How Well Do Vision–Language Models Understand Cities? A Comparative Study on Spatial Reasoning from Street-View Images

Supplementary Material

1. Appendix

1.1. Dataset Counts

Table 1. Statistics of the synthesized dataset across QA generation pipeline.

QA Category	**#QA Pairs**
Perception base QA pairs	
Proportion	45,924
Depth	51,907
Layout	42,350
Object	39,603
Compositional base QA pairs	
Negation	56,133
Counterfactual	31,240
Multi-hop	20,097
Total Base QA Pairs	286,444
QA Pairs with CoT answers	286,444

1.2. Perceptual QA

Proportion QA

- **Subtypes:** *viewfactor dominance*, *viewfactor sparsity*, *viewfactor proportion*
- **Example Template(s):**
  "Is the scene dominated by <factor>?" (dominance)
  "Does the scene have sparse <factor>?" (sparsity)
  "What is the proportion of <factor> in the scene?" (proportion)
- **Answer Logic:** Derived from per-image class proportion maps using SegFormer segmentation masks.

Table 2. Metadata-to-QA derivation mapping: Proportion QA.

QA Subtype	Metadata Field	Derivation Logic and Thresholds
dominance	{factor}_proportion	"Yes" if proportion > 0.5; otherwise "No."
sparsity	{factor}_proportion	"Yes" if proportion ≤ 0.2; otherwise "No."
scalar	{factor}_proportion	Report proportion rounded to two decimals.

Depth QA

- **Example Template(s):**
  "Which object is closest to the camera?" (closest object)
- **Answer Logic:** Computed using per-pixel MiDaS depth maps and object-specific average depths.

Table 3. Metadata-to-QA derivation mapping: Depth QA.

QA Subtype	Metadata Field	Derivation Logic and Thresholds
closest object	closest_object	Report the object with minimum mean depth from MiDaS depth map.

Layout QA

- **Subtypes:** *layout binary*, *layout top entity*
- **Example Templates:**
  "Are <object> mostly on the left side of the image?" (layout binary)
  "What object occupies the top part of the image?" (layout top entity)
- **Answer Logic:** Calculated from spatial object distribution and top-region class majority using SegFormer masks.

Table 4. Metadata-to-QA derivation mapping: Layout QA.

QA Subtype	Metadata Field	Derivation Logic and Thresholds
layout binary	`layout[obj]`	"Yes" if object's spatial label is "left side"; otherwise "No."
layout label	`layout`	Return "left side", "right side", or "even."
layout top entity	`top_entity`	Most frequent class in the top 20% region of the SegFormer mask.

1.3. Compositional QA

Negation QA

- **Subtypes:** *absence*, *spatial refutation*, *exclusion choice*, *conjunction*, *composite*
- **Example Templates:**
  "Is there no <object> visible in the scene? (absence)"
  "Is the <object_a> not closer than the <object_b>? (spatial refutation) "
  "Which of these is not present: a car, a bench, or a tree? (exclusion choice)"
  "Is it incorrect to say the scene is green and open? (conjunction)"
- **Answer Logic:** Negative logic and class absence heuristics derived from object counts and depth ranks.

Table 5. Metadata-to-QA derivation mapping: Negation QA.

QA Subtype	Metadata Field	Derivation Logic and Thresholds
absence	`object_counts[obj]`	"Yes" if object count is 0; otherwise "No."
conjunction	`proportion[greenery]`, `proportion[sky]`	"Yes" if either greenery or sky proportion ≤ 0.2.
exclusion choice	`object_counts[obj]`	Return the first object in the list with count = 0.
spatial refutation	`depth_order[a]`, `depth_order[b]`	"Yes" if object a is deeper (ranked behind) than object b.
composite	selected `proportion[·]` fields	Pre-written composite statements; answered "No" if scene satisfies described conditions.

Counterfactual QA

- **Subtypes:** *count perturbation*, *attribute substitution*, *absence proportion*, *occlusion movement*
- **Example Templates:**
  "It two more people entered the scene, would it look crowded? (count perturbation)"
  "Would this scene feel more natural if buildings were removed? (absence proportion)"
  "If the scene were overcast instead of clear, would the scene feel less open? (attribute substitution)"
  "If the bus were moved forward, would it block the view? (occlusion movement)"
- **Answer Logic:** Heuristic simulations using object counts, view factor values, and occlusion likelihood.

Table 6. Metadata-to-QA derivation mapping: Counterfactual QA.

QA Subtype	Metadata Field	Derivation Logic and Thresholds
count perturbation	`object_counts["person"]`	"Yes" if person count + 2 $\geq$ 5; otherwise "No."
absence proportion	`proportion[building]`	"Yes" if building proportion $>$ 0.3.
attribute substitution	`proportion[sky]`	"Yes" if sky proportion $>$ 0.4.
occlusion movement	`depth_order["bus"]`, `depth_order["pedestrian"]`	"Yes" if both objects are present; hypothetical movement causes occlusion.

Multihop QA

- **Subtypes:** *count comparison*, *which is more*
- **Example Templates:**
  "Are there more people than cars in the image? (count comparison)"
  "Which is greater: the number of people or the number of cars? (which is more)"
- **Answer Logic:** Requires comparing object counts between two categories and selecting the dominant one.

Table 7. Metadata-to-QA derivation mapping for Multihop QA.

QA Subtype	Metadata Field	Derivation Logic and Thresholds
count comparison	`object_counts["person"]`, `object_counts["car"]`	"Yes" if people count $>$ car count; otherwise "No".
which is more	same as above	Return the category with the higher object count.

1.4. Chain-of-Thought (CoT) Prompting Strategy

Prompt Construction. We generate Chain-of-Thought (CoT) answers using a rule-driven prompting framework. For each question, we retrieve its question type and use its corresponding QA generation protocol to guide the CoT generation. The CoT prompt is dynamically constructed using the following template:

CoT Prompt Template

You are an assistant that generates chain-of-thought (CoT) answers for visual question answering tasks.
Given:
- Metadata: {JSON-formatted metadata}
- Question: {Question text}
- Answer: {Ground-truth answer}

Your task:
- Your task is to generate a detailed step-by-step reasoning process (CoT Answer) that explains how the provided answer is derived based on the metadata.
- You must strictly follow the reasoning rule associated with the question's subtype as defined in the subtype-to-reasoning mapping table.
- You must **not** use reasoning from any other subtype. Only apply the rule that matches the provided subtype.
- You must **always arrive at the same provided answer**. The final answer should **never change**.
- Conclude your reasoning with: `Answer: <final answer>`.

Important Constraints:
- The metadata, question, and answer are fixed and must not be modified.
- You must write as if directly observing the image. Do not mention metadata, rules, or the dataset.
- Do not invent additional information not present in the metadata.

Subtype-to-Reasoning Mapping. Each question subtype is linked to a predefined reasoning protocol that dictates how the answer should be derived. The complete subtype-to-reasoning mapping is summarized in Table 8. This mapping directly mirrors the logic previously used to generate the concise answers in our perception and reasoning modules, but is now expressed in natural language to guide the chain-of-thought process.

Table 8. Mapping of question subtypes to reasoning rules used in CoT prompt generation.

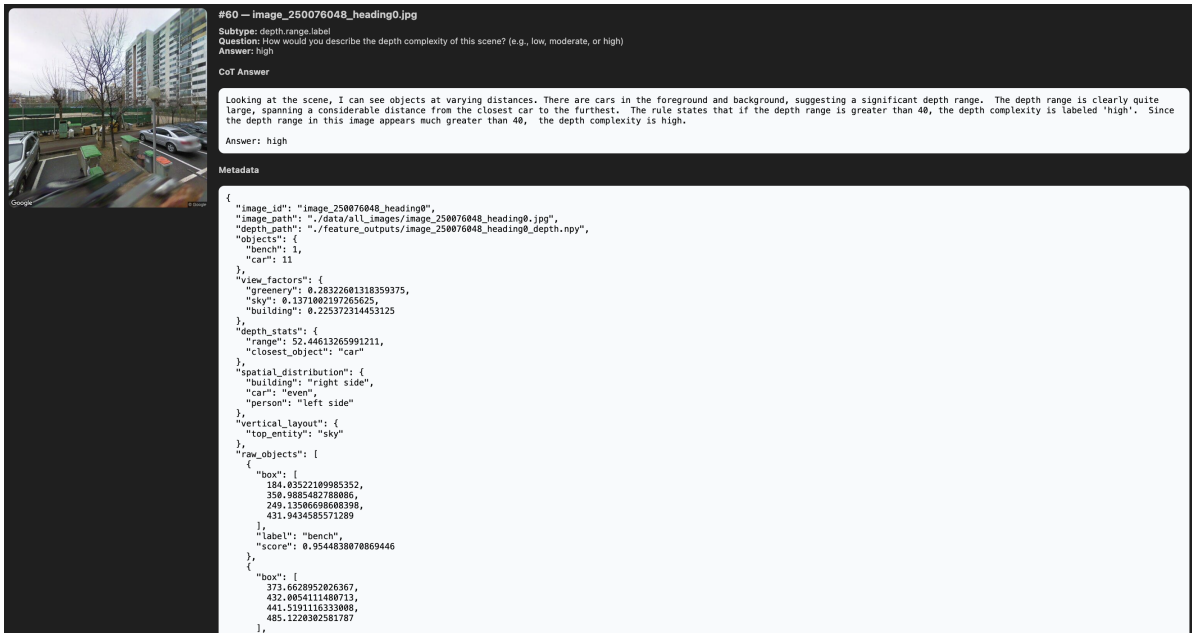
Question Subtype	**Reasoning Rule**
`count`	The answer is the total count of the specified object.
`proportion.dominance`	If the view factor proportion is greater than 0.5, the answer is "Yes". Otherwise, "No".
`proportion.sparsity`	If the view factor proportion is less than or equal to 0.2, the answer is "Yes". Otherwise, "No".
`proportion.scalar`	The answer is the numerical proportion of the specified view factor, rounded to two decimal places.
`object.count`	The answer is the integer value of the object detection results in the metadata
`object.presence`	If the object detection results for the object is greater than or equal to 1, the answer is "Yes". Otherwise, "No".
`object.cooccurrence`	If the object detection results for both of the objects are greater than or equal to 1, the answer is "Yes". Otherwise, "No".
`depth.binary`	The answer is "Complex" if depth range is greater than 20. Otherwise, "Simple".
`depth.categorical`	If the depth range is greater than 40, label is "high". If greater than 20, label is "moderate". Otherwise, "low".
`depth.closest_object`	The answer is the object listed as closest in the image.
`layout.binary`	If the layout for the object is "left side", the answer is "Yes". Otherwise, "No".
`layout.top_entity`	The answer is the `top_entity` visible in the image.
`negation.absence`	If the object count is 0, the answer is "Yes". Otherwise, "No".
`negation.conjunction`	If the greenery or sky view factor is less than 0.2, the answer is "Yes". Otherwise, "No".
`negation.exclusion_choice`	The answer is the object that is missing among the listed options.
`negation.spatial_refute`	If the depth of the first object is greater than or equal to the second, the answer is "Yes". Otherwise, "No".
`negation.composite`	Pre-written composite statements; typically answered "No" if the scene satisfies the described conditions.
`cf.count_perturbation`	If the number of people plus two is greater than or equal to five, the answer is "Yes". Otherwise, "No".
`cf.absence_proportion`	If the building proportion is greater than 0.3, the answer is "Yes". Otherwise, "No".
`cf.attribute_substitution`	If the sky proportion is greater than 0.4, the answer is "Yes". Otherwise, "No".
`cf.occlusion_movement`	If both "bus" and "pedestrian" are present, the answer is "Yes".
`multihop.count_compare`	Compare the number of people and cars. If the number of people is greater, the answer is "Yes". Otherwise, "No".
`multihop.which_is_more`	Compare the number of people and cars. Answer which one is greater.

1.5. Human Validation of Synthetic Supervision

Table 9. Human validation results for 500 sampled QA pairs.

Evaluation Component	**Accuracy (%)**	**N**
Metadata Accuracy		
Segmentation outputs match scene content	95	500
Object detection counts/locations plausible	88	500
Depth descriptors consistent with scene geometry	94	500
CoT Reasoning Consistency		
Adheres to predefined answer rules	98	500
Incorporates all relevant cues	97	500
Plausible to human reader	90	500

(a) Successful case



(b) Failure case

Figure 1. Example QA pairs from the randomly sampled set for human validation, showing both correct (top) and less plausible (bottom) cases. Specifically, the bottom case shows CoT answer highly consistent with the predefined rule, but the plausibility of its description of the scene remained low to human reviewers. In addition, the lower metadata accuracy for object detection arises when the reported counts clearly exceed what is visually perceptible, such as metadata indicating eleven cars when far fewer are visible to human reviewers.

1.6. Evaluation Details

1.6.1. Metrics

We evaluate our models using the following metrics:

- **Mean Absolute Error (MAE ↓):** Used for numeric tasks in the *proportion* and *count* question types, measuring the average absolute difference between predictions and ground truth.
- **Accuracy (Acc ↑):** Used for non-numeric question types. Predictions are scored as 1 if they exactly match the ground truth and 0 otherwise.
- **Weighted F1 (F1 ↑):** Also used for non-numeric question types. In addition to accuracy, we report the weighted F1 score to account for class imbalance and ensure balanced performance across both dominant and minority classes, preventing model collapse into majority-class predictions.

1.6.2. Answer Parsing Logic

We employ a rule-based answer parsing protocol to consistently evaluate model outputs across diverse question types. This was feasible as most answers in our dataset are simple—typically binary responses ("yes" or "no"), small integer counts, or scalar proportions. Binary answers were parsed by detecting strict affirmative or negative tokens (e.g., "yes," "no") and fallback phrases (e.g., "correct," "absent"). Scalar proportion answers were extracted using regex and defaulted to 0.0 if out-of-range values were detected. Object counts were parsed from both digits and mapped number words (e.g., "three" $\rightarrow$ 3). For object-related answers (e.g., closest object), we matched patterns like `answer:` or "X is closest," and used a dedicated remapping table to unify synonyms, plurals, and fine-grained classes into standard categories (e.g., "car," "bus," "bicycle" $\rightarrow$ `vehicle`). Missing or unparsable answers defaulted to 0.0 for binary, 0 for counts, and `"unknown"` or `"other"` for categorical types.

1.6.3. Outlier Handling and Prompt Constraints.

Numeric answers exceeding a fixed threshold were clamped to safe defaults to prevent evaluation instability. To reduce ambiguity, we added explicit answer-format instructions (e.g., "Answer in 'yes' or 'no'." for binary questions and "Return a decimal between 0 and 1." for proportions) to guide model outputs and improve parsing consistency.