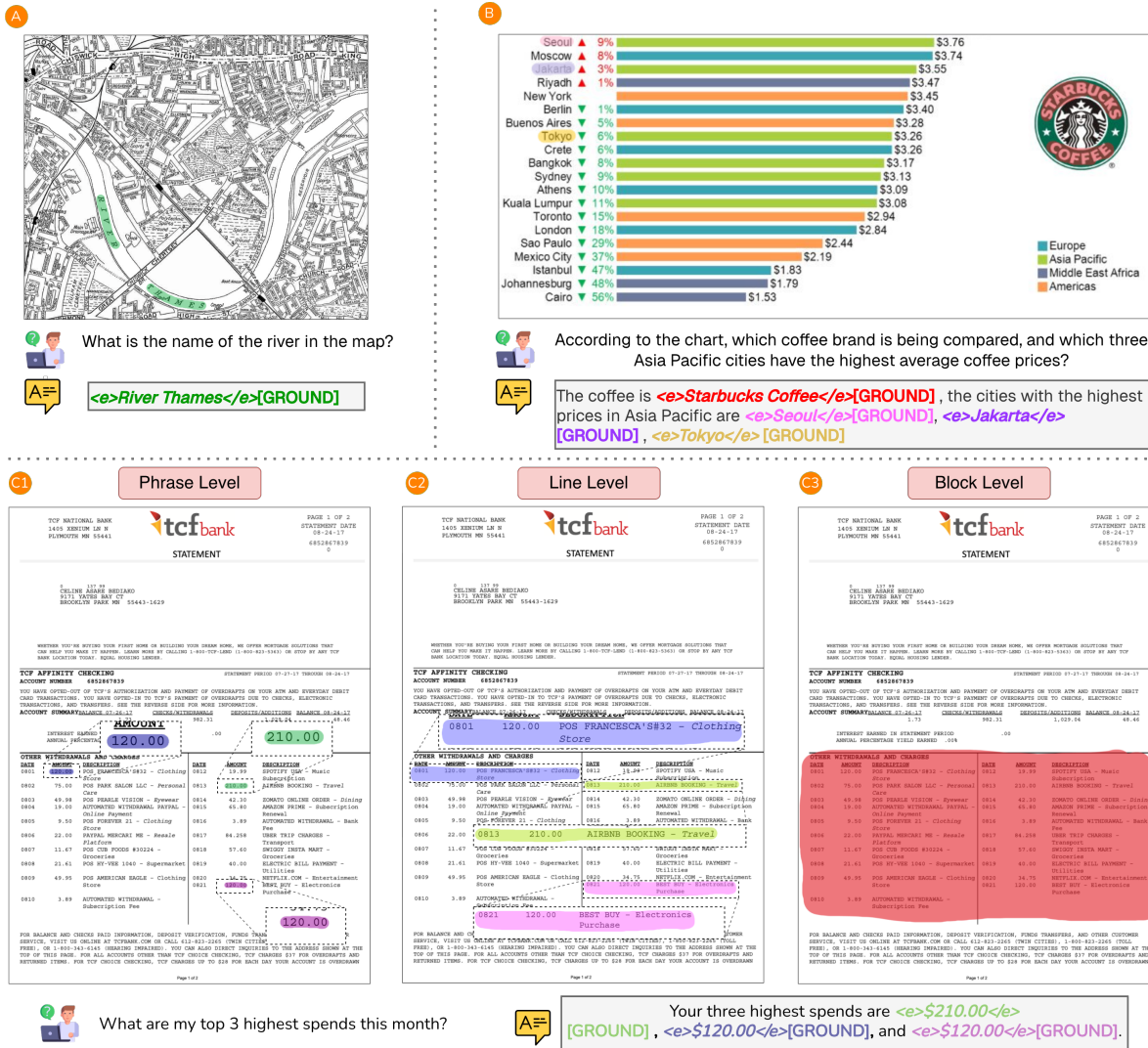


Figure 1. **M3Grounder in action:** Each example shows a QA pair. The predicted answer text contains interleaved [GROUND] tokens which map answer spans to corresponding grounding regions. **A**, **B** demonstrates precise segmentation without spillover into irrelevant regions. **B** shows effective grounding for dense, multi-span evidence in complex document layouts. **C1**, **C2**, and **C3** illustrate multi-granular grounding, where the grounding scope expands hierarchically (phrase \subset line \subset block).

Contents

1. Overview	3
2. Document Grounding	3
2.1. Current Approaches	3
2.2. Segmentation Based Grounding	3
3. Method	3
3.1. Formulation	3
3.2. Multi-Granularity Grounding	4
3.3. Training Objective	4
3.4. Training Implementation	5
3.5. Latency Evaluation	5
4. Dataset Generation Pipeline	5
4.1. Layout-Aware Documents	9
4.2. Curved Text Documents	16
4.3. Charts	17
4.4. Data Verification	21
5. Baselines Implementation	22
5.1. Prompts	22
6. G-Eval for Answer Quality (AQ)	25
7. More details on benchmarks and evaluations	26
7.1. BoundingDocs-Test	27
7.2. DOGR-Bench	27
7.3. MMDocBench	28
7.4. GroundingDocQA-Bench	28
8. Qualitative Results	29
8.1. Comparisons Against the Next Best Model	30
8.2. Standalone Predictions from M3Grounder	39
9. Miscellaneous	46

1. Overview

This supplementary document provides additional details on M3Grounder, GroundingDocQA, GroundingDocQA-Bench, expanding each component presented in the main paper. We first formalize grounding as a multi-granular segmentation task, followed by in depth architectural details and the motivation for our additional loss components. We then describe our data generation pipelines for layout aware, curved text, and chart documents, along with the data verification pipeline to ensure high quality grounded QA pairs. Subsequent sections provide baseline implementations, benchmark specific evaluation settings, and further qualitative examples.

2. Document Grounding

Document grounding refers to the task of identifying the exact visual region within a document that supports a predicted answer. Grounding ensures interpretability by showing where the model obtained the answer from, and enables more understanding across diverse document types such as reports, forms, tables, webpages, posters, and charts.

2.1. Current Approaches

Existing grounding methods treat the problem through the lens of OCR span matching or bounding box prediction. In span matching pipelines, the model predicts an answer string and grounding is obtained by locating this string in the OCR output. This strategy collapses whenever the same phrase appears multiple times, when OCR misses or distorts text, or when the underlying text is curved, rotated, or visually complex. Bounding-box prediction improves over span matching but remains limited: boxes cannot represent curved or skewed regions. And also these methods operate at only one granularity and rely on coarse shapes, they fail to capture the true intent and structure of grounding in real documents.

2.2. Segmentation Based Grounding

Segmentation based grounding addresses these limitations by predicting pixel-level masks that accurately follow the shape and geometry of the supporting regions. Masks can capture curved text, irregular boundaries, and disjoint spans far more precisely than bounding boxes. By combining pixel-level precision with multi-granular hierarchy, segmentation-based grounding provides a more faithful and expressive representation of evidence, enabling models to better capture the structure and semantics of complex document layouts.

3. Method

Grounding in document question answering requires the model to produce not only an accurate answer but also the exact visual region that justifies that answer. Instead of direct coupling between text generation and grounding prediction such as embedding coordinates inside the textual output, M3Grounder avoids this by *decoupling* the two tasks.

Key idea: The VLM determines the answer and provides grounding cues for each answer span, and the segmentation module uses these cues to generate pixel-level masks for the corresponding regions in the document. The VLM and segmentation module interact through [GROUND] tokens: the VLM emits a [GROUND] token immediately after each answer span, and the segmentation module uses the token’s hidden state to generate the corresponding phrase, line, and block-level grounding masks.

3.1. Formulation

Given a document image x and question q , the VLM autoregressively produces an output sequence in which the answer is decomposed into explicit answer spans. Formally,

$$\hat{a} = (\dots \langle e \rangle y_k \langle /e \rangle [\text{GROUND}], \dots). \quad (1)$$

where $\langle e \rangle$ and $\langle /e \rangle$ denotes the start and end of the answer span y_k within the generated sequence. [GROUND] token appears immediately after each answer span y_k and signals the model to generate its corresponding hierarchical grounding masks.

For each [GROUND] token generated, the corresponding hidden state from the last layer of the VLM $\tilde{\mathbf{h}}_k$ is passed through three granularity specific MLP projection heads $\{\gamma_i\}_{i \in \mathcal{G}}$ where $\mathcal{G} = \{p, l, b\}$ (p, l, b corresponds to phrase, line and block levels respectively). These projections produce hierarchical grounding prompt embeddings $\mathbf{h}_k^{(i)}$ for phrase (p), line (l), and block (b) levels.

$$\text{for } i \in \mathcal{G} : \mathbf{h}_k^{(i)} = \gamma_i(\tilde{\mathbf{h}}_k) \quad (2)$$

In parallel, the segmentation module’s vision encoder \mathcal{F}_{enc} processes the input document image \mathbf{x} to extract dense visual features $\mathbf{z} = \mathcal{F}_{\text{enc}}(\mathbf{x})$. The grounding prompt embeddings $\mathbf{h}_k^{(i)}$ and image features \mathbf{z} are provided as input to the segmentation module’s mask decoder \mathcal{F}_{dec} to predict grounding masks at multiple granularities:

$$\text{for } i \in \mathcal{G} : \hat{\mathbf{M}}_k^{(i)} = \mathcal{F}_{\text{dec}}(\mathbf{h}_k^{(i)}, \mathbf{z}) \quad (3)$$

This yields a hierarchy of grounding masks $\{\hat{\mathbf{M}}_k^{(p)}, \hat{\mathbf{M}}_k^{(l)}, \hat{\mathbf{M}}_k^{(b)}\}$ that capture phrase-, line-, and block-level spatial grounding for each [GROUND] token.

3.2. Multi-Granularity Grounding

Documents exhibit a natural hierarchical structure: words form lines, and lines form coherent blocks such as paragraphs, table rows, item groups, or semantic sections. Grounding answer spans at a single scale is therefore insufficient, different intents rely on different scopes of visual context. M3Grounder addresses this by predicting grounding masks at three complementary granularities: phrase, line, and block. Figure 1 C1-C3 illustrates how the grounding region expands hierarchically for the same answer span.

Phrase Level (Fine Grained Evidence): The phrase mask captures the *minimal* text region supporting an answer. This includes individual words, numbers, monetary amounts, table-cell values, or short key value fields. In the example (C1), M3Grounder isolates each numeric spend (e.g., “120.00”, “210.00”) with crisp, shape-preserving masks, even when values appear in dense or cluttered rows. Phrase-level grounding is essential for extractive QA, where the answer corresponds to a precise textual span.

Line Level (Local Context Evidence): Some require context beyond a single phrase, such as disambiguating repeated values or understanding the semantic role of the number within its line. The line mask corresponds to the *entire visual line* containing the answer. In Figure 1 C2, the model highlights full transaction rows, including merchant names, timestamps, and categories. This granularity helps the model associate a spend amount with its transaction type and disambiguate identical numeric values that occur in multiple lines.

Block Level (Global Context Evidence): Block masks expand grounding beyond individual lines to larger structural units such as multi line form fields, table sections, grouped transactions, or entire semantic blocks. This is crucial for reasoning-oriented questions that require summarizing or comparing values across a broader region. In Figure 1 C3, the model correctly identifies the full group of high-spend transactions, showing that block level grounding captures the layout level context needed for multi-span QA pairs. The block region encompasses multiple rows while excluding irrelevant parts of the page.

3.3. Training Objective

M3Grounder is trained end-to-end to jointly optimize answer generation and hierarchical mask prediction. The overall objective combines four complementary components:

$$\mathcal{L} = \lambda_{\text{lm}} \mathcal{L}_{\text{lm}} + \lambda_{\text{seg}} \mathcal{L}_{\text{seg}} + \lambda_{\text{bleed}} \mathcal{L}_{\text{bleed}} + \lambda_{\text{hier}} \mathcal{L}_{\text{hier}}$$

Each term addresses a different aspect of M3Grounder

Language Modeling: The answer sequence is supervised using standard cross-entropy loss [31]: This term ensures that the model generates accurate answer text, correctly places span delimiters ($\langle e \rangle$, $\langle /e \rangle$), and appropriate number of [GROUND] tokens, and preserves the structure of multi-span answers.

Segmentation: For each grounded span k and granularity $i \in \mathcal{G} = \{p, l, b\}$, the corresponding mask $\hat{\mathbf{M}}_k^{(i)}$ is supervised using a combination of Dice [24] and BCE [31] losses:

$$\mathcal{L}_{\text{seg}} = \sum_{i \in \mathcal{G}} \sum_{k=1}^K \left(\lambda_{\text{dice}}^{(i)} \ell_{\text{dice}}(\hat{\mathbf{M}}_k^{(i)}, \mathbf{M}_k^{(i)}) + \lambda_{\text{bce}}^{(i)} \ell_{\text{bce}}(\hat{\mathbf{M}}_k^{(i)}, \mathbf{M}_k^{(i)}) \right)$$

Bleed Suppression: SAM struggles in infographic rich documents, where visual , non-textual elements, icons often cause masks to spill into nearby regions. Since SAM is not trained for fine grained text localization, it is susceptible to bleed into such regions. To address this, bleed suppression loss penalizes any predicted foreground outside the reference text region, providing an additional constraint for document text.

Let \mathbf{M}_{pred} denote the predicted mask and \mathbf{M}_{ref} the union of all ground-truth text pixels. The *Bleed* is defined as

$$\text{Bleed} = \mathbf{M}_{\text{pred}} \odot (1 - \mathbf{M}_{\text{ref}})$$

Constraining this helps the segmentation head to stay tightly aligned with true text shapes and prevents leakage into surrounding graphical content, yielding cleaner masks. The final bleed loss (Fig. 2) is formulated as:

$$\mathcal{L}_{\text{bleed}} = \sum_{i \in \mathcal{G}} \sum_{k=1}^K \frac{\sum_{j \in \Omega} \hat{\mathbf{M}}_k^{(i)}(j) [1 - \mathbf{M}_{\text{ref}}(j)]}{\sum_{j \in \Omega} \hat{\mathbf{M}}_k^{(i)}(j) + \epsilon}$$

Hierarchical Enclosure: Since phrase, line, and block masks represent nested levels of the same answer evidence, we explicitly enforce containment. The hierarchical enclosure loss (Fig. 3) penalizes pixels where a finer mask extends beyond its coarser counterpart:

$$\mathcal{L}_{\text{hier}} = \sum_{k=1}^K \sum_{(i,j) \in \{(p,l),(l,b)\}} \frac{\sum_{m \in \Omega} \hat{\mathbf{M}}_k^{(i)}(m) [1 - \mathbf{M}_k^{(j)}(m)]}{\sum_{m \in \Omega} \hat{\mathbf{M}}_k^{(i)}(m) + \epsilon}.$$

This constraint stabilizes multi-granular predictions and prevents contradictory mask shapes across scales. In visually dense layouts such as double-column pages, tightly packed lines etc , line-level masks may drift into an adjacent column, or phrase-level masks may partially overlap with a neighboring line. The hierarchical loss is constructed specifically to suppress such cross-level leakage. By forcing finer masks to remain strictly within their parent masks, it ensures that even in cluttered layouts the model preserves structural consistency and assigns each phrase, line, and block to the correct spatial hierarchy.

3.4. Training Implementation

We train M3Grounder end-to-end using our hybrid language segmentation objective on the full 2M multi-span, multi-granular QA pairs of the GroundingDocQA. All experiments are conducted on a cluster of 64×NVIDIA H100 (80GB) GPUs using DeepSpeed ZeRO-3 for full parameter, gradient, and optimizer state sharding. The vision encoder of the VLM backbone is kept frozen during training. Within the VLM stack, only the LM decoder is updated. Within the segmentation stack, only the mask decoder is trainable. All remaining segmentation components remain frozen. We train for 1 epoch over the full dataset using a per-device batch size of 2, yielding an effective global batch size of 128. We use the AdamW [14] optimizer with a learning rate of 2×10^{-6} , warmup ratio of 3%, cosine decay schedule, zero weight decay, and gradient norm clipping of 1.0. Following the design of the original SAM implementation, we keep the entire segmentation module (prompt encoder, mask decoder, and the associated heads) in FP32. This matches SAM’s publicly released training configuration, where the mask decoder is stored and executed in full precision. All remaining components of the VLM backbone operate in bfloat16 mixed precision, with FP32 master weights maintained by ZeRO.

3.5. Latency Evaluation

We compare inference latency between the vanilla Qwen3-VL [36] backbone and our full M3Grounder model. All measurements are averaged over GroundingDocQA-Bench on a single NVIDIA H100 GPU. Qwen3-VL requires 445 ms per sample, whereas M3Grounder with hierarchical projection layers and SAM-based mask decoding runs at 508 ms per sample, incurring only +63 ms additional latency while providing multi-level grounding capability. This overhead is small relative to the substantial gains in grounding. Demonstrating that grounding-aware decoding can be added to larger VLMs with minimal computational cost.

4. Dataset Generation Pipeline

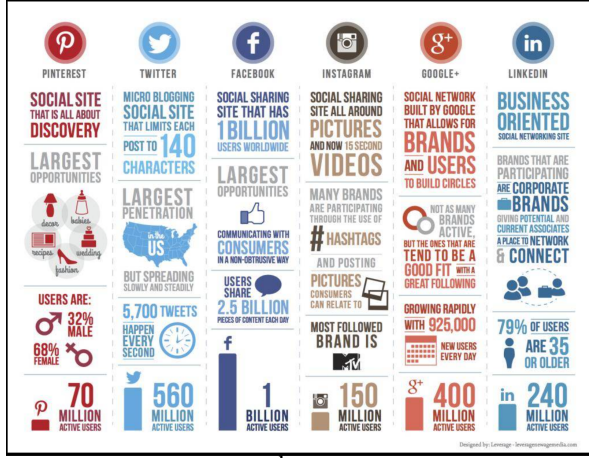
Most existing document grounding datasets [10, 48] rely on OCR text extraction followed by span to bounding box matching. The grounding region is determined by locating where a token string appears in the OCR output. This approach fails in several practical cases:(1) Many documents contain repeated key phrases (e.g., “Total”, “Address”,



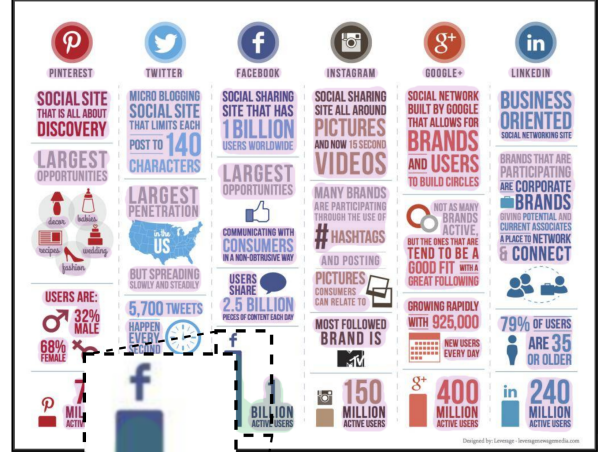
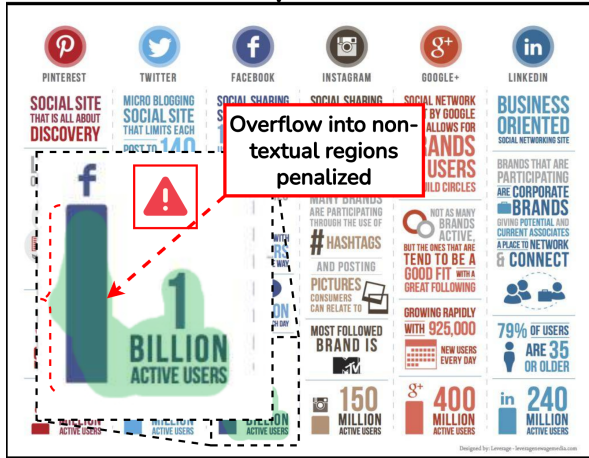
How many active users does Facebook have? Provide the answer with grounding



<e>1 BILLION ACTIVE USERS </e>[GROUND]



M3 Grounder (While training)



The over flow here in green color is penalised more due to bleed loss

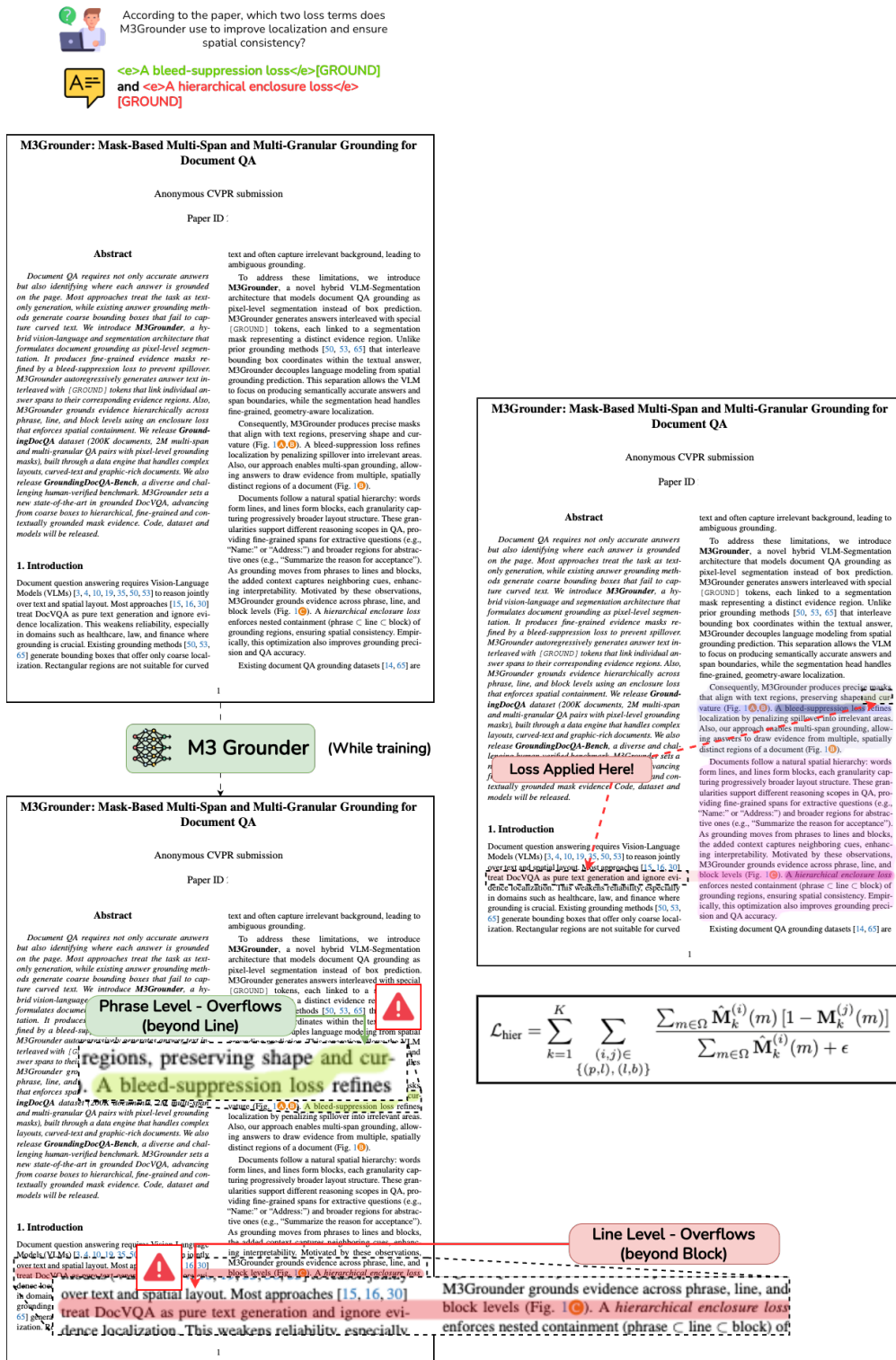
$$\mathcal{L}_{\text{bleed}} = \sum_{i \in \mathcal{G}} \sum_{k=1}^K \frac{\sum_{j \in \Omega} \hat{M}_k^{(i)}(j) [1 - M_{\text{ref}}(j)]}{\sum_{j \in \Omega} \hat{M}_k^{(i)}(j) + \epsilon}$$

Figure 2. Bleed Loss

“Name”, repeated section headers). OCR-span matching cannot determine which instance corresponds to the correct semantic answer. As a result, grounding becomes ambiguous and often incorrect.(2) Further, OCR pipelines flatten the document into a single unstructured text sequence, discarding layout hierarchy and spatial relationships.

Hence, grounding diversity degrades significantly. To address these limitations, a grounding dataset must preserve hierarchical layout, fine grained spatial structure, and visual geometry of text elements. Many real world documents contain curved headlines, skewed paragraphs, multi-column layouts, tables with merged cells, and visually dense charts. These structures cannot be faithfully represented by OCR based pipelines, which operate purely at the token level and ignore geometric attributes such as curvature, rotation, and shape.

Moreover, existing document grounding QA datasets [10, 48] do not provide segmentation masks, offering only bounding boxes. These boxes are insufficient for irregular text, curved text, or chart elements, where a rectangular region captures large amounts of background and fails to localize the true answer region. Reliable grounding requires pixel-level masks that tightly follow the shape of text lines, phrases, and visual elements.



Phrase Level - Overflows (beyond Line)

A bleed-suppression loss refines localization by penalizing spillover into irrelevant areas.

Line Level - Overflows (beyond Block)

M3Grounder grounds evidence across phrase, line, and block levels (Fig. 1(c)). A hierarchical enclosure loss enforces nested containment (phrase \subset line \subset block) of grounding regions, ensuring spatial consistency.

$$\mathcal{L}_{hier} = \sum_{k=1}^K \sum_{(i,j) \in \{(p,l), (l,b)\}} \frac{\sum_{m \in \Omega} \hat{M}_k^{(i)}(m) [1 - M_k^{(j)}(m)]}{\sum_{m \in \Omega} \hat{M}_k^{(i)}(m) + \epsilon}$$

Figure 3. Hierarchical Loss

- To handle this, we design three complementary data generation pipelines: ① a pipeline for layout aware documents, ② a dedicated pipeline for curved and skewed text documents, and ③ a rendering-driven pipeline for charts.

Dataset Statistics: Our dataset contains 52% single-span QAs (16% charts, 28% text-rich documents, 8% curved) and 48% multi-span QAs (17% charts, 24% text-rich documents, 7% curved), ensuring diverse QA pairs. Our chart corpus is primarily composed of Bar Charts (18.62%), Line Charts (17.19%), and Scatter Plots (15.20%), followed by Pie Charts (12.28%), Box Plots (8.82%), Area Charts (7.43%), Heatmaps (5.58%), Radar Charts (4.52%), and Donut Charts (3.87%), while the remaining chart types collectively account for 6.49% of the dataset.

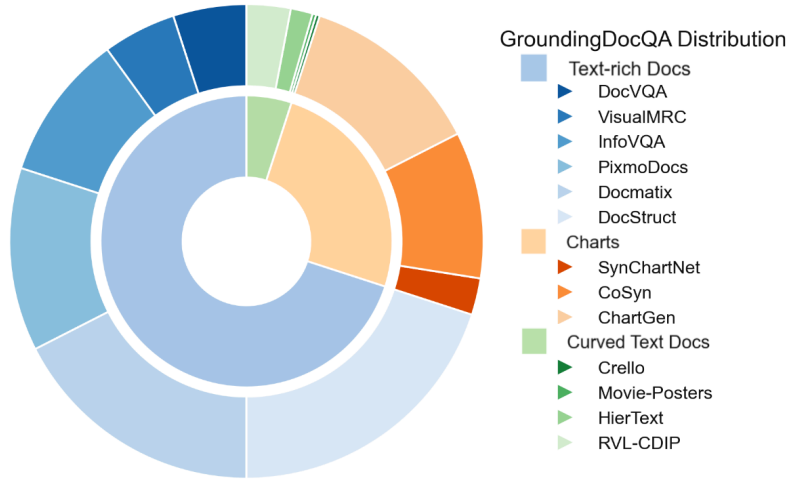


Figure 4. GroundingDocQA Dataset Distribution

QA Taxonomies: We design our GroundingDocQA dataset to capture a broad spectrum of question patterns rather than restricting it to simple text-extraction tasks. The final collection includes grounded QA pairs spanning multiple taxonomies such as extractive queries, structural questions, procedural instructions, validation checks, comparisons, and summarisation oriented reasoning. As shown in Fig. 5, the dataset is organised into four categories extractive, structural, procedural, and reasoning, each further divided into fine-grained subtypes including localisation, numerical queries, NER-style questions, comparative analysis, counterfactual reasoning, and abstractive summarisation.

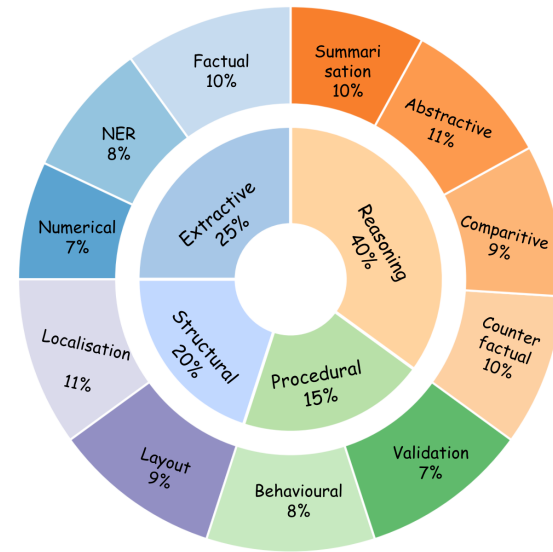


Figure 5. Distribution of QA taxonomies used during dataset generation. The inner ring shows the four categories (Extractive, Structural, Procedural, and Reasoning), while the outer ring shows the further subtypes.

4.1. Layout-Aware Documents

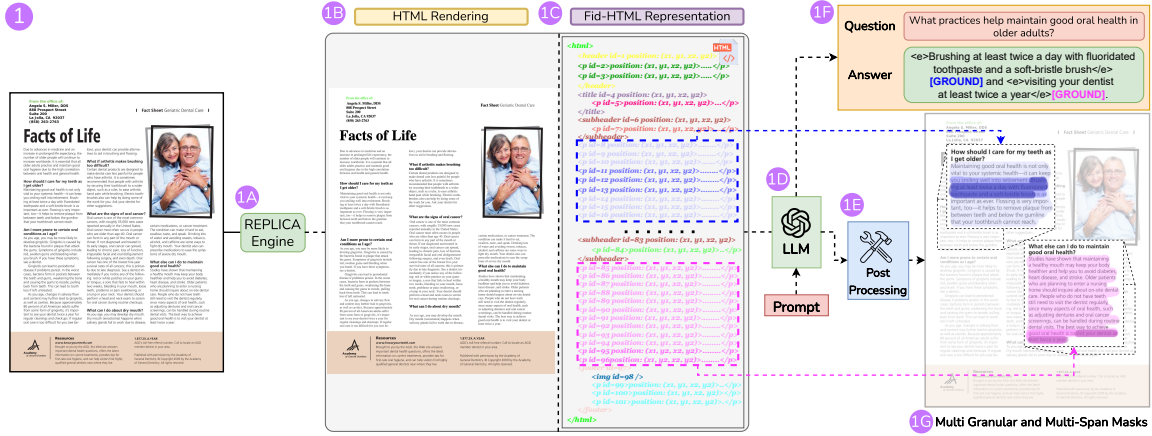


Figure 6. Grounded QA generation pipeline for Layout-Aware Documents

Why REPLICa for Text-Rich Documents. For text rich documents, we rely on REPLICa [2] engine for grounded QA generation because it preserves both the visual fidelity and the hierarchical structure of each page. REPLICa’s [2] *Fid-HTML* (1C) provides stable line level bounding boxes, their corresponding block-level regions, and an explicit reading order that matches the true layout of the document. Figs. 7 to 9 Shows that REPLICa [2] captures layout and geometry of document, while OCR or Markdown based representations often collapses structure, merges unrelated regions.

Overview of Fid-HTML (1C): Fid-HTML is a high-fidelity, semantically enriched HTML representation produced by the REPLICa [2] engine. It is designed to capture both the semantic structure and the visual presentation of a document, enabling downstream tasks that require faithful reconstruction as well as machine-readable semantics.

Fid-HTML encodes document content along multiple complementary dimensions:

1. *Textual content:* complete text is extracted and preserved at both word and block level.
2. *Hierarchical structure:* nested `<div>` elements represent document hierarchy such as sections, blocks, and lines.
3. *Semantic tags:* structural HTML tags such as ``, `<table>`, `<tr>`, `<td>` and semantic class names provide rich semantic grounding.
4. *Positional information:* global layout is preserved via absolute coordinates, while local alignment is captured through relative positioning; textual segments are wrapped in `<p>` tags inside higher-level containers.
5. *Styling metadata:* font size, color, and text attributes (bold, italics, underline, strikethrough) are maintained as inline CSS for high-fidelity style preservation.
6. *Figures and backgrounds:* images, figures, and background regions are incorporated into the HTML; alt captions for figures is automatically generated by VLMs to enhance accessibility.

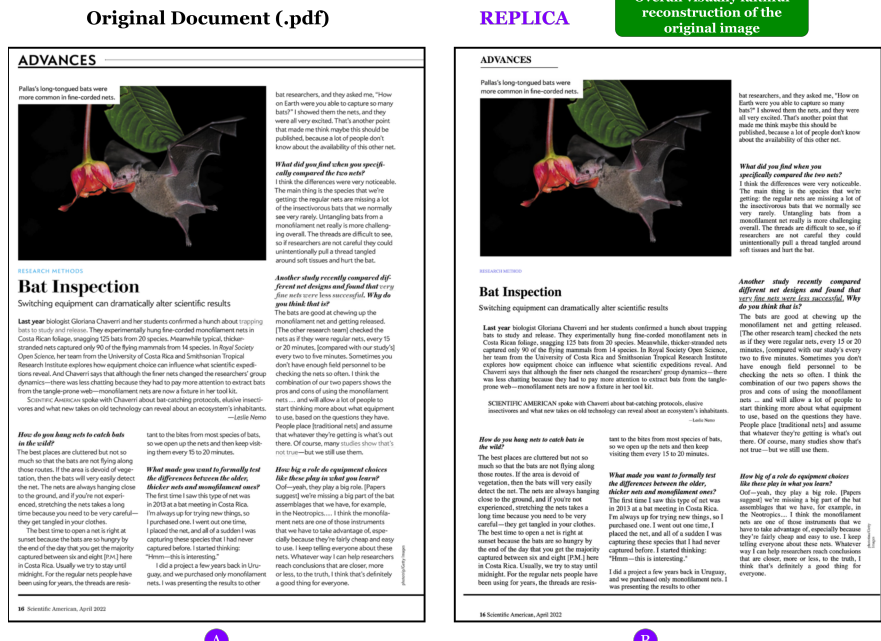


Figure 7. Reference Fid-HTML representation generated using the REPLICA engine (Sample 1).

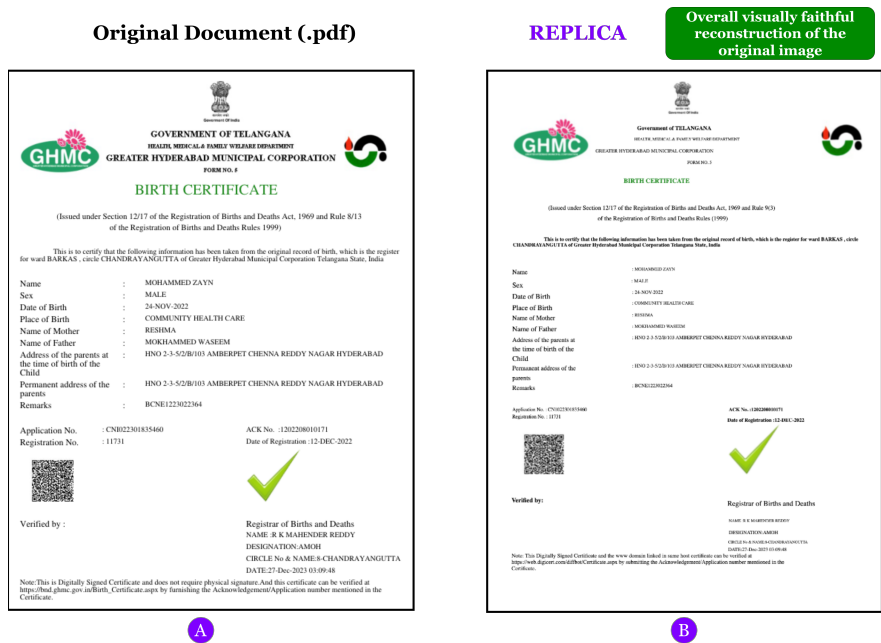


Figure 8. Reference Fid-HTML representation generated using the REPLICA engine (Sample 2).

Original Document (.pdf)

REPLICA

Overall visually faithful
reconstruction of the
original image

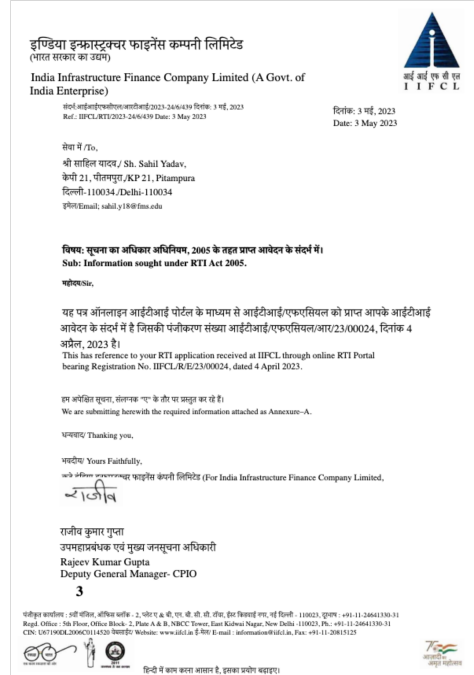
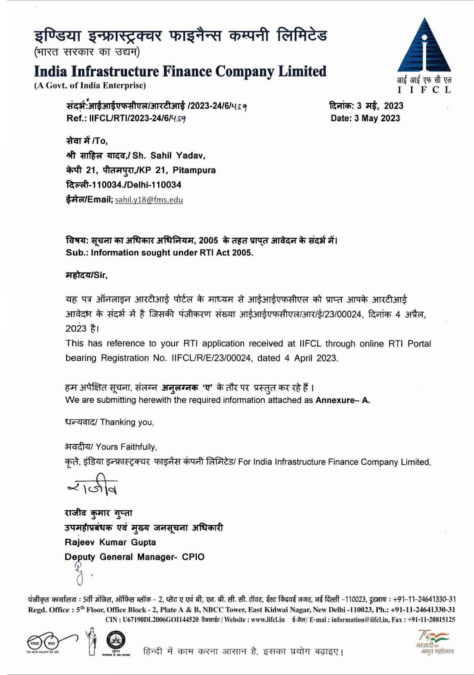


Figure 9. Reference Fid-HTML representation generated using the REPLICA engine (Sample 3).

QA Prompt Specification for Layout-Aware Documents ^{ID} Given the document's Fid-HTML, we design a QA prompt that guides the LLM [1] to produce diverse, spatially grounded QA pairs. The prompt covers diverse taxonomies including extractive, structural, procedural and reasoning-oriented questions.

Input & Task

Input: One Element-ID HTML that maps visible elements (titles, paragraphs, headers, tables, lists, captions, footnotes, etc.) to unique element IDs together with their corresponding text and bounding boxes.

Task:

- Generate **3–7 diverse, non-overlapping** QA pairs grounded in the provided HTML.
- Ensure **at least one QA per requested taxonomy** (extractive, abstractive, procedural, reasoning).
- Produce a mixture of short factual questions and longer abstractive/reasoning questions.
- Use only visible `<body>` content; ignore `<head>`, `<style>`, `<script>` or any hidden metadata.
- Do not correct OCR, paraphrase evidence phrases, or invent content not present in the HTML.
- No hallucination every answer must be evidence based.

Evidence Rules & Answer Alignment

Element usage

- Use **only** the element IDs present in the Element-ID HTML. Never invent or modify IDs.
- For every QA include the element's own ID in the evidence field when possible.
- For multi-span answers include all relevant element IDs.

Minimal evidence rule

- Prefer the smallest set of element IDs that fully supports the answer.
- Add parent/heading IDs only when required for disambiguation or context.
- For numeric/tabular answers, include the specific cell ID plus the label IDs that disambiguate the number.

Answer alignment

- Each QA must include `phrases`: minimal verbatim substrings (3–4 words) copied exactly from the referenced element(s).
- If an answer requires multiple spans, include multiple `phrases` in the listed order.
- If a phrase match is ambiguous (the same substring occurs multiple times on the same line and cannot be uniquely resolved), discard that QA rather than guessing.
- Never paraphrase or invent text when constructing `phrases`.

Required Fields & Forbidden Questions**Required QA object fields (per item)**

- `qa_number`: integer (sequential starting at 1)
- `category`: human-readable taxonomy (e.g., factual, ner, numerical, abstractive)
- `question`: standalone string
- `answer`: string (or array/object when appropriate)
- `answer_html_id`: array of element IDs used as evidence
- `phrases`: array of minimal verbatim substrings (3–4 words) copied exactly from the referenced element(s)

Final checks (validator)

- Output must be a valid UTF-8 JSON array containing 3–7 QA objects. No markdown, no commentary, no trailing commas.
- At least one QA per requested taxonomy; no hallucinations; all `phrases` verbatim.
- If phrase-to-box mapping is ambiguous the QA must be discarded.

Forbidden questions

- Do not ask meta-questions about element IDs (e.g., “What element IDs are present?”).
- Do not ask about raw HTML attributes, tags or source code (e.g., “What is the image src?”).
- Do not generate questions that require internal document-structure inspection rather than human-readable content.
- Avoid empty placeholders and meaningless comparisons.

Example (single QA object)

```
[
  {
    "qa_number": 1,
    "category": "factual",
    "question": "What is the GST rate for the extra bed?",
    "answer": "18% on Rs. 8000",
    "answer_html_id": ["8.30", "8.32"],
    "phrases": ["18% on Rs. 8000"]
  }
]
```

Extractive QA Taxonomy

Factual. Factual questions require retrieving a specific piece of information explicitly stated in the document. These include short answers such as names, dates, durations, titles, or direct statements without any reasoning.

Examples:

- What is Janes studying? → “Literature.”

NER (Named Entity Recognition). NER-based questions focus on identifying proper nouns present in the document, such as people, organizations, locations, dates, or product names. The answer must match the exact entity span without paraphrasing.

Examples:

- Which company manufactured the device? → “ZenLabs Pvt. Ltd.”

Numerical. Numerical questions require extracting values such as percentages, counts, prices, quantities, units, or any numeric expression that appears verbatim in the document. No arithmetic or reasoning is needed.

Examples:

- What percentage of tax is applied? → “18%”

Structural QA Taxonomy

Layout & Structure

Questions about the document’s layout, logical grouping, and spatial relationships. This category includes reasoning over table structure, list membership, header–body relations, and identifying which block or section contains a given piece of information.

Examples:

- “Which row lists the product price?” → “Row 4 (Price column)”

Localisation

Questions that require locating content in page coordinate space or mapping answers directly to visual regions (blocks, lines, words, or polygons). Answers should reference element IDs and/or explicit bounding boxes and may include pixel coordinates when needed. This taxonomy evaluates precise grounding and spatial alignment rather than semantic retrieval.

Examples:

- “Which element contains the invoice total? Return its element ID.” → “7.2”

Procedural QA Taxonomy

Validation

Binary (Yes / No, True / False) questions that check presence, compliance, or correctness of required information. These questions confirm whether a condition is satisfied and must cite the specific element(s) that justify the decision. Validation often covers regulatory checks, required-field presence, signatures, and checkbox/status verification.

Examples:

- “Is the application signed?” → “Yes”

Behavioral

Questions about required actions, step-by-step procedures, or compliance workflows. Answers should list

actionable steps or describe procedural consequences, and must reference the element IDs that correspond to each step or requirement. Procedural questions evaluate whether the document contains the instructions, forms, or fields needed to complete a process.

Examples:

- “How to renew the license?” → “1. Fill renewal form 2. Attach photo ID 3. Pay fee”

Reasoning QA Taxonomy

Abstractive & Summarisation

Higher-level questions that require synthesising content from multiple elements into a concise, human-readable answer. Summarisation asks for a short synthesis of a policy/paragraph; abstractive items may require combining or rephrasing multiple evidence spans (but the `phrases` field must still contain verbatim substrings used as anchors).

Examples:

- “Give a short summary of the delivery exceptions.” → “Deliveries delayed for severe weather and customs holds; contact support for re-scheduling.”

Comparative & Calculation

Questions that require comparing values across elements or performing small calculations using explicit numeric spans. Always cite the specific element IDs used for the comparison or calculation and include the verbatim numeric phrases.

Examples:

- “Which plan is cheaper per month, Plan A or Plan B? Show numbers.” → “Plan A :Rs. 199/month vs Plan B :Rs. 249/month; Plan A is cheaper.”

Counterfactual

Hypothetical “what-if” questions that ask the model to reason about alternative scenarios or consequences when a document condition changes. Answers should be concise, logically derived, and cite the clause(s) or elements that justify the inference when possible.

Examples:

- “If the delivery address is changed after dispatch, what is the likely outcome?” → “Delivery may be rerouted at cost to the sender or returned to sender if already shipped.”

Postprocessing for Layout-Aware Documents 1E

For each document, the model outputs a QA pair together with its corresponding line-level bounding boxes, extracted from the REPLICA [2] hierarchy. From this line-level information, we obtain both block-level and phrase-level bounding boxes through a two-stage postprocessing procedure described below.

Block-Level Bounding Boxes: REPLICA [2] provides block-level bounding boxes directly, together with a hierarchical mapping from each line to its parent block. For every predicted line, we use this hierarchy to identify its corresponding block in the Fid-HTML.

Phrase-level Bounding Boxes: The model outputs the key phrase along with HTML span IDs that correspond to the matched text along with the QA pair. We retrieve the HTML content of these IDs to obtain the exact words that appear in the predicted line. We combine two sources of word-level geometry: docTR [25] which provides reliable text content and word boxes, and Hi-SAM [45], which offers more accurate and visually aligned word bounding boxes. By aligning docTR’s OCR words with Hi-SAM’s precise boxes, we obtain accurate bounding boxes of words. To obtain the phrase bounding box, we match the generated key phrase against the OCR words present in the predicted line. When a word appears multiple times within the same line and leads to ambiguous matches, the corresponding QA pair is discarded to avoid incorrect grounding. For valid cases, we gather the matched words and merge their adjacent Hi-SAM derived word boxes to obtain a unified phrase-level bounding region. This process ensures that phrase-level regions are

grounded precisely to the accurate answer text.

Segmentation masks: With the finalized phrase, line, and block-level bounding boxes, we use Hi-SAM to obtain the corresponding segmentation masks. Hi-SAM provides a tight polygonal mask of the bounding box region that adheres to the true shape of the text, ensuring accurate segmentation even for small fonts, irregular spacing, or visually complex regions.

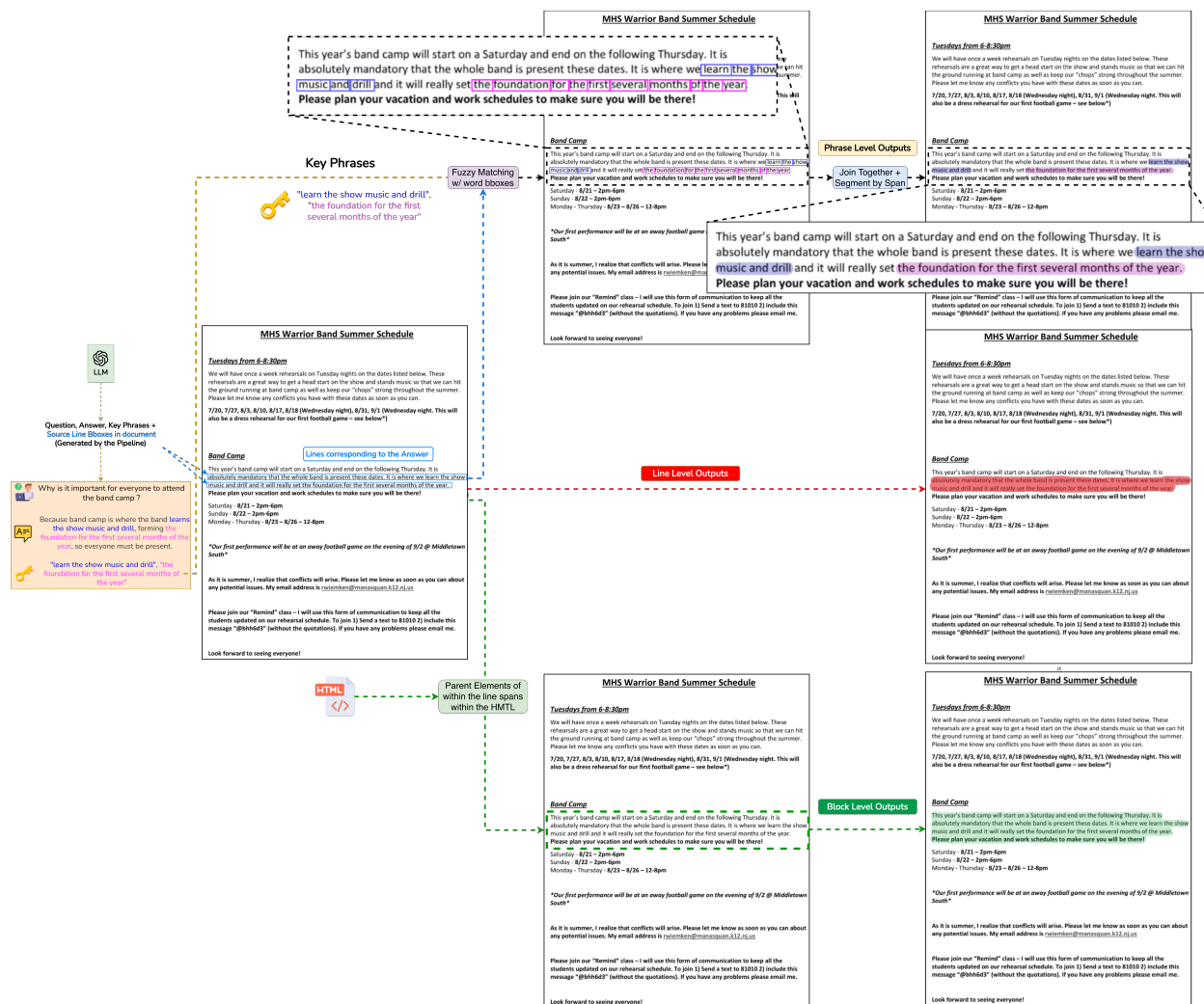


Figure 10. Post Processing for Layout Aware Documents

4.2. Curved Text Documents



Figure 11. Grounded QA generation pipeline for Curved-Text Documents

QA Prompt Specification for Curved Text Documents We design a prompt for curved text documents where QA generation is driven solely on the highlighted text regions in the document. Each prompt may contain one or two highlighted irregular regions, and questions must be formed strictly from the content visible within these highlighted regions. This setup ensures that QA pairs remain grounded in the actual curved, skew and irregular texts.

Prompt for Curved Text Documents

Input:

A document image with a *highlighted curved or irregular text region*. The highlighted region may contain warped, slanted, circular, semi-circular,skewed or otherwise non-linear text.

Task:

Generate question–answer pairs strictly from the highlighted curved text. The model must ignore all unhighlighted parts of the document.

Instructions:

- Extract the exact text from the highlighted curved region.
- Generate QA pairs only if the highlighted text contains meaningful, interpretable content.
- Do **not** use or reference any text outside the highlighted region.
- If the highlighted text is unreadable, too short, or cannot support a valid question, return **zero** QA pairs.
- All generated questions must be answerable directly from the curved text itself.

Output Format:

```
{
  "qa_pairs": [
    {
      "question": "What milestone is being celebrated at the annual fest?",
      "answer": "5th Year Celebrations",
    }
  ]
}
```

If no valid QA pairs can be generated:

```
{
  "qa_pairs": []
}
```

4.3. Charts

Rendering-Based Element Extraction: ^{3B} To obtain precise geometric information, we process charts at render time. We execute each chart’s script using libraries such as Matplotlib[12], Seaborn[41], [38], and Plotly[28], and intercept the rendering pipeline to capture low-level drawing operations. For examples, Matplotlib-based [12] scripts, we access the internal renderer via:

```
renderer = fig.canvas.get_renderer()
```

Using this, we query all visible artists on the chart such as Axes, Patch (bars, pie wedges, areas), Line2D (lines, markers), Text (titles, labels, ticks), and Collection (scatter/heatmap elements). Each artist exposes a `get_window_extent(renderer)` method, which returns the exact bounding box used to draw that element on the final image.

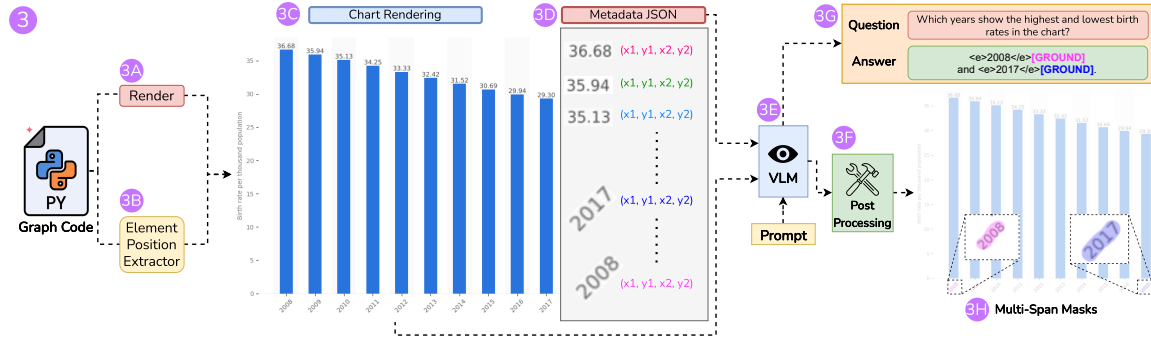


Figure 12. Grounded QA generation pipeline for Charts

All extracted elements, together with their bounding boxes, textual content, and element types, are stored in a metadata JSON [8] file ^{3D}. This JSON serves as a complete specification of the chart: each element is assigned a unique ID, and its geometric and semantic values are recorded. The rendered chart image and the corresponding JSON are later passed to the VLM for QA generation.

Example: Input Metadata JSON

```
{
  "globals": {
    "axes": [
      {
        "title": { "bbox": "1", "text": "U.S. views of Education" },
        "xlabel": { "bbox": "2", "text": "Year" },
        "ylabel": { "bbox": "3", "text": "Percentage" },
        "xticks": [
          { "text": "2005", "bbox": "4" },
          { "text": "2015", "bbox": "9" }
        ],
        "legend": {
          "frame": [763,79,893,146],
          "labels": [
            { "bbox": "10", "text": "Dissatisfied" },
            { "bbox": "11", "text": "Satisfied" }
          ]
        }
      ]
    ],
    "items": [
      {
```

```

    "type": "bar",
    "marker_bbox": "12",
    "x_tick_text": "2005",
    "x_tick_bbox": "13",
    "y_value": 35.0,
    "y_tick_bbox": "14",
    "extra": {
      "orientation": "v", "stack_group": null,
      "data": { "height": 35.0, "value": 35.0, "width": 0.40, "x": 2004.6, "y": 0.0 }
    }
  },
  {
    "type": "bar",
    "marker_bbox": "27",
    "x_tick_text": "2015",
    "x_tick_bbox": "28",
    "y_value": 34.0,
    "y_tick_bbox": "29",
    "extra": {
      "orientation": "v", "stack_group": null,
      "data": { "height": 34.0, "value": 34.0, "width": 0.40, "x": 2014.6, "y": 0.0 }
    }
  },
  {
    "type": "bar",
    "marker_bbox": "30",
    "x_tick_text": "2005",
    "x_tick_bbox": "31",
    "y_value": 65.0,
    "y_tick_bbox": "32",
    "extra": {
      "orientation": "v", "stack_group": null,
      "data": { "height": 65.0, "value": 65.0, "width": 0.40, "x": 2005.0, "y": 0.0 }
    }
  },
  .
  .
  .

```

QA Prompt Specification for Charts We design a structured chart-prompt format that captures all visual and semantic elements of a plot, including axes, tick labels, legends, and data marks. This metadata representation ensures that every graphical component is explicitly annotated. Such a unified prompt format allows models to interpret over charts consistently across diverse chart types.

Chart QA Prompt — Input & Task Specification

Inputs

- One chart image (Bar, Line, Pie, Donut, Scatter etc.).
- One Element-ID JSON mapping all visible chart components (titles, axes, ticks, legends, bars, lines, points,

slices) to unique string IDs (e.g., “1”, “2”, “37”).

- The Element-ID JSON is the **only** source of truth for evidence IDs.

Task

- Generate **upto 10** diverse, non-overlapping QA pairs grounded in the chart.
- Cover at least one QA from each category:
 1. Comparison
 2. Ranking
 3. Top-k list
 4. Aggregation & Arithmetic
 5. Threshold check
 6. Ratio
 7. Yes/No
 8. Counting
 9. Color-based
 10. Factual
 11. Summarisation
 12. Chart type identification
- Questions must be answerable using the visible chart content.
- Ensure a mix of short and medium-length answers (some with brief explanation).
- Avoid template-like repetition; diversify phrasing and chart elements.

Valid Chart Types Area Chart, Bar Chart, Box Plot, Bubble Chart, Candlestick Chart, Donut Chart, Funnel Chart, Heatmap, Histogram, Line Chart, Pie Chart, Radar Chart, Ring Chart, Rose Chart, Scatter Plot, Stem Plot, Tornado Chart, Treemap, Violin Plot.

Chart QA Prompt – Evidence & Grounding Rules

Element-ID Usage

- Use only IDs present in the Element-ID JSON; never fabricate IDs.
- IDs are strings (e.g., “24” not 24).
- For every QA, include the **minimal sufficient** evidence IDs.

Universal Mark Evidence Rule For every referenced chart element:

1. Always include the mark’s own bbox ID (e.g., marker_bbox, point_bbox, wedge_bbox, rect_bbox, cell_bbox, line_bbox, area_bbox, candle_bbox).
2. Include value reference IDs when present (value_label_bbox preferred; else y_tick_bbox).
3. Include category/position reference IDs when relevant (x_tick_bbox for Cartesian charts; slice/legend label for radial charts).

When Questions Involve Multiple Marks

- Provide a complete evidence set for each mark involved (comparisons, rankings, ratios, aggregations).

Chart QA Prompt — Output Format & Final Constraints

Output Format

- Return **one** valid UTF-8 JSON array with upto **10 objects**.
- Each QA object must contain:

```
{
  "qa_number": int (1..10),
  "category": string,
```

```

"question": string,
"answer": string/number/array,
"ids": [ "<element_id>", ... ]
}

```

Category Values: comparison, ranking, top_k, aggregation, threshold, ratio, yes_no, counting, color_based, factual.

Diversity Requirements

- No duplicate questions.
- Spread references across different marks/years/series.
- Mix concise and explanatory answers.

Final Checks

- Upto 10 QA objects.
- IDs are valid and appear in the JSON.
- Evidence strictly reflects the minimal required set.
- No extra commentary, no markdown fences.
- Response ends immediately after the closing bracket] .

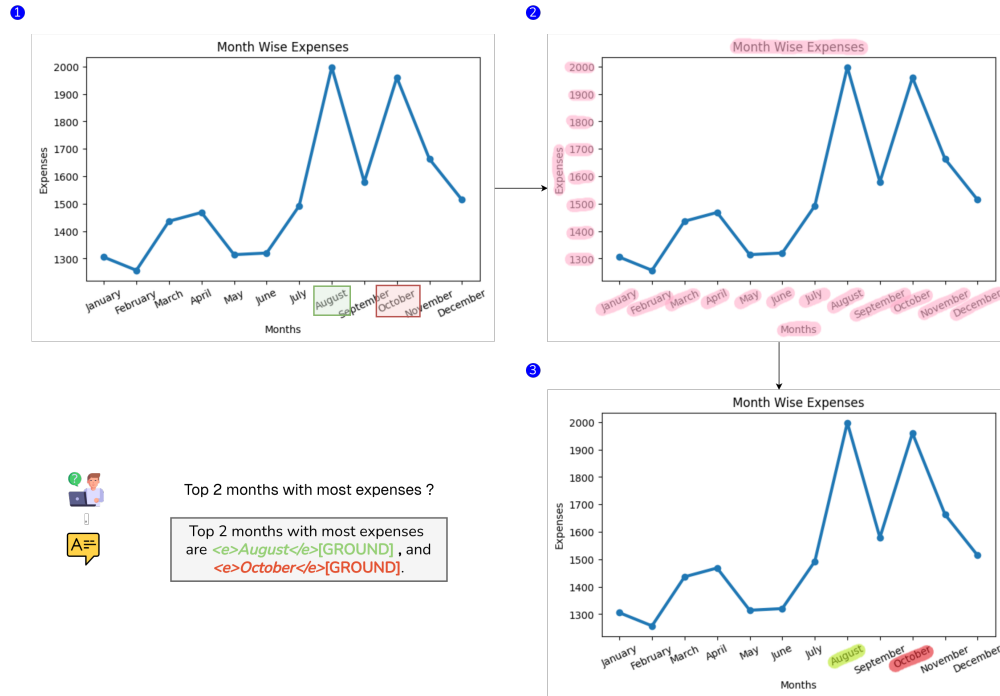


Figure 13. ① We obtain the bounding boxes of the answer evidence, such as *August* and *October*. ② We extract all valid text regions from the chart. ③ For each detected word from ②, we check its intersection with the bounding boxes from ①. Any word that lies completely within an evidence bounding box is considered a valid evidence text region.

Postprocessing of Chart Elements ③F Once we obtain the bounding boxes of all text elements from the rendering stage, we generate segmentation masks using Hi-SAM. Each bounding box is passed to Hi-SAM, which produces a precise mask for the corresponding chart element. This provides pixel-level grounding for all chart-based QA pairs, ensuring that each referenced text element has an accurate segmentation mask (see Fig. 13).

4.4. Data Verification

We apply a two-stage verification process to ensure that all grounded QA pairs in our dataset are both textually correct and layout-consistent. In the first stage, we check semantic alignment between the predicted answer and the OCR text from its grounded regions. For each answer span, we compute the sentence embedding similarity between the answer text and the OCR text extracted from the corresponding bounding boxes. QA pairs whose similarity falls below a fixed threshold are removed, as they indicate unreliable text-region alignment.

In the second stage, we verify answer completeness and grounding consistency using an LLM-as-a-Judge [9, 47]. Each QA pair, along with its key phrases, the bounding boxes of the answer, and the full document HTML as context are given as input to the LLM. Because we explicitly know which phrase maps to which region, the LLM evaluates whether (i) all essential parts of the answer are represented in the phrases, (ii) each phrase is correctly grounded in the OCR content of its assigned bounding boxes, and (iii) no part of the answer contradicts the surrounding HTML content. QA pairs are discarded if any phrase is missing, partially covered, incorrectly mapped, or inconsistent with the extracted text.

In a similar manner, we also verify chart-based QA pairs by checking answer correctness against the structured chart-metadata JSON using a VLM. For curved-text documents, we use a VLM, which receives the image and QA pair and determines whether the answer aligns with the text in the image. Together, these stages ensure that retained QA pairs are semantically valid, fully covered by their supporting text regions, and faithfully grounded in the document layout.

Verification Examples (Stage 1 & Stage 2)

```
{
  {
    "id": "1",
    "question": "What are all the documents are required to complete a child's school enrolment?",
    "answer": "You just need a birth certificate and the latest school report", ✗
    "answer_html_id": ["4.21", "4.22", "4.23", "4.24"],
    "phrases": [ "birth certificate", "latest school report" ]
  },
  {
    "id": "2",
    "question": "What are all the documents are required to complete a child's school enrolment?",
    "answer": "You need to submit birth certificate and proof of address are required.", ✗
    "answer_html_id": ["4.21", "4.22"],
    "phrases": [ "Birth certificate", "proof of address" ]
  },
  {
    "id": "3",
    "question": "What are all the documents are required to complete a child's school enrolment?",
    "answer": "You have to provide birth certificate and proof of address, and immunisation.", ✗
    "answer_html_id": ["4.21", "4.22", "7.02"],
    "phrases": [ "Birth certificate", "proof of address", "immunisation record" ]
  },
  {
    "id": "4",
    "question": "What are all the documents are required to complete a child's school enrolment?",
    "answer": "We need Birth certificate, proof of address, immunisation record and a passport copy for international students.", ✓
    "answer_html_id": ["4.21", "4.22", "4.23", "4.24"],
    "phrases": [ "Birth certificate", "proof of address", "immunisation record", "passport copy for" ]
  }
}
```

international students”
] }

ID	Stage 1	Stage 2	Reason
1	✗	–	Wrong answer content → low similarity.
2	✓	✗	Missing mandatory phrases → incomplete answer.
3	✓	✗	Incorrect bbox → phrase–region mismatch.
4	✓	✓	All phrases and bboxes correctly grounded.

5. Baselines Implementation

Different VLM families vary in how reliably they follow structured-output instructions, often producing the same information in different formats. To ensure fair evaluation across all baselines, we adopt a *model family aware* prompting strategy.

5.1. Prompts

This family-aware prompting strategy ensures that each model is evaluated under conditions that match its strengths, resulting in fairer comparison and more robust grounding across diverse VLM families.

OpenAI Models

GPT-4o[13] and GPT-5[26]

You are a vision-language model performing document question answering with spatial grounding.

- Given the document image <IMAGE> and the question <QUESTION>, identify all answer span(s) that appear in the image.
- For each span, extract: The exact evidence as it appears in the document, The bounding box in pixel coordinates: [x_min, y_min, x_max, y_max].
- Return your output as a valid JSON object:

```
{
  "answer" : "<ANSWER>" ,
  "answer_spans": [
    {
      "text": "<exact evidence from the document>",
      "bbox": [x_min, y_min, x_max, y_max]
    } ,
    {
      "text": "<exact evidence from the document>",
      "bbox": [x_min, y_min, x_max, y_max]
    }
  ]
}
```

Rules:

- Output valid JSON only (no markdown, commentary, or explanations).
- Every "bbox" must contain four numeric pixel values.
- Do not fabricate or paraphrase text; only use evidence present in the document.
- If no answer is present in the document, return: { "answer_spans": [] }.

Google Models

The prompt for this model family is constructed with reference to its official implementation [11].

Gemini 2.5 Pro [6] and Gemma 3 [35]

You are a vision-language model performing document question answering with spatial grounding.

Given the document <IMAGE> and the <QUESTION> below:

- Identify all span(s) in the image that answer the question.
- Return your answer as a JSON object with the following structure:

```
{
  "answer" : "<ANSWER>" ,
  "answer_spans": [
    {
      "text": "<exact evidence from the document>",
      "box_2d": [y_min, x_min, y_max, x_max]
    } ,
    {
      "text": "<exact evidence from the document>",
      "box_2d": [y_min, x_min, y_max, x_max]
    }
  ]
}
```

Rules:

- Return valid JSON only (no markdown, code fences, or explanations).
- Each "bbox" must contain four numerical values in pixel coordinates only.
- If multiple spans exist, include each as a separate object in the "answer_spans" list.
- If there is no visible answer, return:

```
{ "answer_spans": [] }
```

OpenGVLab Models

InternVL3.5 8B [40]

You are a vision-language model performing document question-answering with spatial grounding.

Given the document <IMAGE> and the <QUESTION>:

Your tasks:

- Identify all span(s) in the document that answer the question.
- For each span, extract:
 - The exact evidence as present in the document.
 - Its bounding box in the format: [x_min, y_min, x_max, y_max].
- Produce the answer in an **interleaved style**: mix your normal explanation with the extracted spans and their bounding boxes inserted at the appropriate points.
- If the answer contains multiple spans, present them sequentially in the same interleaved format.
- Do NOT invent any unwanted text; the exact evidence as present in the document. At the end of your response, clearly list all extracted answer spans in a structured form:

Span k: "<text>" [x_min, y_min, x_max, y_max]

Qwen Models

Qwen3-VL 8B [36]

You are a vision-language model performing document question-answering with spatial grounding.

Given the document <IMAGE> and the <QUESTION>:

Your tasks:

- Identify all span(s) in the document that answer the question.
- For each span, extract:
 - The exact evidence as present in the document.
 - Its bounding box in the format: [x_min, y_min, x_max, y_max].
- Produce the answer in an **interleaved style**: mix your normal explanation with the extracted spans and their bounding boxes inserted at the appropriate points.
- If the answer contains multiple spans, present them sequentially in the same interleaved format.
- Do NOT invent any unwanted text; the exact evidence as present in the document. At the end of your response, clearly list all extracted answer spans in a structured form:

Span k: "<text>" [x_min, y_min, x_max, y_max]

Microsoft Models

Kosmos 2.5 Chat [21]

You are a model that answers questions about a document image with grounding.

Given the <IMAGE> and <QUESTION>:

Your tasks:

- Find all text spans in the image that answer the question.
- For each span, provide:
 - The exact evidence as present in the document.
 - Its bounding box: [x_min, y_min, x_max, y_max].
- Write your answer in an **interleaved style**: normal explanation mixed with the spans and their bounding boxes inserted at the right points.
- Use only evidence text that is present in the image.
- At the end, output a clean list:

Span k: "<text>" [x_min, y_min, x_max, y_max]

Fine-Tuned Models (Ours)

InternVL3.5 (ft.) and Qwen3-VL (ft.)

You are a vision-language model performing document question-answering with spatial grounding.

Given the document <IMAGE> and the <QUESTION>:

Your tasks:

- Identify all answer span(s) in the document.
- For every span that appears in the document, output it in the following interleaved format:
 - <e>EVIDENCE_FROM_DOCUMENT</e><bbox>x_min, y_min, x_max, y_max</bbox>
- Return the answer in an interleaved format, mixing your reasoning text with grounding spans.
- If multiple spans are needed, list them sequentially using the same format:

...<e>span1</e><bbox>...</bbox>...<e>span2</e><bbox>...</bbox>...

- Use explanatory text in between if needed, but every piece of evidence taken from the document must be wrapped using <e>...</e> with its corresponding bounding box using <bbox>...</bbox>.
- Do not fabricate text. Only include spans that appear in the document.

M3GrounderVariants

M3Grounder-I and M3Grounder-Q

You are a vision-language model for document question answering with segmentation-based grounding.

Given the <IMAGE> and the <QUESTION>:

- Identify all answer spans in the document.
- After every span that appears in the document, append a [GROUND] token.
- Return the answer as normal explanatory text interleaved with [GROUND] tokens.
- For every span that appears in the document output it in the following interleaved format:

...<e>span1</e> [GROUND] ...<e>span2</e> [GROUND] ...

- Do not output bounding boxes, coordinates or any other metadata only the answer text with the [GROUND] markers placed immediately after each grounded span.

6. G-Eval for Answer Quality (AQ)

In our evaluation protocol for BoundingDocs (Test) and GroundingDocQA-Bench, we adopt **G-Eval** [19] as the metric for Answer Quality (AQ) instead of relying on traditional metrics like Exact Match (EM) [29] or ANLS [32]. As these metrics have limitations in evaluating abstractive answers common in document VQA.

Lexical metrics are sensitive, for example, a correct prediction of "properties assistant" receiving an EM score of 1.0, a similar answer like "Properties Assistant" (with a capital 'P') or "The properties assistant" would receive a score of 0.0. This score fails to capture the model's correct understanding.

For abstractive question answers evaluated with BLEU-4 [27], this problem is even more severe. As shown by the real examples shown in Table 1, models frequently provide correct answers that receive a BLEU-4 score of 0.0. For examples, this happens when the model's answer is a concise fact (e.g., "20") while the ground truth is a descriptive sentence ("The number of users decreased is 20").

G-Eval [19], by using an LLM as a judge, evaluates the **semantic equivalence** and **factual consistency** of the prediction. It correctly identifies the answers (see Tab. 1) as being semantically correct, aligning far more closely with human judgment. This provides a much fairer and more robust measure of a model's abilities.

For our implementation, we use **DeepSeek-R1- 70B** [9] as the LLM powering G-Eval. Our G-Eval implementation was configured for robustness and reproducibility. We set the passing threshold at 0.5, meaning any answer with a semantic similarity score of 0.5 or higher was considered correct. The final Answer Quality (AQ) score is computed as the average of all successfully evaluated QA pairs. The specific prompt template used to guide the model’s evaluation

Prompt used for G-Eval

You are an evaluator model. Follow the rubric below strictly to score the semantic equivalence between a Ground Truth and a Prediction.

Your tasks:

- **Semantic Focus:**
 - Evaluate only the textual content produced by the model. Ignore layout information such as bounding box coordinates, pixel positions, or any other visual metadata.
- **Meaning Preservation:**
 - Determine whether the predicted text expresses the same semantic information as the ground-truth reference.
- **Tolerance:**
 - Minor variations in punctuation, casing, or formatting that do not change meaning should be accepted.
- **Error Sensitivity:**
 - Missing, substituted, or additional words that alter the meaning should be treated as errors.
- **Numeric Content:**
 - Numerical tokens should be evaluated as part of the textual content unless they clearly represent layout metadata (e.g., coordinates).

Question	Ground Truth	Predicted Answer	BLEU-4	EM	G-Eval
...what role did franny kromminga have...	properties assistant	Properties Assistant	0.0000	0.0	Correct
...which year showed a decrease in the number of users...	the number of users decreased in 2020 where the count was 1038000...	2020	0.0000	0.0	Correct
...what is the difference in percentage points between the two age groups...	...the difference is 12 percentage points as the share for 18-19 years old is 16%, and for 20-24 years is 28%.	28% - 16% = 12%	0.0000	0.0	Correct
...where will the event take place...	the event will take place at '930 carnegie st'.	930 carnegie st	0.0154	0.0	Correct
...what is the intended purpose of this poster...	...provide ' 1o tips for ', ' business ', ' online '...	...provide ' 10 tips for ', ' online ', ' business '.	0.2234	0.0	Correct
...what is the total value of the top three countries combined...	...total is 4960 billion dollars.	...total value ... is 4657.90 billion dollars...	0.0096	0.0	Incorrect

Table 1. Examples of Lexical Metric Failures: These examples show semantically correct model predictions that receive near-zero scores from traditional metrics (BLEU-4 / EM), demonstrating the need for a semantic metric like G-Eval (threshold ≥ 0.5).

7. More details on benchmarks and evaluations

We evaluate our models across four document grounding QA benchmarks. Below, we describe each benchmark and the evaluation followed for it.

7.1. BoundingDocs-Test

The **BoundingDocs** [10] test split contains 4,832 documents and 13,351 question-answer pairs. However, our evaluation protocol required two significant filtering steps to align the dataset. First, as our models are designed for single-page document processing, we filtered out all multi-page documents. Second, we observed Q/A pairs were in other than english so these were discarded. After applying these two filters, we then clean repeated documents and incomplete grounded QA pairs. From this filtered set, we obtain our final evaluation split Dataset Statistics (Grounding):

- Images: 1,000
- QA Pairs: 3,000

Evaluation Protocol: Our evaluation protocol for BoundingDocs [10] assesses two aspects: grounding performance and answer quality [17, 37, 47].

- Grounding Performance: This is measured using F1 grounding score $F1_g$. A prediction is counted as a True Positive if the Intersection over Union (IoU) between the predicted and ground-truth bounding box is ≥ 0.5 .
- Answer Quality (AQ): We use **G-Eval** [19] as a unified semantic scoring metric. This approach provides a balanced semantic-spatial evaluation that complements simple lexical or structural measures.

Model	Grounding					
	G_a		G_r		G_o	
	Acc	F1 _{all}	Acc	F1 _{all}	BLEU4	F1 _{all}
Gemini 1.5 Flash*[34]	32.7	0.7	59.0	1.1	13.6	1.7
GPT-4o mini*[13]	64.5	0.8	48.7	0.6	9.4	0.5
Gemini 1.5 Pro*[34]	77.7	9.4	62.0	5.8	13.4	6.7
Gemini 2.0 Flash*	74.2	5.2	63.5	3.1	15.2	2.9
Gemini 2.5 Flash*[7]	80.2	38.8	59.3	25.4	12.8	21.5
Gemini 2.5 Pro*[7]	61.8	30.1	47.8	17.2	13.6	14.7
GPT-4o*[13]	79.0	8.8	47.0	3.8	12.5	9.2
Qwen2-VL-7B*[39]	41.7	1.8	4.3	1.4	9.8	5.5
InternVL2-8B*[5]	51.2	2.4	26.2	0.1	10.2	1.1
InternVL2.5-8B*[4]	58.3	3.8	33.0	0.6	13.8	2.0
Qwen2.5-VL-7B*[3]	63.8	19.5	35.0	9.8	6.1	11.3
DOGR [48]	83.2	<u>76.3</u>	67.7	<u>54.8</u>	38.3	<u>59.4</u>
InternVL3.5 8B ft.[40]	79.8	29.9	65.8	32.0	27.1	27.7
Qwen3-VL 8B ft.[36]	81.6	38.5	<u>68.9</u>	33.7	29.6	34.4
M3Grounder-I (p)	<u>84.3</u>	75.4	66.3	53.6	34.8	56.6
M3Grounder-Q (p)	85.7	77.9	70.1	56.4	<u>35.3</u>	62.1

Table 2. **DOGR Grounding Split Results.** We evaluate on the official DOGR [48] *Grounding Split*, which includes the G_a , G_r , and G_o tasks (plain-text questions with grounded answers). Following the DOGR protocol, we report Exact Match for G_a/G_r , BLEU-4 for G_o , and F1_{all} for grounding. This table reproduces the original DOGR evaluation format to enable a fair, directly comparable assessment of our models. **Bold** denotes the highest score among all models, and Underline denotes the second highest.

7.2. DOGR-Bench

DOGR-Bench [48] is designed to evaluate the grounding and referring capabilities of VLMs on document images. It features tasks that require models to not only answer questions but also provide spatial grounding for their answers.

Dataset Statistics (Grounding):

- Images: 200
- QA Pairs: 1,100

Evaluation Protocol: The DOGR benchmark contains multiple task splits (GA, GR, GO, PA etc). For our evaluation, we use only the split, which includes the three tasks that the DOGR paper [48] designates for grounding capability: (i) G_a : Grounded Answer, (ii) G_r : Grounded Reasoning, and (iii) G_o : Grounded Open-ended Answer. These tasks contain plain-text questions and answers with bounding boxes.

In the main paper, we reported $F1_g$ and Answer Quality (AQ) using G-Eval [19] for DOGR. In this section, we extend those results by following the exact DOGR evaluation protocol using Exact Match for G_a/G_r and BLEU4 for G_o and

by reproducing the same table format as the original DOGR benchmark. This ensures a fair and directly comparable evaluation using the same split and metrics defined in the benchmark. Our evaluation is bifurcated into two components as specified by DOGR: Text Answer Accuracy and Grounding Performance.

Text Answer Accuracy:

- **Exact Match (Acc):** For the short answer tasks, G_a (Grounded Answer) and G_r (Grounded Reasoning), we report the Exact Match (EM) accuracy.
- **BLEU4:** For the long answer G_o (Grounded Open-ended Answer) task, we report the BLEU4 score, which is more suitable for evaluating.

Grounding Performance ($F1_{all}$): We use $F1_{all}$ score to measure grounding performance, which requires a simultaneous match in both text content and spatial location. Our implementation precisely follows the DOGR [48] definition.

A predicted grounded span is counted as a True Positive (TP) if and only if it meets two conditions:

- **Spatial Match:** The Intersection over Union (IoU) between the predicted bounding box and a ground truth bounding box is ≥ 0.5 .
- **Text Match:** The cleaned text within the predicted span (using the same normalization process as above) exactly matches ground truth text.

This evaluation ensures that our results for M3Grounder and our fine tuned models are fairly comparable to the original DOGR baseline.

7.3. MMDocBench

MMDocBench [50] is a multi-task benchmark covering 15 tasks and 48 sub-tasks across diverse document types (receipts, reports, tables, charts, research papers), each annotated with supporting bounding regions.

Dataset Statistics:

- Images: 2,400
- QA Pairs: 4,338

Evaluation Protocol: We follow the official evaluation with no modifications:

- **Text Accuracy:** Exact Match (EM) and word-level F1 ($F1_{txt}$).
- **Grounding:** Intersection-over-Union (IoU) against annotated supporting regions.

7.4. GroundingDocQA-Bench

Existing document QA grounding benchmarks [10, 48, 50] contain only bounding box grounding. Therefore, we introduce **GroundingDocQA-Bench**, a segmentation based benchmark. We manually select document images from private sources as well as from the test sets of various public document datasets [15, 16, 18, 20, 22, 23, 33, 42, 44, 46]. Documents with curved and skewed text are specifically included to enable robust geometric and structural evaluation. Apart from regular text-rich documents, we specifically include other popular document categories such as charts, reports, forms, tables, webpages and infographics for improved diversity. We generate QA pairs using our data engine. Each QA pair is annotated by both text geometry-straight (70%) or curved (30%) and span type (2,820 single-span and 2,180 multi-span instances). The benchmark is manually curated and verified to ensure quality, and unbiased evaluation.

Dataset Statistics:

- Total Images: 2.5K
- Total Q/A Pairs: 5K
- Span Type Split: 2,820 single-span (56.4%) and 2,180 multi-span (43.6%) QA Pairs.
- Geometry Split: 70% straight text and 30% curved/skewed text.
 - Of the 30% curved Q/A pairs, 75% are single-span and 25% are multi-span.

Human Annotation & Verification Process To ensure fine grained grounding and answer correctness, all annotations in GroundingDocQA-Bench were validated by a team of 30 qualified human annotators. The annotators were intentionally recruited from diverse backgrounds including undergraduate students, office clerks, hospital staff, and front-desk operators to ensure diversity.

Each annotator underwent a short training module covering mask-annotation protocol, span definitions, curved-text segmentation, and QA correctness criteria. All pixel-level masks, span labels, and QA pairs were curated using a multi-stage verification pipeline:

- **Primary Verification:** Each document and its associated QA pair were first verified independently by one annotator, with segmentation performed using an XP-Pen tablet for high-precision boundary tracing.

- **Secondary Human Verification:** A different annotator reviewed the masks and QA pair for correctness, checking for leakage, incorrect spans, missed curved segments, and ambiguity.

This human-in-the-loop pipeline ensured that GroundingDocQA Bench provides high-quality, human-verified, multi-granular mask annotations and reliable QA pairs.

Evaluation Protocol: We evaluate our models at all three mask levels: phrase, line and block. Since our benchmark contains different types of answer spans, we report four grounding metrics:

- SS (Single-Span F1): F1 score computed on QA pairs that have only one grounded span. This measures how well the model can ground a single piece of text in the image.
- MS (Multi-Span F1): F1 score computed on questions that require grounding two or more spans. This checks whether the model can find all required regions for multi-part answers.
- CS (Curved-Skewed F1): F1 score on QA pairs that contain curved or skewed text. This shows the model’s ability to handle non-straight text which is common in posters, infographics and charts.
- Overall F1_g: Overall grounding F1 across the full benchmark. This combines SS, MS and CS and gives one final grounding score.

For answer quality, we use **AQ** computed with G-Eval [19]. AQ measures if the predicted answer text is semantically correct when compared to the ground truth. This setup allows a clear comparison between models on different types of grounding difficulty.

8. Qualitative Results

We provide qualitative results illustrating the behavior of M3Grounder across a variety of document types. To organize the examples clearly, we divide them into two groups: (1) comparisons between our model and the next best performing model, and (2) standalone outputs produced solely by M3Grounder. These examples span charts, financial documents, infographics, curved text, and layout-heavy pages.

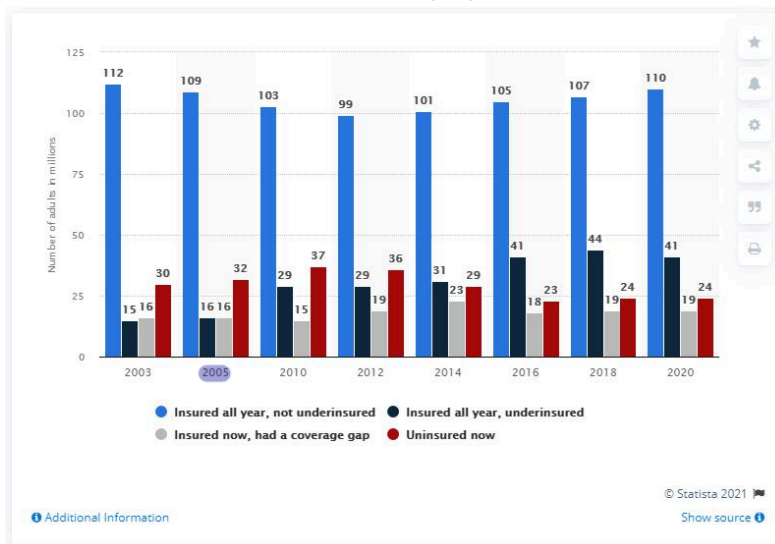
8.1. Comparisons Against the Next Best Model



Q: Which year has a ratio of 1:1 for Insured all year, under insured and Insured now, had a coverage gap?

Ground Truth: 2005

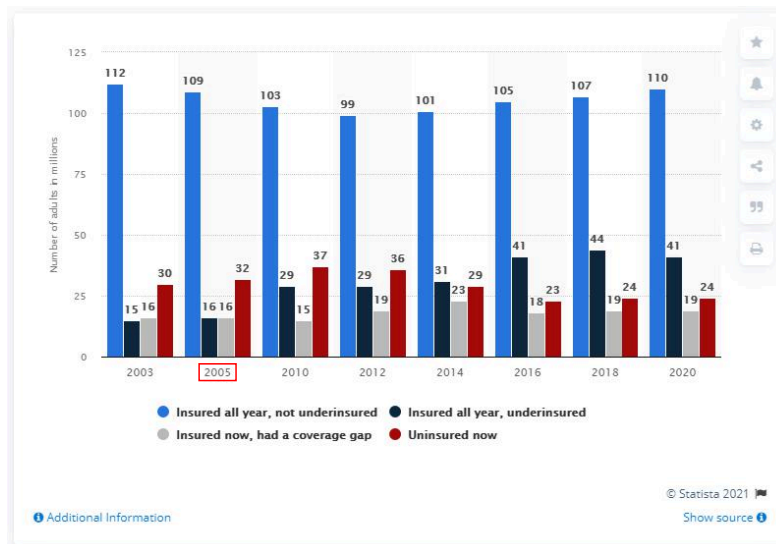
Model: M3Grounder (Ours)



Predicted Answer: <2005>[GROUND]



Model: Gemini 2.5 Pro



Predicted Answer: 2005 [BBOX]

Figure 14. Comparison Example

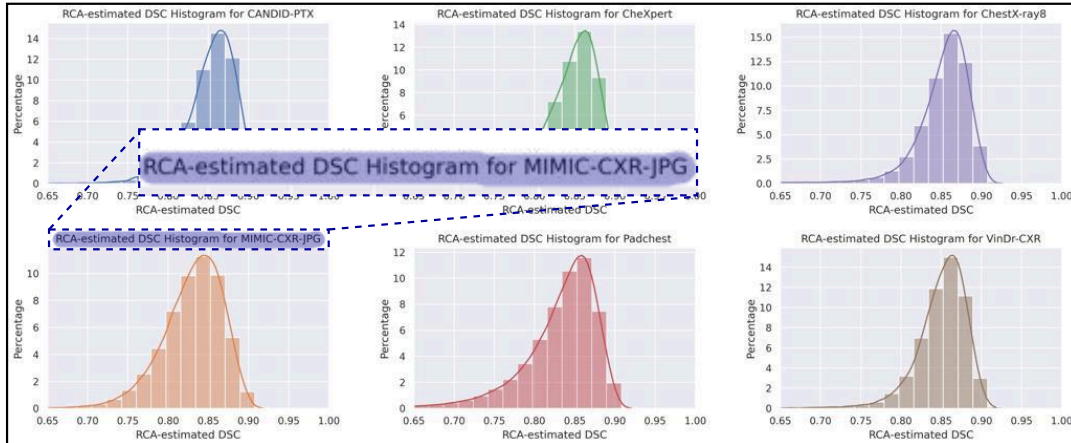


Q: For the subplot at row 2 and column 1, what is its title?



Ground Truth: RCA-estimated DSC Histogram for MIMIC-CXR-JPG

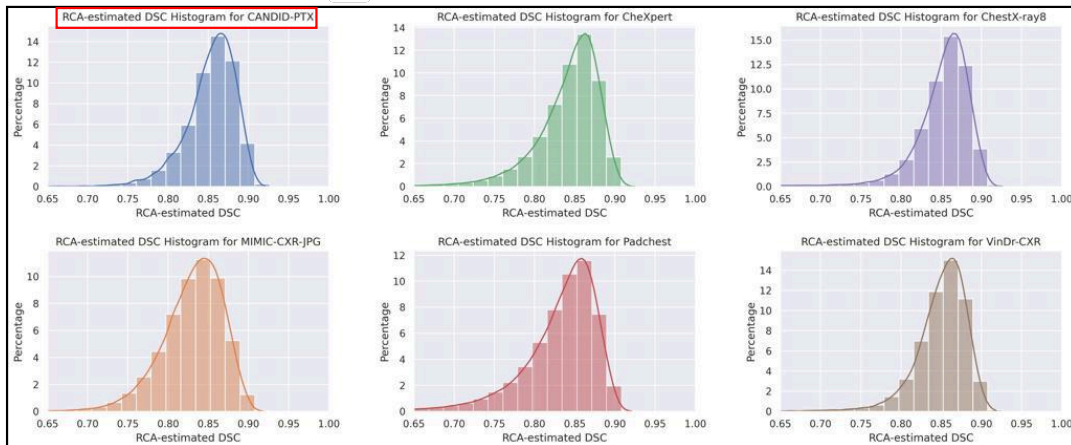
Model: M3Grounder (Ours)



Predicted Answer: <e>RCA-estimated DSC Histogram for MIMIC-CXR-JPG</e>[GROUND]



Model: Gemini 2.5 Pro



Predicted Answer: : RCA-estimate DSC Histogram for CANDID-PTX[BBOX]

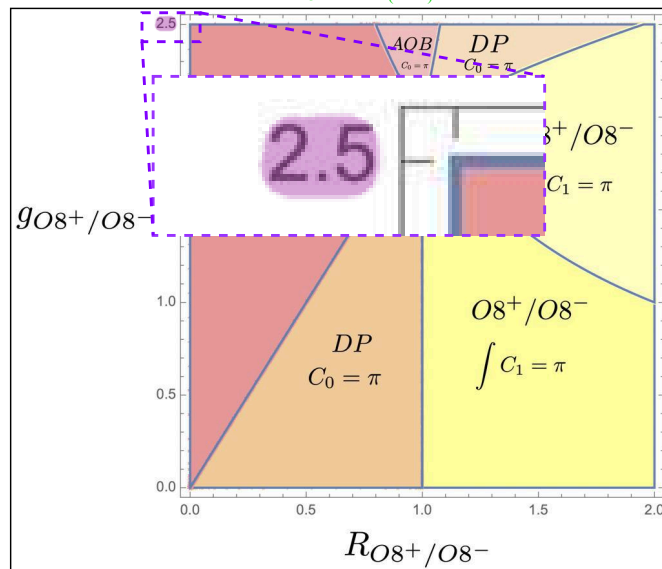
Figure 15. Comparison Example



Q: For the current plot, what is the spatially highest labeled tick on the y-axis?

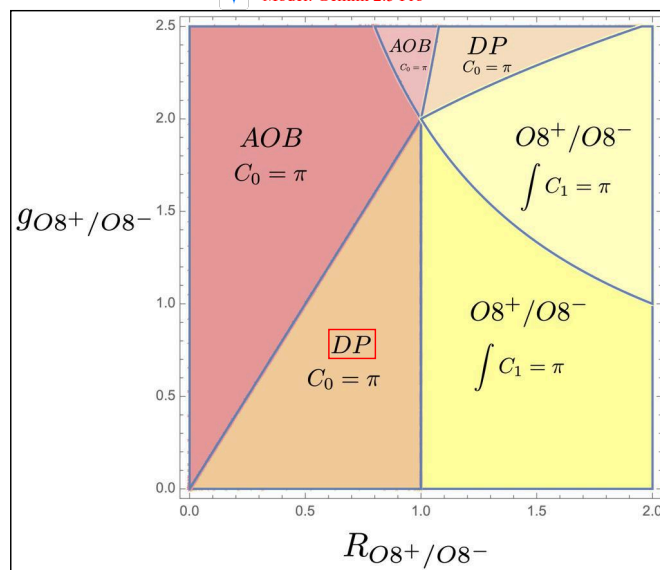
Ground Truth: 2.5

Model: M3Grounder (Ours)



Predicted Answer: $\langle e \rangle 2.5 \langle /e \rangle$ [GROUND]

Model: Gemini 2.5 Pro



Predicted Answer: 2.5 [BBOX]

Figure 16. Comparison Example



Q: Your task is to identify the text of the field "tax amount"



Model: M3Grounder (Ours)

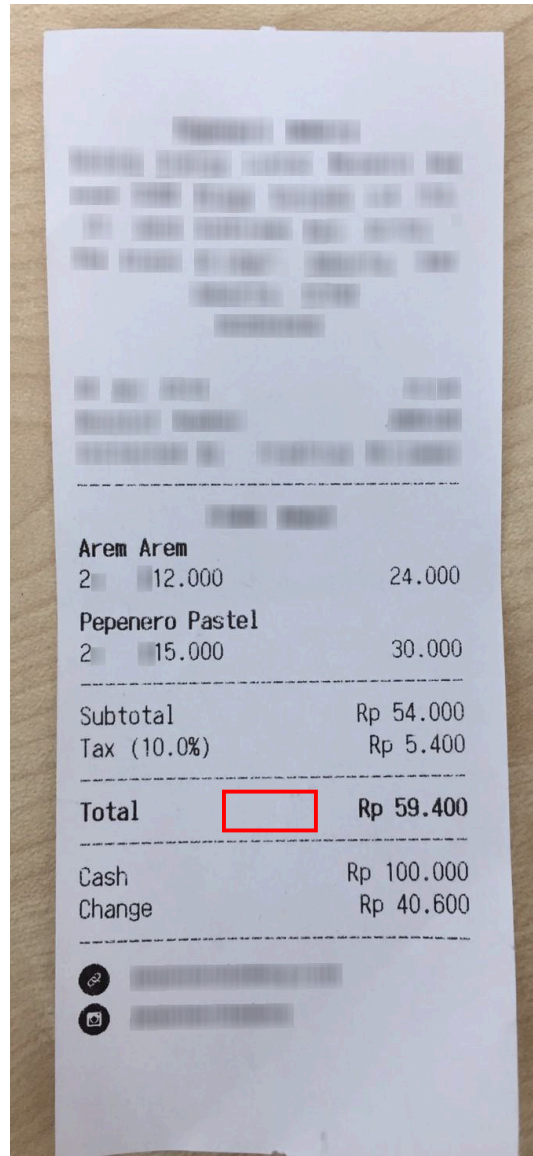
Ground Truth: 5.400



Model: Gemini 2.5 Pro



Predicted Answer: `<e>5.400</e>`[GROUND]



Predicted Answer: 5.400 [BBOX]

Figure 17. Comparison Example

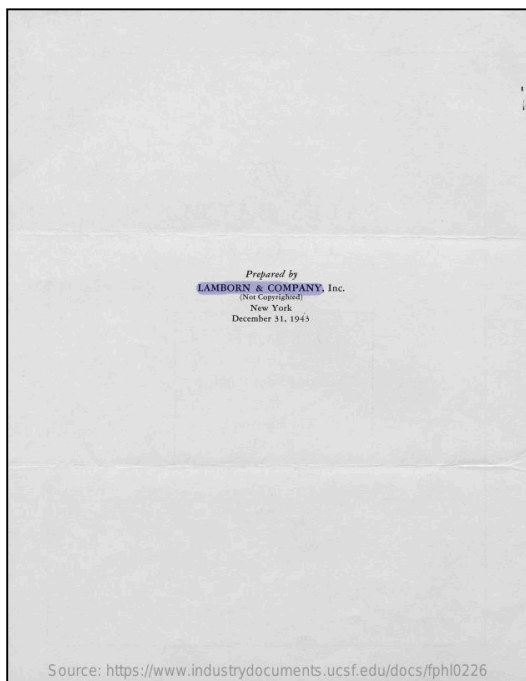


Q: What is written in bold letters?



Ground Truth: LAMBORN & COMPANY

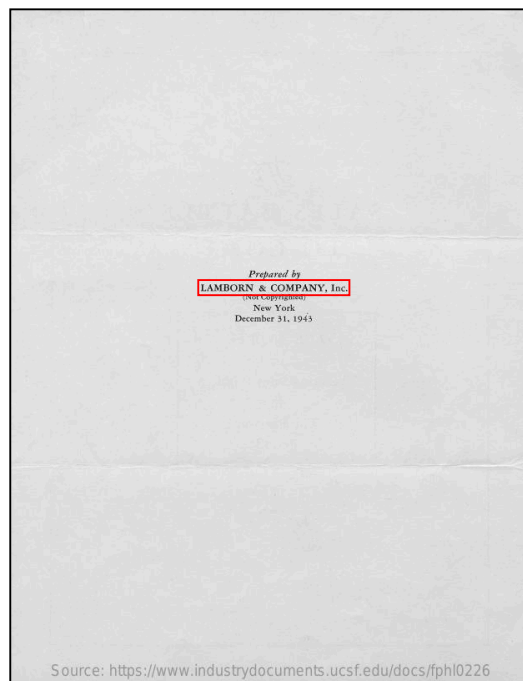
Model: M3Grounder (Ours)



Predicted Answer: <e>LAMBORN & COMPANY</e>[GROUND]



Model: Gemini 2.5 Pro



Predicted Answer: LAMBORN & COMPANY Inc. [BBOX]

Figure 18. Comparison Example



Q: What percentage of the total football club revenue is contributed by the commercials in 2018?

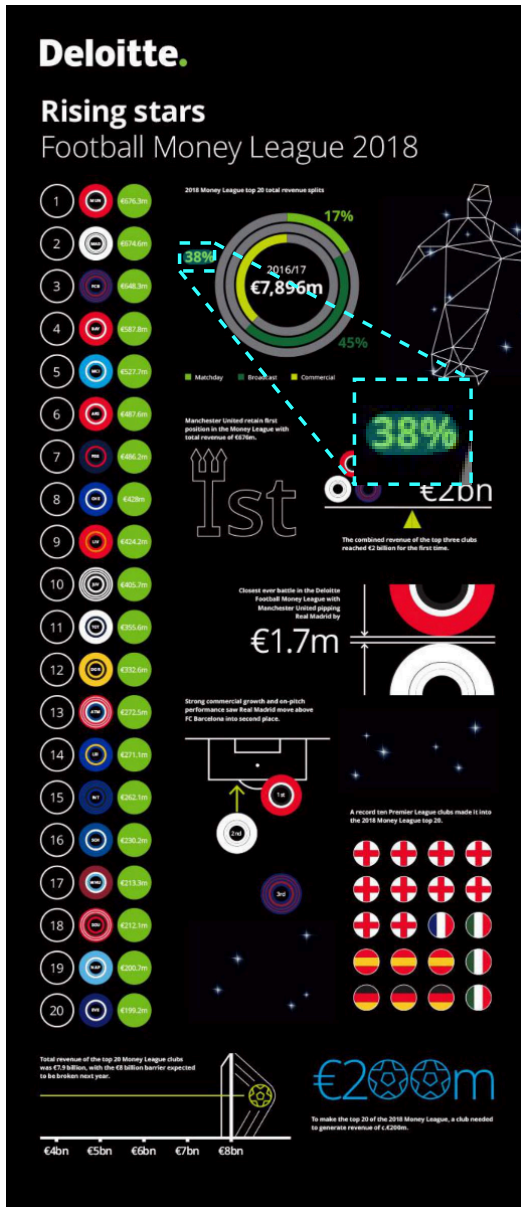


Ground Truth: 38%

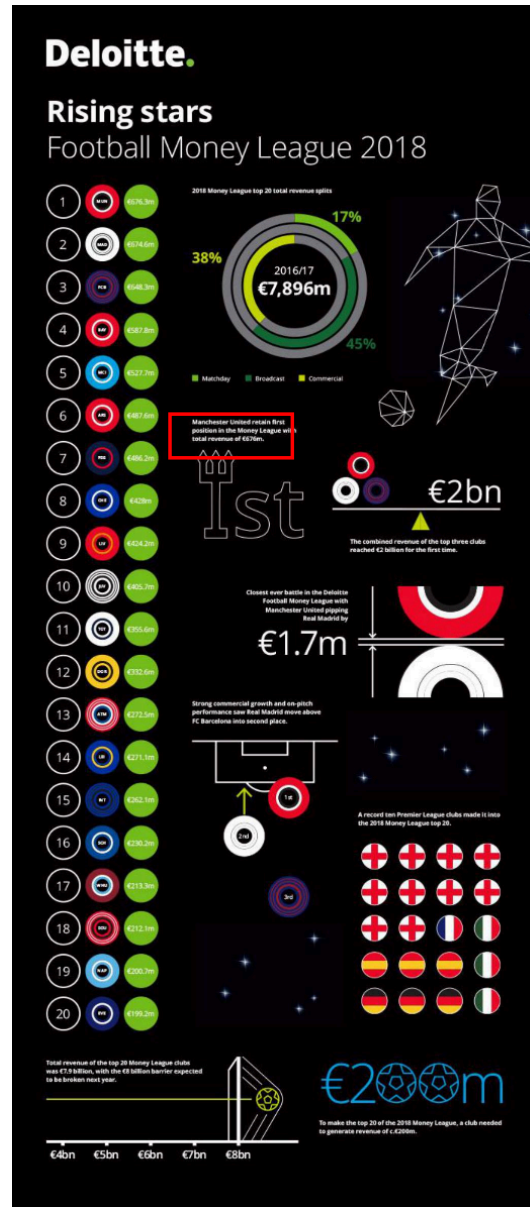
Model: M3Grounder (Ours)



Model: Gemini 2.5 Pro



Predicted Answer: <e>38%</e>[GROUND]



Predicted Answer: 38% [BBOX]

Figure 19. Comparison Example



Q: What were the income taxes payable in 2019?

Ground Truth: 46.9

Model: M3Grounder (Ours)

Table of Contents	
Inventories	
The components of inventories consist of the following (in millions):	
	March 31,
	2019 2018
Raw materials	\$ 74.5 \$ 26.0
Work in process	413.0 311.8
Finished goods	224.2 136.4
Total inventories	\$ 711.7 \$ 476.2
Inventories are valued at the lower of cost and net realizable value using the first-in, first-out method. Inventory impairment charges establish a new cost basis for inventory and charges are not subsequently reversed to income even if circumstances later suggest that increased carrying amounts are recoverable.	
Property, Plant and Equipment	
Property, plant and equipment consists of the following (in millions):	
	March 31,
	2019 2018
Land	\$ 83.4 \$ 73.4
Building and building improvements	647.6 508.5
Machinery and equipment	2,095.5 1,943.9
Projects in process	119.2 118.3
Total property, plant and equipment, gross	2,945.7 2,644.1
Less accumulated depreciation and amortization	1,949.0 1,876.2
Total property, plant and equipment, net	\$ 996.7 \$ 767.9
Depreciation expense attributed to property, plant and equipment was \$180.6 million, \$123.7 million and \$122.9 million for the fiscal years ending March 31, 2019, 2018 and 2017, respectively.	
Accrued Liabilities	
Accrued liabilities consists of the following (in millions):	
	March 31,
	2019 2018
Accrued compensation and benefits	\$ 133.2 \$ 87.6
Income taxes payable	46.9 27.5
Sales related reserves	366.9 —
Accrued expenses and other liabilities	240.3 114.5
Total accrued liabilities	\$ 787.3 \$ 229.6
Sales related reserves represent price concessions and stock rotation rights that the Company offers to many of its distributors. For the fiscal year ending March 31, 2018, these sales related reserves were recorded within accounts receivable, and therefore did not exist within accrued liabilities. The Company made this change in classification as part of its adoption of ASC 606. For additional information regarding the Company's adoption of ASC 606, refer to Note 1 of the consolidated financial statements.	
F-33	

Predicted Answer: <e>46.9</e>[GROUND]



Model: Gemini 2.5 Pro

Table of Contents	
Inventories	
The components of inventories consist of the following (in millions):	
	March 31,
	2019 2018
Raw materials	\$ 74.5 \$ 26.0
Work in process	413.0 311.8
Finished goods	224.2 136.4
Total inventories	\$ 711.7 \$ 476.2
Inventories are valued at the lower of cost and net realizable value using the first-in, first-out method. Inventory impairment charges establish a new cost basis for inventory and charges are not subsequently reversed to income even if circumstances later suggest that increased carrying amounts are recoverable.	
Property, Plant and Equipment	
Property, plant and equipment consists of the following (in millions):	
	March 31,
	2019 2018
Land	\$ 83.4 \$ 73.4
Building and building improvements	647.6 508.5
Machinery and equipment	2,095.5 1,943.9
Projects in process	119.2 118.3
Total property, plant and equipment, gross	2,945.7 2,644.1
Less accumulated depreciation and amortization	1,949.0 1,876.2
Total property, plant and equipment, net	\$ 996.7 \$ 767.9
Depreciation expense attributed to property, plant and equipment was \$180.6 million, \$123.7 million and \$122.9 million for the fiscal years ending March 31, 2019, 2018 and 2017, respectively.	
Accrued Liabilities	
Accrued liabilities consists of the following (in millions):	
	March 31,
	2019 2018
Accrued compensation and benefits	\$ 133.2 \$ 87.6
Income taxes payable	46.9 27.5
Sales related reserves	366.9 —
Accrued expenses and other liabilities	240.3 114.5
Total accrued liabilities	\$ 787.3 \$ 229.6
Sales related reserves represent price concessions and stock rotation rights that the Company offers to many of its distributors. For the fiscal year ending March 31, 2018, these sales related reserves were recorded within accounts receivable, and therefore did not exist within accrued liabilities. The Company made this change in classification as part of its adoption of ASC 606. For additional information regarding the Company's adoption of ASC 606, refer to Note 1 of the consolidated financial statements.	
F-33	

Predicted Answer: 46.9 [BBOX]

Figure 20. Comparison Example



Q: Using the provided image and text "Pre-pregnancy weight", identify and return the text's location in the image.

Model: M3Grounder (Ours)

Characteristic	Births by smoking status,* %						Total (% missing)
	Never n = 197,583	Former n = 12,256	1-4 n = 7,806	5-9 n = 5,839	10 + n = 10,407	Missing n = 2,512	
Maternal Age	< 20	2.3	7.8	14.0	12.2	9.2	11.8
	20-24	12.4	24.3	32.4	32.7	30.3	29.0
	25-29	27.9	29.5	26.8	27.2	26.8	25.7
	30-34	34.7	24.5	17.4	17.3	20.3	21.1
	35-39	18.7	11.5	7.7	8.8	10.8	10.1
	40+	4.0	2.4	1.7	1.8	2.5	2.2
Parity ≥ 1	No	44.3	59.3	54.7	47.3	40.2	52.5
	Yes	55.7	40.7	45.3	52.7	59.8	47.5
	missing	0.0	0.0	0.0	0.0	0.0	0.0
Single Parent	No	92.7	87.1	76.5	77.8	75.1	75.6
	Yes	3.7	8.7	15.0	15.1	16.7	18.3
	Unknown	3.6	4.1	8.6	7.1	8.2	6.1
Has Grade 12	No	0.8	3.0	5.2	5.2	5.4	3.1
	Yes	10.7	12.9	9.6	9.4	6.5	6.7
	missing	88.5	84.1	85.2	85.4	88.1	90.2
Gestational diabetes	No	93.3	94.6	96.2	95.8	95.1	94.7
	Yes	6.7	5.4	3.8	4.2	4.9	5.3
Pre-existing Diabetes	No	99.6	99.7	99.6	99.6	99.4	99.6
	Yes	0.4	0.3	0.4	0.4	0.6	0.4
Hypertension in pregnancy	No	97.8	97.1	98.1	98.0	98.2	97.8
	Yes	2.2	2.8	1.9	2.0	1.8	2.2
Indication of Alcohol Use	No	99.7	99.1	99.9	99.6	95.1	93.8
	Yes	0.4	1.8	4.1	3.4	4.9	6.3
Indication of Drug Use	No	99.2	96.8	90.8	90.3	84.6	86.7
	Yes	0.8	3.1	9.2	9.7	15.4	13.1
Pre-pregnancy weight	< 55	21.0	16.4	19.0	19.9	18.8	16.4
	55-74	42.9	43.6	38.8	38.6	36.2	35.0
	> 74	14.7	22.8	18.1	18.4	20.3	16.7
	missing	21.5	18.1	24.1	23.1	24.7	31.8
	≥ 3	92.1	93.4	92.3	91.2	89.5	85.9
Prenatal Care Visits	< 3	1.2	1.2	2.5	3.0	3.4	3.8
	≥ 3	92.1	93.4	92.3	91.2	89.5	85.9
	missing	6.7	5.5	5.2	5.8	7.1	10.3

Predicted Answer: <e>Pre-pregnancy weight</e>[GROUND]

Model: Gemini 2.5 Pro

Characteristic	Births by smoking status,* %						Total (% missing)
	Never n = 197,583	Former n = 12,256	1-4 n = 7,806	5-9 n = 5,839	10 + n = 10,407	Missing n = 2,512	
Maternal Age	< 20	2.3	7.8	14.0	12.2	9.2	11.8
	20-24	12.4	24.3	32.4	32.7	30.3	29.0
	25-29	27.9	29.5	26.8	27.2	26.8	25.7
	30-34	34.7	24.5	17.4	17.3	20.3	21.1
	35-39	18.7	11.5	7.7	8.8	10.8	10.1
	40+	4.0	2.4	1.7	1.8	2.5	2.2
Parity ≥ 1	No	44.3	59.3	54.7	47.3	40.2	52.5
	Yes	55.7	40.7	45.3	52.7	59.8	47.5
	missing	0.0	0.0	0.0	0.0	0.0	0.0
Single Parent	No	92.7	87.1	76.5	77.8	75.1	75.6
	Yes	3.7	8.7	15.0	15.1	16.7	18.3
	Unknown	3.6	4.1	8.6	7.1	8.2	6.1
Has Grade 12	No	0.8	3.0	5.2	5.2	5.4	3.1
	Yes	10.7	12.9	9.6	9.4	6.5	6.7
	missing	88.5	84.1	85.2	85.4	88.1	90.2
Gestational diabetes	No	93.3	94.6	96.2	95.8	95.1	94.7
	Yes	6.7	5.4	3.8	4.2	4.9	5.3
Pre-existing Diabetes	No	99.6	99.7	99.6	99.6	99.4	99.6
	Yes	0.4	0.3	0.4	0.4	0.6	0.4
Hypertension in pregnancy	No	97.8	97.1	98.1	98.0	98.2	97.8
	Yes	2.2	2.8	1.9	2.0	1.8	2.2
Indication of Alcohol Use	No	99.7	99.1	99.9	99.6	95.1	93.8
	Yes	0.4	1.8	4.1	3.4	4.9	6.3
Indication of Drug Use	No	99.2	96.8	90.8	90.3	84.6	86.7
	Yes	0.8	3.1	9.2	9.7	15.4	13.1
Pre-pregnancy weight	< 55	21.0	16.4	19.0	19.9	18.8	16.4
	55-74	42.9	43.6	38.8	38.6	36.2	35.0
	> 74	14.7	22.8	18.1	18.4	20.3	16.7
	missing	21.5	18.1	24.1	23.1	24.7	31.8
	≥ 3	92.1	93.4	92.3	91.2	89.5	85.9
Prenatal Care Visits	< 3	1.2	1.2	2.5	3.0	3.4	3.8
	≥ 3	92.1	93.4	92.3	91.2	89.5	85.9
	missing	6.7	5.5	5.2	5.8	7.1	10.3

Predicted Answer: Pre-pregnancy weight [BBOX]

Figure 21. Comparison Example



Q: Which vehicle class originally had the same price as a car with trailer?

Ground Truth: Class 4

Model: M3Grounder (Ours)

Date introduced	Class 1 (e.g. Motorbike)	Class 2 (e.g. Car)	Class 3 (e.g. Car with trailer)	Class 4 (e.g. Van)	Class 5 (e.g. HGV)
9 December 2003 ^[13]	£1.00	£2.00	£5.00	£5.00	£10.00
23 July 2004 ^[8]	£1.00	£2.00	£5.00	£5.00	£6.00
16 August 2004 ^[8]	£2.00	£3.00	£6.00	£6.00	£6.00
14 June 2005 ^[14]	£2.50	£3.50	£7.00	£7.00	£7.00
1 January 2008 ^{[15][16]}	£2.50	£4.50	£8.00	£9.00	£9.00
1 January 2009 ^[17]	£2.70	£4.70	£8.40	£9.40	£9.40
1 March 2010 ^[18]	£2.70	£5.00	£9.00	£10.00	£10.00
1 March 2011 ^[19]	£3.00	£5.30	£9.60	£10.60	£10.60
1 March 2012 ^[20]	£3.00	£5.50	£10.00	£11.00	£11.00

Predicted Answer: <e>Class 4 (e.g. Van)</e>[GROUND]



Model: Gemini 2.5 Pro

Date introduced	Class 1 (e.g. Motorbike)	Class 2 (e.g. Car)	Class 3 (e.g. Car with trailer)	Class 4 (e.g. Van)	Class 5 (e.g. HGV)
9 December 2003 ^[13]	£1.00	£2.00	£5.00	£5.00	£10.00
23 July 2004 ^[8]	£1.00	£2.00	£5.00	£5.00	£6.00
16 August 2004 ^[8]	£2.00	£3.00	£6.00	£6.00	£6.00
14 June 2005 ^[14]	£2.50	£3.50	£7.00	£7.00	£7.00
1 January 2008 ^{[15][16]}	£2.50	£4.50	£8.00	£9.00	£9.00
1 January 2009 ^[17]	£2.70	£4.70	£8.40	£9.40	£9.40
1 March 2010 ^[18]	£2.70	£5.00	£9.00	£10.00	£10.00
1 March 2011 ^[19]	£3.00	£5.30	£9.60	£10.60	£10.60
1 March 2012 ^[20]	£3.00	£5.50	£10.00	£11.00	£11.00

Predicted Answer: 5.400 [BBOX]

Figure 22. Comparison Example

8.2. Standalone Predictions from M3Grounder



According to the document, what was the Customer Support renewal rate for the quarter ended June 30, 2019, and what was the Customer Support revenue in the Americas for 2019? Provide the answer with grounding



The document reports a Customer Support renewal rate of **<e>approximately 91%</e>** [GROUND] and an Americas Customer Support revenue of **<e>718,209</e>** [GROUND] for 2019.

an increase in Americas of \$56.4 million, and an increase in EMEA of \$12.4 million and an increase in Asia Pacific of \$3.7 million.

The \$6.4 million difference between cloud service and subscription revenues recognized under Topic 606 and those proforma Topic 605 cloud services and subscriptions revenues described above is primarily the result of timing differences on professional services related to cloud contracts, where under Topic 605, revenues would have been deferred over the estimated life of the contract, but under Topic 606 these revenues are recognized as services are performed. For more details, see note 3 "Revenues" to our Consolidated Financial Statements.

3) Customer Support:

Customer support revenues consist of revenues from our customer support and maintenance agreements. These agreements allow our customers to receive technical support, enhancements and upgrades to new versions of our software products when and if available. Customer support revenues are generated from support and maintenance relating to current year sales of software products and from the renewal of existing maintenance agreements for software licenses sold in prior periods. Therefore, changes in Customer support revenues do not always correlate directly to the changes in license revenues from period to period. The terms of support and maintenance agreements are typically twelve months, and are renewable, generally on an annual basis, at the option of the customer. Our management reviews our Customer support renewal rates on a quarterly basis and we use these rates as a method of monitoring our customer service performance. For the quarter ended June 30, 2019, our Customer support renewal rate was **approximately 91%**, stable compared with the Customer support renewal rate during the quarter ended June 30, 2018.

Cost of Customer support revenues is comprised primarily of technical support personnel and related costs, as well as third party royalty costs.

(In thousands)	Year Ended June 30,				
	2019	Change increase (decrease)	2018	Change increase (decrease)	2017
Customer Support Revenues:					
Americas	\$ 718,209	\$ 12,924	\$ 705,285	\$ 122,870	\$ 582,415
EMEA	427,712	3,939	423,773	103,145	320,628
Asia Pacific	101,994	(1,452)	103,446	25,387	78,059
Total Customer Support Revenues	1,247,915	15,411	1,232,504	251,402	981,102
Cost of Customer Support Revenues	124,343	(9,546)	133,889	11,324	122,565
GAAP-based Customer Support Gross Profit	\$ 1,123,572	\$ 24,957	\$ 1,098,615	\$ 240,078	\$ 858,537
GAAP-based Customer Support Gross Margin %	90.0%		89.1%		87.5%

% Customer Support Revenues by Geography:

Americas	57.6%	57.2%	59.4%
EMEA	34.3%	34.4%	32.7%
Asia Pacific	8.1%	8.4%	7.9%

Customer support revenues increased by \$15.4 million or 1.3% during the year ended June 30, 2019 as compared to the prior fiscal year; up 3.1% after factoring the impact of \$23.2 million of foreign exchange rate changes. Geographically, the overall change was attributable to an increase in Americas of \$12.9 million, an increase in EMEA of \$3.9 million, partially offset by a decrease in Asia Pacific of \$1.5 million.

Cost of Customer support revenues decreased by \$9.5 million during the year ended June 30, 2019 as compared to the prior fiscal year, due to a decrease in labour-related costs of approximately \$9.9 million, partially offset by an increase in other miscellaneous costs of \$0.4 million. Overall, the gross margin percentage on Customer support revenues increased to approximately 90% from approximately 89%.

For illustrative purposes only, had we accounted for revenues under proforma Topic 605, customer support revenues would have been \$1,246.3 million for the year ended June 30, 2019, which would have been higher by approximately \$13.8 million or 1.1% as compared to the prior fiscal year; and would have been up 3.0% after factoring the impact of \$23.3 million of foreign exchange rate changes. Geographically, the overall change would have been attributable to an increase in Americas of \$13.0 million and an increase in EMEA of \$2.7 million, partially offset by a decrease in Asia Pacific of \$1.9 million.

Figure 23. Qualitative Example




What message is written across the curved banner below the motorcycle?



<e>HAPPY TOURING AND FUN DAYS AHEAD</e> [GROUND]

RIDE A NEW HARLEY-DAVIDSON




HAPPY TOURING AND FUN DAYS AHEAD

The new Harley-Davidsons are rolling off the production line — ready to thrill more and more riders who want the power, rugged dependability and proved performance of "The World's Champion Motorcycles." Get set for fullest enjoyment of glorious, fun-packed days ahead — as you ride *your own* Harley-Davidson on pleasure-filled tours and exciting endurance runs, or take in race meets, hillclimbs and other motorcycling good times. Get your order in early — see your Harley-Davidson dealer NOW!

HARLEY-DAVIDSON
Motorcycles

HARLEY-DAVIDSON MOTOR COMPANY, Dept. P5, Milwaukee 1, Wis.
Please send at once free copy of 24-page "Enthusiast" Magazine, filled with motorcycle action pictures and thrilling stories.

Name _____
Address _____
City _____ State _____



224 POPULAR SCIENCE



Figure 24. Qualitative Example



What can be inferred from the low values in this blood report? Provide the answer with grounding



The report shows a low **MEAN CORPUSCULAR VOLUME, MCV** of **72.0 fL** which indicates **Microcytic anemia** [GROUND]

Mr. Saubhik Bhaumik Age / Sex : 27YRS / M Referred by : Self Reg. no. : 1001	 1001 Registered on : 17/10/2024 04:55 PM Collected on : 17/10/2024 Received on : 17/10/2024 Reported on : 17/10/2024 04:55 PM	
--	---	---


HAEMATOLOGY
COMPLETE BLOOD COUNT (CBC)

TEST	VALUE	UNIT	REFERENCE
HEMOGLOBIN	15	g/dl	13 - 17
TOTAL LEUKOCYTE COUNT	5,100	cumm	4,800 - 10,800
DIFFERENTIAL LEUCOCYTE COUNT			
NEUTROPHILS	79	%	40 - 80
LYMPHOCYTE	28	%	20 - 40
EOSINOPHILS	1	%	1 - 6
MONOCYTES	4	%	2 - 10
BASOPHILS	1	%	< 2
PLATELET COUNT	3.5	lakhs/cumm	1.5 - 4.1
TOTAL RBC COUNT	5	million/cumm	4.5 - 5.5
HEMATOCRIT VALUE, HCT	42	%	40 - 50
MEAN CORPUSCULAR VOLUME, MCV	L 72.0	fL	83 - 101
MEAN CELL HAEMOGLOBIN, MCH	30.0	Pg	27 - 32
MEAN CELL HAEMOGLOBIN CON, MCHC	32.7	%	31.5-34.5


Clinical Notes:
A complete blood count (CBC) is used to evaluate overall health and detect a wide range of disorders, including anemia, infection, and leukemia. There have been some reports of WBC and platelet counts being lower in venous blood than in capillary blood samples, although still within these reference ranges.

Possible causes of abnormal parameters:

	High	Low
RBC, Hb, or HCT	Dehydration, polycythemia, shock, chronic hypoxia	Anemia, thalassemia, and other hemoglobinopathies
MCV	Macrocytic anemia, liver disease	Microcytic anemia
WBC (Leucocytes)	Acute stress, infection, malignancies	Infection risk, weak immune system
Platelets	Risk of thrombosis	Risk of bleeding


Mr. Sachin Sharma
DMLT, Lab Incharge

Page 1 of 2


Dr. A. K. Asthana
MBBS, MD Pathologist

NOT VALID FOR MEDICO LEGAL PURPOSE

Work timings: Monday to Sunday, 8 am to 8 pm

Please correlate clinically. Although the test results are checked thoroughly, in case of any unexpected test results which could be due to machine error or typing error or any other reason please contact the lab immediately for a free evaluation.

Figure 25. Qualitative Example



Which types of organizations may be considered as the Auditee according to this NDA?



The NDA specifies that the Auditee may belong to <e>Central Government Ministry/ Departments #/State Government Departments DLSP/DL Repository</e>[GROUND] as well as <e>Autonomous Societies/ Not-for-profit companies/ Public sector Undertakings/ Private sector DLSP/DL Repository</e>[GROUND]

NON-DISCLOSURE AGREEMENT	
(Between STQC empaneled Auditor & Auditee)	
THIS NON-DISCLOSURE AGREEMENT is made on this Day (date) of..... (Year) By and between	
# In case of Central Government Ministry/ Departments #/State Government Departments DLSP/DL Repository	
President of India/Governor of (name of state) acting through (Name, Designation) of (Name of Ministry/ Department) address hereinafter referred to as "Auditee" which expression shall, unless repugnant to the context or meaning thereof, include its successors and assigns) of the first part.	
# In case of Autonomous Societies/ Not-for-profit companies/ Public sector Undertakings/ Private sector DLSP/DL Repository	
..... (Name of Company/ Society) incorporated /registered under the Companies Act,1956/2013/ the societies registration Act,1860 having its registered/corporate office at (hereinafter referred to as "Auditee" which expression shall, unless repugnant to the context or meaning thereof, includes its successors, administrators and permitted assigns) of the first part.	
And	
Name incorporated/registered under the Name of the Act having its registered/corporate office at (Herein referred to as "Auditor" which expression shall, unless repugnant to the context or meaning thereof, include its successors, assigns, administrators, liquidators and receivers) of the second part	
WHEREAS	
A. Auditor (Audit Organisation) is a services organization empaneled by the Standardisation, Testing and Quality Certification Response Team (hereinafter referred to as STQC) under Ministry of Electronics & IT, for auditing, including vulnerability assessment and penetration testing of Digital Locker services, networks, computer resources & applications of various agencies or departments of the Government.	
B. Auditor, as an empanelled Information Security and service management Auditing organization, has agreed to fully comply with the "Rules and procedures of Digital Locker Service Provider (DLSP) / DL Repositories certification scheme, Terms & conditions of empanelment and Policy guidelines for handling audit related data"	

Figure 26. Qualitative Example



Which sectors border the Village Sohana layout?



<e>SECTOR-88</e>[GROUND], <e>SECTOR-76</e>
[GROUND], <e>SECTOR-70</e>[GROUND]

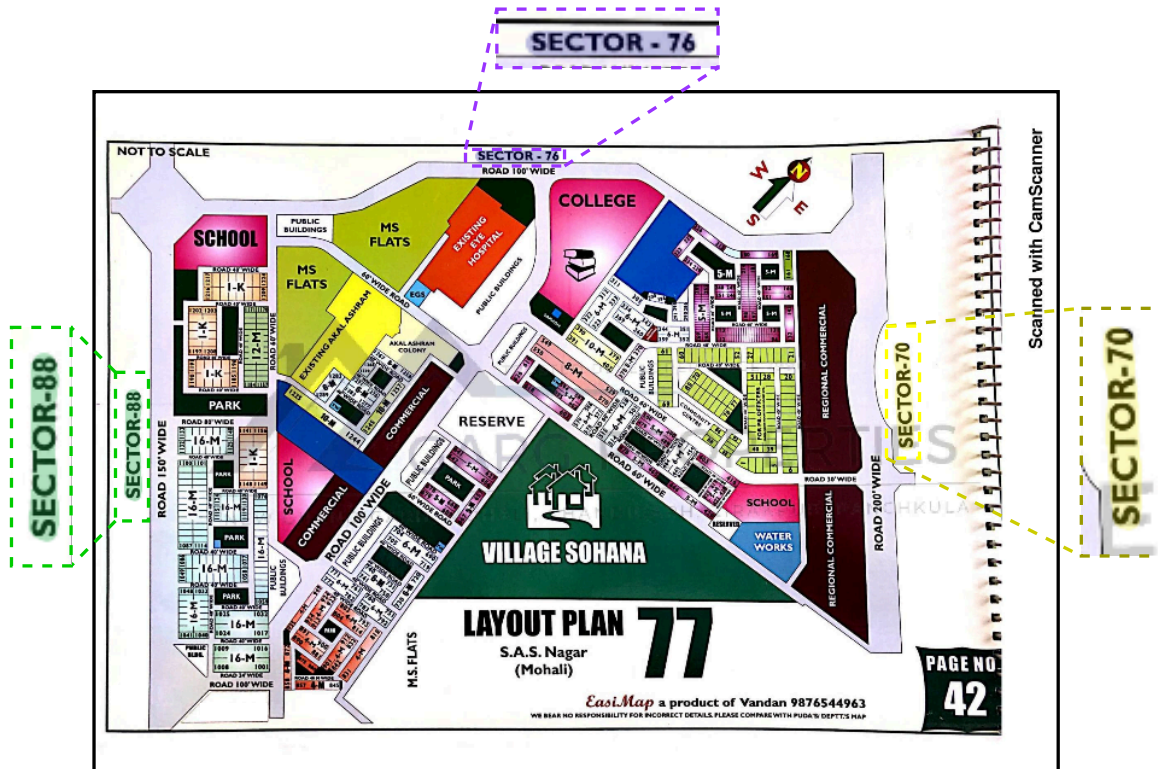


Figure 27. Qualitative Example



What does Popeye say in the speech bubble?



`<e>I YAM WHAT I YAM!</e>[GROUND]`

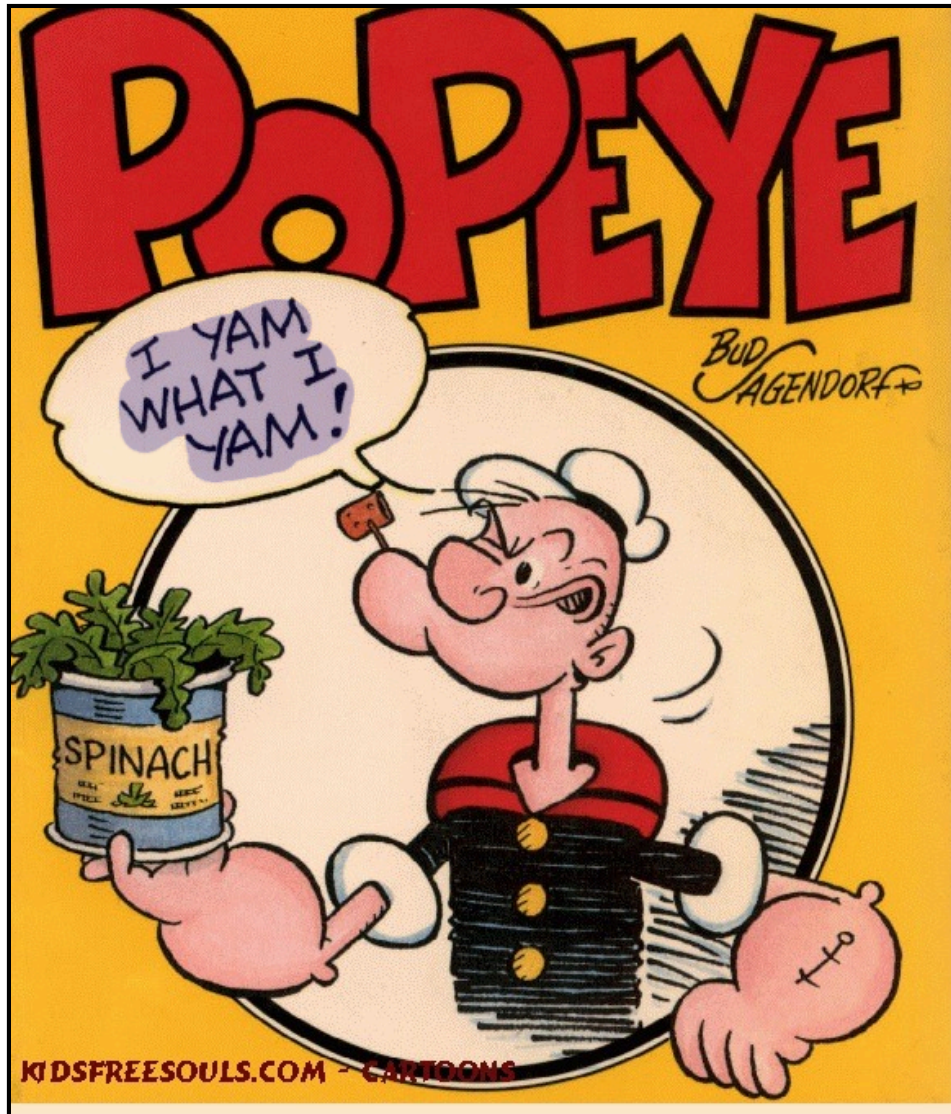


Figure 28. Qualitative Example



What is the student's number, and what grade did they receive in Data Analytics?



The student's number is **3123456** and the grade they received in Data Analytics is **HD**


Student Name					
Student Number		3123456			
Foundation Studies: Business					
Completion		16 January 2025			
Conferral		31 January 2025			
Year	Course Code	Course Title	Unit	Mark	Grade
2024	ACCT2339	Foundation Accounting	12	66	CR
	ECON1604	Principles of Economics	12	69	CR
	ISYS3453	Data Analytics	12	93	HD
	LANG1377	Comparative Texts	12	79	DI
	LANG1383	Exploring Narratives	12	74	DI
	LAW2617	Law and Society	12	71	DI
	MATH2425	Statistics and Probability	12	79	DI
	MKTG1492	Marketing and Communication	12	62	CR
Cumulative GPA: Not Applicable					
Bachelor of Business					
Year	Course Code	Course Title	Unit	Mark	Grade
2025	BUSM2565	Understanding the Business Environment	12	61	CR
	BUSM2575	Business in Society	12	65	CR
	BUSM2580	Integrated Perspectives on Business Problems	12	68	CR
	OMGT2085	Introduction to Logistics and Supply Chain Management	12	**	**
	OMGT2199	Operations Management	12	**	**
	OMGT2267	Supply Chain Technologies	12	**	**
	OMGT2277	Supply Chain Analytics	12	**	**
Cumulative GPA: 2.0					
English for Teens					
Year	Course Code	Course Title	Unit	Mark	Grade
2020	ENGL2069	Tee-L10-Upper Intermediate B	**	**	A
	ENGL2072	Tee-L11-Pre Advanced A	**	**	A
Cumulative GPA: Not Applicable					
End of Academic Record					
					
Connie Merlino University Secretary and Academic Registrar					
Date of Issue: 23 July 2025					
Page 1 of 1					

Figure 29. Qualitative Example

Overall, these qualitative examples demonstrate that segmentation-based hierarchical grounding yields higher spatial fidelity and more reliable multi-span coverage than bounding-box-based approaches, particularly on visually complex

document types.

9. Miscellaneous

Other Baselines

Natural image segmentation models transfer poorly to document grounding QA [48]. To assess this, we evaluate two segmentation models: LISA++ [43] and GLaMM [30]. We test these models on the same set of benchmarks used in the main paper: BoundingDocs (Test) [10], DOGR-Bench [48], MMDocBench [50], and GroundingDocQA-Bench(ours).

Model	Params	BD-Test		DOGR-Bench		MMDocBench			GroundingDocQA-Bench(Ours)				
		F1 _g	AQ	F1 _g	AQ	EM	F1 _{txt}	IoU	SS	MS	CS	F1 _g	AQ
LISA++	7B	2.1	9.4	1.8	12.3	8.5	14.1	1.5	2.5	0.5	1.2	1.9	11.8
LISA++ ft.	7B	23.3	29.4	12.9	21.4	18.1	23.2	15.6	29.6	13.7	20.9	25.4	33.7
GLaMM	7B	3.4	12.2	2.6	14.9	12.8	16.4	2.8	3.8	0.9	2.1	3.1	15.5
GLaMM ft.	7B	32.1	41.2	20.8	32.6	21.7	25.2	19.1	41.8	17.3	30.8	35.2	39.3

Table 3. **Quantitative results for natural image segmentation models.** We evaluate LISA++ [43] and GLaMM [30] across four benchmarks: BoundingDocs (test) (BD-Test) [10], DOGR-Bench [48], MMDocBench [50], and GroundingDocQA-Bench (ours). Evaluation metrics follow the protocols defined in the main paper: For BD-Test and DOGR-Bench, grounding is measured by F1_g (IoU>0.5) and answer quality (AQ) by G-Eval [19]. For MMDocBench, we report Exact Match (EM), word-level F1_{txt}, and Region IoU. For GroundingDocQA-Bench, SS, MS, and CS denote single-span, multi-span, and curved/skewed grounding F1 (IoU>0.5), respectively, with overall F1_g aggregating across all spans.

Based on the scores we can conclude that these models are not optimized for grounding document question answering.

Removing Repeated Samples

To ensure the integrity of GroundingDocQA and prevent data leakage, we implemented a deduplication pipeline at both the document and question levels. Since our data engine aggregates images from multiple public datasets, ensuring uniqueness is important to avoid any kind of overlap and redundancy.

Document Samples Filtering:

- Text-Rich Documents: We utilized the Fid-HTML representations generated by the REPLICA engine [2]. By hashing the normalized HTML structure, we created a unique signature for each document layout.
- Charts: We utilized the underlying metadata JSON. We computed hash signatures based on the semantic content (titles, labels, data values) rather than the rendered pixels.

These structural signatures were cross-referenced both internally (to ensure pairwise uniqueness within our collection) and externally against the test splits of existing benchmarks (DOGR [48], MMDocBench [50], BoundingDocs [10]) to strictly prevent data leakage.

QA-Level Filtering: Within a document, we further filtered to ensure there is no spatial redundancy. To prevent artificial inflation of grounding scores, we identified and removed QA pairs that referenced identical grounding masks. This ensures that the model is evaluated on its ability to localize diverse regions rather than repeatedly predicting the same element.

DOGR Model

We utilize the scores reported in the original DOGR paper [48]. As our attempts to conduct local inference using the official codebase [49] were constrained by specific dependencies. The available implementation appears optimized for NPU setup, which resulted in compatibility challenges within our GPU setup, particularly for vision encoder checkpoints and tensor shape alignment. Attempts made to solve these issues were unsuccessful [49]. To avoid misrepresenting the model’s capabilities due to these issues, we cite the authors’ original numbers. (<https://github.com/Tencent/DOGR/issues>)

References

- [1] Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025. 11
- [2] Anonymous. VFDR-Bench: A multi-lingual, multi-domain benchmark for visually faithful document reconstruction, 2025. Concurrent work under review. Available in supplementary. 9, 14, 46

- [3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 27
- [4] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 27
- [5] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024. 27
- [6] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Naveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 23
- [7] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Naveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 27
- [8] D. Crockford. Rfc 4627: The application/json media type for javascript object notation (json), 2006. 17
- [9] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. 21, 26
- [10] Simone Giovannini, Fabio Coppini, Andrea Gemelli, and Simone Marinai. Boundingdocs: a unified dataset for document question answering with spatial annotations. *arXiv preprint arXiv:2501.03403*, 2025. 5, 6, 27, 28, 46
- [11] Google Cloud. Bounding box detection | generative ai on vertex ai. <https://docs.cloud.google.com/vertex-ai/generative-ai/docs/bounding-box-detection>, 2025. Accessed: 2025-11-18. 23
- [12] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. 17
- [13] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 22, 27
- [14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 5
- [15] Jovana Kondic, Pengyuan Li, Dhiraj Joshi, Zexue He, Shafiq Abedin, Jennifer Sun, Ben Wiesel, Eli Schwartz, Ahmed Nassar, Bo Wu, et al. Chartgen: Scaling chart understanding via code-guided synthetic chart generation. *arXiv preprint arXiv:2507.19492*, 2025. 28
- [16] Stefan Larson, Gordon Lim, and Kevin Leach. On evaluation of document classification with RVL-CDIP. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2665–2678, Dubrovnik, Croatia, 2023. Association for Computational Linguistics. 28
- [17] Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. Llm-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*, 2024. 27
- [18] Mengchen Liu, Qixiu Li, Dongdong Chen, Dong Chen, Jianmin Bao, and Yunsheng Li. Synchart: Synthesizing charts from language models. *arXiv preprint arXiv:2409.16517*, 2024. 28
- [19] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore, 2023. Association for Computational Linguistics. 25, 27, 29, 46
- [20] Shangbang Long, Siyang Qin, Dmitry Panteleev, Alessandro Bissacco, Yasuhisa Fujii, and Michalis Raptis. Icdar 2023 competition on hierarchical text detection and recognition. In *International Conference on Document Analysis and Recognition*, pages 483–497. Springer, 2023. 28
- [21] Tengchao Lv, Yupan Huang, Jingye Chen, Lei Cui, Shuming Ma, Yaoyao Chang, Shaohan Huang, Wenhui Wang, Li Dong, Weiyao Luo, et al. Kosmos-2.5: A multimodal literate model. *arXiv preprint arXiv:2309.11419*, 2023. 24
- [22] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021. 28
- [23] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022. 28
- [24] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016. 4
- [25] Mindee. doctr: Document text recognition. <https://github.com/mindee/doctr>, 2021. 14
- [26] OpenAI. Introducing gpt-5. <https://openai.com/index/introducing-gpt-5/>, 2025. Accessed: 2025-10-25. 22
- [27] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, 2002. Association for Computational Linguistics. 25
- [28] Chris Parmer and the Plotly Contributors. Plotly: An open-source interactive data visualization library for python, 2025. Accessed: 2025-11-20. 17

- [29] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, 2016. Association for Computational Linguistics. 25
- [30] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13009–13018, 2024. 46
- [31] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986. 4
- [32] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019. 25
- [33] Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. Visualmrc: Machine reading comprehension on document images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13878–13888, 2021. 28
- [34] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. 27
- [35] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. 23
- [36] Qwen Team. Qwen3 technical report, 2025. 5, 24, 27
- [37] Tony Cheng Tong, Sirui He, Zhiwen Shao, and Dit-Yan Yeung. G-veval: A versatile metric for evaluating image and video captions using gpt-4o. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7419–7427, 2025. 27
- [38] Jacob VanderPlas, Brian Granger, Jeffrey Heer, Dominik Moritz, Kanit Wongsuphasawat, Arvind Satyanarayan, Eitan Lees, Ilia Timofeev, Ben Welsh, and Scott Sievert. Altair: interactive statistical visualizations for python. *Journal of open source software*, 3(32):1057, 2018. 17
- [39] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 27
- [40] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025. 23, 27
- [41] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021. 17
- [42] Kota Yamaguchi. Canvasvae: Learning to generate vector graphic documents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5481–5489, 2021. 28
- [43] Senqiao Yang, Tianyuan Qu, Xin Lai, Zhuotao Tian, Bohao Peng, Shu Liu, and Jiaya Jia. Lisa++: An improved baseline for reasoning segmentation with large language model. *arXiv preprint arXiv:2312.17240*, 2023. 46
- [44] Yue Yang, Ajay Patel, Matt Deitke, Tanmay Gupta, Luca Weihs, Andrew Head, Mark Yatskar, Chris Callison-Burch, Ranjay Krishna, Aniruddha Kembhavi, et al. Scaling text-rich image understanding via code-guided synthetic multimodal data generation. *arXiv preprint arXiv:2502.14846*, 2025. 28
- [45] Maoyuan Ye, Jing Zhang, Juhua Liu, Chenyu Liu, Baocai Yin, Cong Liu, Bo Du, and Dacheng Tao. Hi-sam: Marrying segment anything model for hierarchical text segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 14
- [46] S. Zhao et al. movie_posters_100k_controlnet. https://huggingface.co/datasets/stzhao/movie_posters_100k_controlnet, 2025. Accessed: 2025-8-12. 28
- [47] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023. 21, 27
- [48] Yinan Zhou, Yuxin Chen, Haokun Lin, Yichen Wu, Shuyu Yang, Zhongang Qi, Chen Ma, and Li Zhu. Dogr: Towards versatile visual document grounding and referring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3596–3606, 2025. 5, 6, 27, 28, 46
- [49] Yinan Zhou, Yuxin Chen, Haokun Lin, Yichen Wu, Shuyu Yang, Zhongang Qi, Chen Ma, Li Zhu, and Ying Shan. Dogr: Towards versatile visual document grounding and referring. GitHub repository, <https://github.com/Tencent/DOGR>, 2025. Accessed: 2025-11-07. 46
- [50] Fengbin Zhu, Ziyang Liu, Xiang Yao Ng, Haohui Wu, Wenjie Wang, Fuli Feng, Chao Wang, Huanbo Luan, and Tat Seng Chua. Mmdocbench: Benchmarking large vision-language models for fine-grained visual document understanding. *arXiv preprint arXiv:2410.21311*, 2024. 28, 46