Supplementary Material for Robo2VLM: Visual Question Answering from Large-Scale In-the-Wild Robot Manipulation Datasets

Kaiyuan Chen[1,*] **Shuangyu Xie**[1,*] **Zehan Ma**[1] **Pannag R Sanketi**[2] **Ken Goldberg**[1]
[1]University of California, Berkeley Google DeepMind [*]Equal contribution
{kych, syxie, zehanma, goldberg}@berkeley.edu
https://huggingface.co/datasets/keplerccc/Robo2VLM-1

Contents

Preprint.

A Broader Impact

The development of Robo2VLM and Robo2VLM-1 aims to accelerate progress in robotic manipulation by providing a robust framework for evaluating and improving Vision-Language Models. Positive societal impacts are significant. More capable robots, enhanced by VLMs rigorously tested on such benchmarks, can revolutionize various sectors. In manufacturing, they can lead to more efficient, flexible, and safer production lines by undertaking complex assembly or hazardous material handling. In healthcare, advanced robotic assistants could support surgeons with greater precision, provide personalized care for the elderly or individuals with disabilities, thereby improving their quality of life and independence, and assist in laboratory automation for faster medical research. For domestic tasks, robots could alleviate household burdens, freeing up human time for more creative or relational pursuits. Beyond these, such advancements can contribute to safer work environments by automating dangerous jobs in construction, mining, or disaster response, and even aid in environmental conservation efforts through automated monitoring and intervention. The increased productivity and innovation spurred by these technologies could lead to economic growth and the creation of new job categories focused on designing, maintaining, and overseeing these intelligent systems. However, it is important to consider potential negative societal impacts. As VLMs become more powerful through evaluation on such benchmarks, there's a risk of misuse if these capabilities are applied to autonomous systems without appropriate safeguards, potentially leading to unintended actions or job displacement in certain sectors. For example, if the underlying trajectory data in Robo2VLM inadvertently contains biases (e.g., related to specific environments, objects, or human demonstrators), models trained or evaluated on Robo2VLM-1 might perpetuate or amplify these biases. Future work should actively consider methods to detect and mitigate such biases in the dataset and the models. Furthermore, while the goal is to advance AI for beneficial applications, any significant improvement in generative or understanding capabilities of models could, in principle, be adapted for unintended purposes. Therefore, ongoing discussion and development of ethical guidelines and safety protocols are crucial as VLM capabilities advance in robotics and other fields.

B Question Analysis

The complete dataset can be found in the huggingface website, https://huggingface.co/datasets/keplerccc/Robo2VLM-1. We provide representative examples to show the diversity and quality of the dataset. Each VQA contains one/multiple images showing the robot current position and the scene, a language description question, and multiple choices as candidate answer.

B.1 Example Questions from Different Tasks

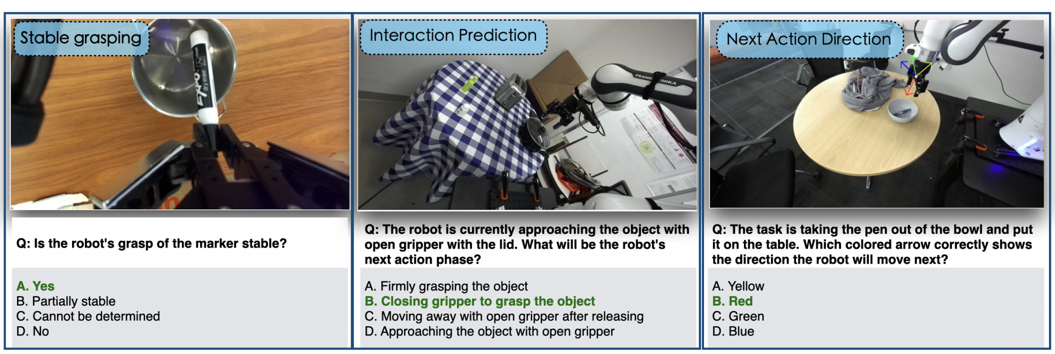


Figure 1: **Example VQAs**. Each panel illustrates a distinct category of visual question answering grounded in real robot interactions.

The examples in Figs. 1,2 highlight the diversity and complexity of visual question answering (VQA) tasks grounded in real-world robotic manipulation. Each question may be associated with multiple images, which can originate from different phases of the manipulation sequence or from distinct camera viewpoints. This design reflects the inherently temporal and multi-perspective nature of robotic tasks, requiring models to reason over a sequence of actions or fuse complementary observations. The questions span reasoning types such as goal configuration prediction, task outcome evaluation, grasp stability assessment, and interaction phase forecasting. These diverse formats challenge models to integrate spatial understanding, temporal progression, and multimodal cues, making the dataset a rigorous benchmark for evaluating the task-level reasoning capabilities of vision-language models in robotics.

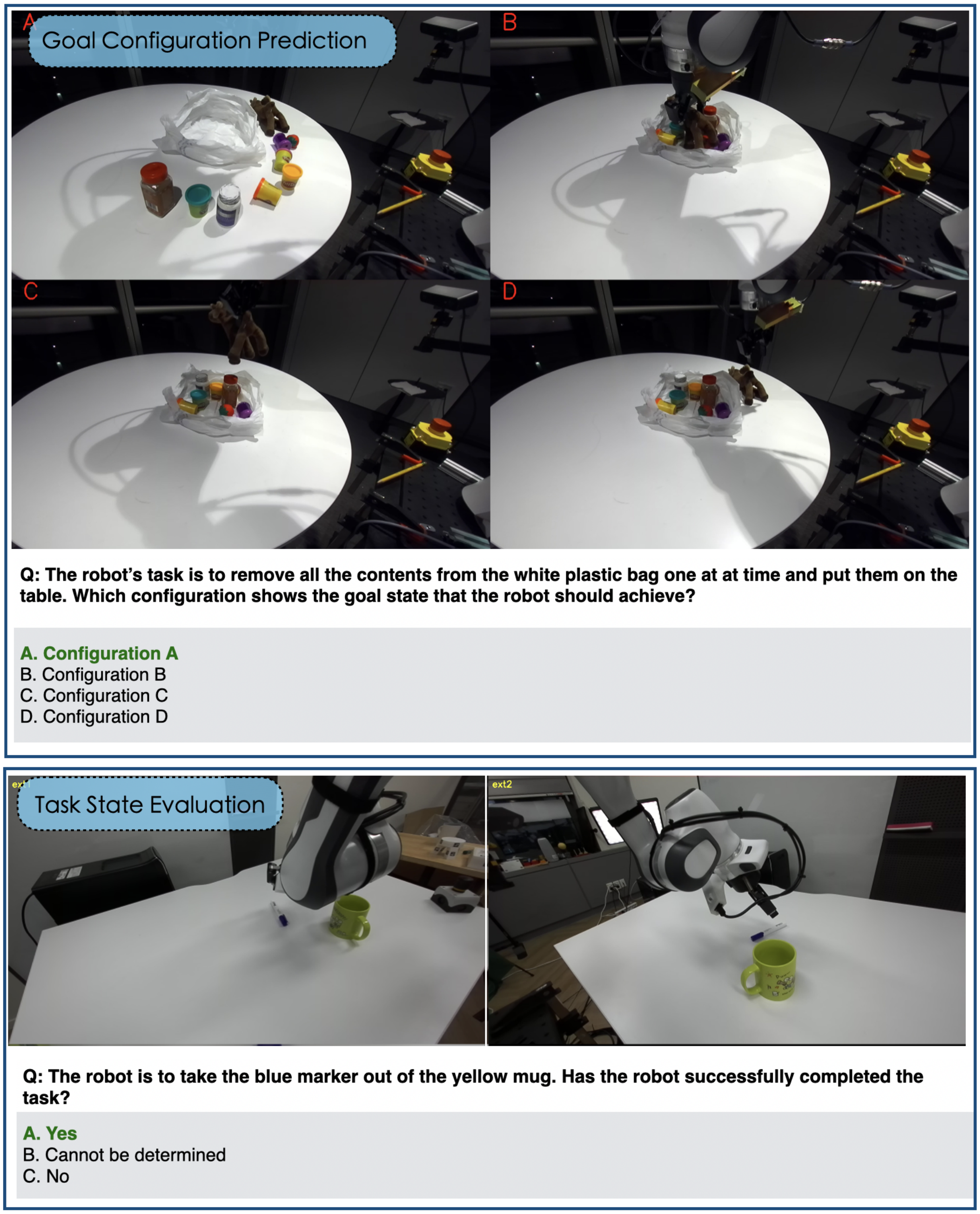
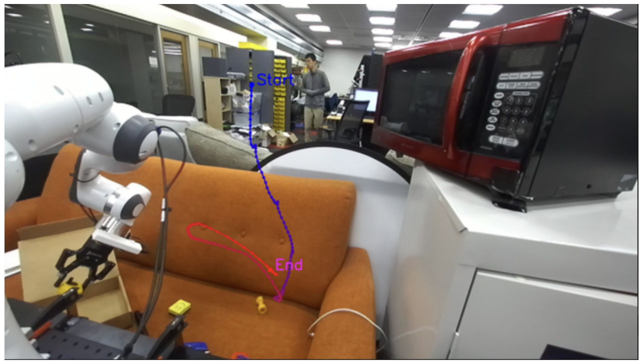


Figure 2: **Example VQAs using with mutlple images.** Each panel presents a unique type of VQA grounded in real-world robot trajectories. Goal Configuration Prediction (top) asks which scene configuration matches the task goal. Task State Evaluation (bottom) queries whether the robot has successfully completed a specified action. These examples demonstrate the need for multimodal reasoning over visual observations and task context. Correct answers are highlighted in green.

B.2 Challenging Questions

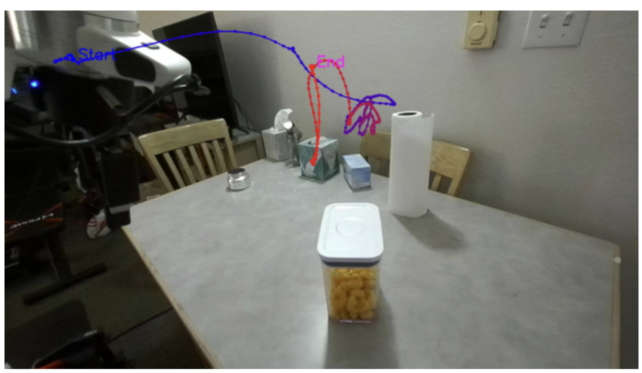
The following figures illustrate several visual question answering (VQA) tasks conducted using robotic trajectories. Each figure presents a unique scenario where human expertise was used to validate the correctness of robotic actions or spatial understanding based on visual inspection. These are questions human experts consider challenging but answered correctly. We will introduce more details for human expert instruction and feedback in Sec. F.



Question: Which language instruction best describes the robot's trajectory shown in the image? *[Pick up the black from the drawer, Drop the box into the shelf, put the yellow and black object in the box, Align the black with the table, Move the box to the floor]*

- **Correct Answer:** Put the yellow and black object in the box
- **Expert Rationale:** The trajectory isn't directly at the objects, but the gripper position suggested interaction with the box. This reasoning led me to identify the correct choice clearly.

Figure 3: Identifying the appropriate language instruction corresponding to a robot trajectory involving interaction with a yellow and black object.



Question: Which language instruction best describes the robot's trajectory shown in the image? *[Move the container to the tray, Push the pen towards the bin, Align the box with the drawer, open the container lid, Lift the cup upward]*

- **Correct Answer:** Open the container lid
- **Expert Rationale:** Answers involving absent objects (pen, cup) were quickly eliminated. The trajectory clearly aligned with the container, making the correct answer straightforward.

Figure 4: Robot trajectory clearly aligned with opening a container lid, excluding irrelevant options involving absent items.

Question: The robot task is to move the spoon. Which colored arrow shows the most likely direction the robot will move next?
[Yellow, Purple, Blue, Green, Red]

- **Correct Answer:** Red
- **Expert Rationale:** Initially unclear about the spoon's exact position, I carefully inspected to confirm the gripper already grasped the spoon, identifying the red arrow direction correctly.

Figure 5: Discerning the direction of spoon movement based on visual cues, highlighting careful visual analysis.

Question: Is the robot's grasp of the sponge stable?
[Yes, No, Cannot be determined, Partially stable]

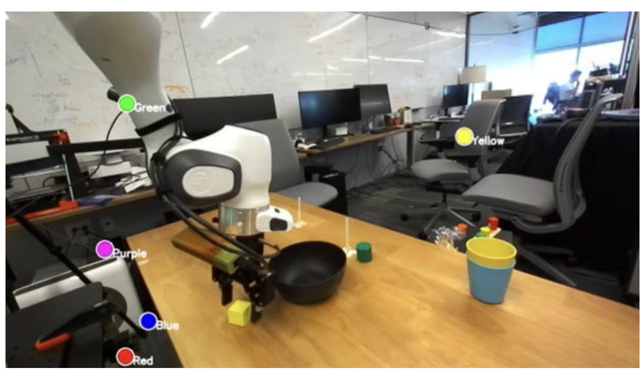
- **Correct Answer:** No
- **Expert Rationale:** At first glance, the grip seemed stable, but closer examination revealed the grasp was inadequate on the sponge's edge, confirming instability.

Figure 6: Evaluating the stability of a robotic grasp on a sponge, emphasizing close visual inspection to determine grasp quality.

Question: In the left image (ext1 camera), a red dot is marked. Which point in the right image (ext2 camera) corresponds closest to this dot?
[A, B, C, D]

- **Correct Answer:** D
- **Expert Rationale:** Distinguishing between similarly close points (A and B) required careful analysis. By comparing unique features (such as the wrist camera and the joint's white part), the correct point became evident.

Figure 7: Identifying corresponding points between two camera views, requiring detailed analysis of visual similarities.



Question: In the ext2 camera image, which colored point is closest to the camera?
[A, B, C, D, E]

- **Correct Answer:** C
- **Expert Rationale:** Initially, I assumed the blue point was on the table, but a second look revealed it was on the floor, indicating clearly that point C was closest to the camera.

Figure 8: Illustrating depth perception from camera viewpoints and correcting initial visual assumptions upon closer inspection.

C VLM Output Examples

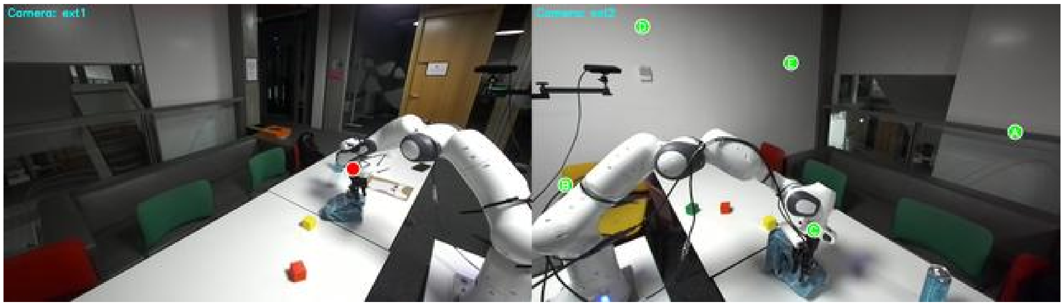


Figure 9: **Question**: In the left image (ext1 camera), a red dot is marked. Which point is the closest point in the right image (ext2 camera) corresponding to the same 3D location? The correct answer is **D. point C**.

Model: `llava-hf/llava-1.5-7b-hf`

Prediction: The closest point in the right image (ext2 camera) corresponding to the same 3D location as the red dot in the left image (ext1 camera) is **point B**.

Step-by-step explanation:

- In the left image, the red dot is marked at the top-left corner.
- In the right image, point B is also located in the top-left corner.
- The red dot's position is consistent across both views.
- Thus, point B corresponds to the same 3D location as the red dot.

Final Answer: `B` ✗

Model: `Qwen/Qwen2.5-VL-7B-Instruct`

Prediction: To determine the corresponding point in the right image, we analyze spatial relationships in both views.

Step-by-step explanation:

- The red dot in the left image is on the table, slightly left of center and near the bottom.
- In the right image, a small red object appears in the same relative position.
- Candidate analysis:
  - A: Point B — too far from the red object.
  - B: Point A — located on the wall.
  - C: Point E — not aligned spatially.
  - D: Point C — best matches the red dot position.
  - E: Point D — farther than Point C.

Final Answer: `D` ✓

Model: `meta-llama/Llama-3.2-11B-Vision-Instruct`

Prediction: Point **E** in the right image corresponds to the same 3D location as the red dot in the left image.

Step-by-step explanation:

- Locate the red dot in the left image (ext1 camera) — it's on the table in front of the robot.
- In the right image (ext2 camera), observe all labeled points.
- Point E is spatially aligned with the red dot's 3D location.

Final Answer: `E` ✗

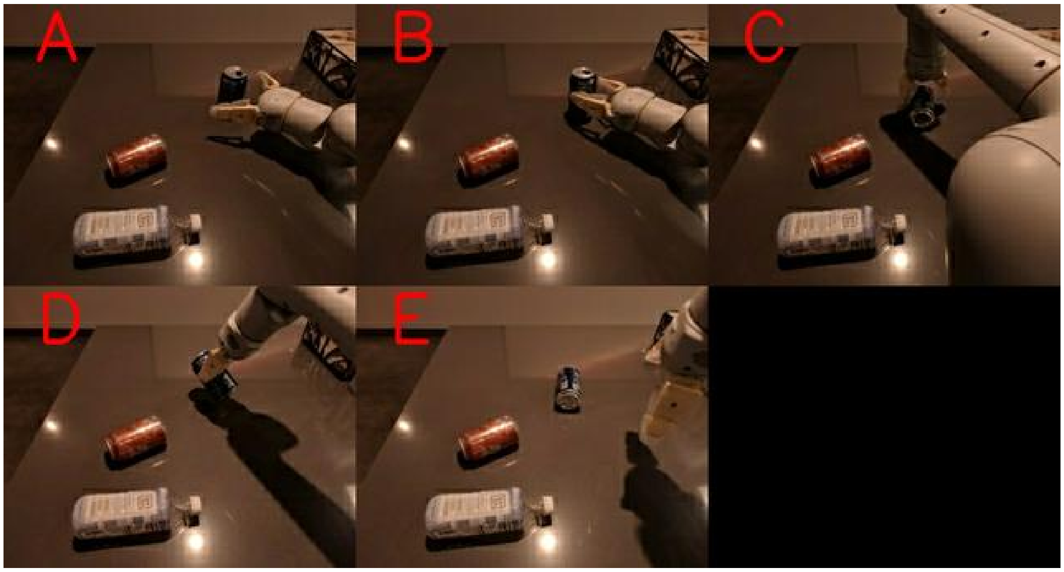


Figure 10: **Question**: The robot's task is to place pepsi can upright. Which configuration shows the goal state that the robot should achieve? ["Configuration D", "Configuration E", "Configuration B", "Configuration C", "Configuration A"]

Model: `llava-hf/llava-v1.6-mistral-7b-hf`

Prediction: To determine the configuration that shows the goal state for the robot, we consider:

- The task requires placing Pepsi cans upright.
- The robot starts with cans on their sides.
- We must identify the configuration where the cans are fully upright.

Analysis of configurations:

- **Configuration A:** One Pepsi can is upright; two are on their sides.
- **Configuration B:** All cans are on their sides.
- **Configuration C:** One can is upright; two are on their sides.
- **Configuration D:** Two cans upright, two on their sides.
- **Configuration E:** All cans are on their sides.

Final Answer: `A` ✗

Model: `Qwen/Qwen2.5-VL-7B-Instruct`

Prediction: To determine which configuration shows the goal state that the robot should achieve, let's analyze the sequence of images step by step:

- **Configuration A:** The Pepsi can is initially lying on its side.
- **Configuration B:** The can has been moved to a vertical position.
- **Configuration C:** The can remains vertical but is shifted slightly to the right.
- **Configuration D:** The can is still transitioning toward an upright position.
- **Configuration E:** The can is upright and appears stable.

Given the goal of achieving a final, stable upright position:

- B and D are transitional.
- C is upright but not final.
- E is upright and stable.

Final Answer: `E` ✓

Model: `meta-llama/Llama-3.2-11B-Vision-Instruct`

Prediction: The task is to place the Pepsi can upright.

- **Configuration A:** The can is already upright, which the model assumes is the goal state.
- **Configurations B–E:** The model states the can is still lying on its side in each.

Conclusion: The model selects Configuration A as the goal state.

Final Answer: `A` ✗

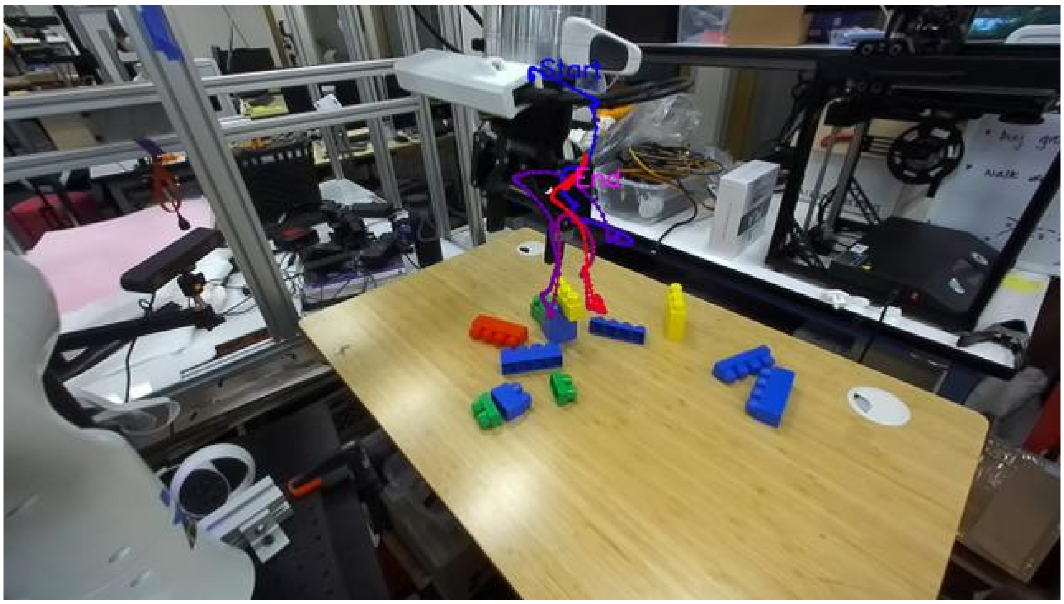


Figure 11: **Question**: The robot's task is to place pepsi can upright. Which configuration shows the goal state that the robot should achieve? ["Configuration D", "Configuration E", "Configuration B", "Configuration C", "Configuration A"]

Model: `llava-hf/llava-v1.6-mistral-7b-hf`

Prediction: The model notes uncertainty due to limited visual context and evaluates each option:

- A – Rejected: no tower visible in the image.
- B – Possible but lacks visual confirmation of gripper state.
- C – Rejected: claimed to be irrelevant (note: this is the correct answer).
- D – Selected: plausible if the robot is about to pick up a block.
- E – Rejected: no tower present.

Based on this reasoning, the model selects: **Final Answer:** `D` ✗

Model: `Qwen/Qwen2.5-VL-7B-Instruct`

Prediction: The model analyzes the trajectory and visual setup:

- The robot's arm is extended toward a block on the table.
- The movement suggests an upcoming grasp action.
- The gripper is not yet in contact with the object.

Reasoning:

- A – No rotation or tower is observed.
- D – The object is not on the floor.
- E – No tray is visible.

- C – Incorrectly dismissed, despite covering the multi-step nature of the trajectory.

The model concludes the most likely intent is to grab the block:
Final Answer: `B` ✗

Model: `meta-llama/Llama-3.2-11B-Vision-Instruct`

Prediction: The robot's trajectory involves multiple steps:

- Rotating the tower clockwise
- Grabbing a block with the gripper
- Picking up the block from the floor
- Dropping the tower into a tray

Each option in the list describes only a partial action:

- A – Rotation only
- B – Grabbing only
- D – Picking up only
- E – Dropping only

Since no single instruction fully describes the trajectory, the model selects: **Answer:** `C. None of the above` ✓

D Prompt Design for VQA Inference and Verification

This section presents the prompt design strategies used for three key VQA inference settings: answer verification, zero-shot inference, and chain-of-thought (CoT) reasoning. The verifier prompt guides a model to extract and isolate the correct multiple-choice answer from a generated explanation, ensuring alignment between reasoning and final answer format. The zero-shot prompt enforces concise behavior by instructing the model to output only the letter corresponding to the correct answer without additional reasoning. In contrast, the CoT prompt encourages step-by-step reasoning before concluding with the final answer, enabling the model to explain its decision-making process. Additionally, Table 1 outlines prototype question types used in Robo2VLM.

D.1 Prompt for Verifier

The verifier prompt is used to post-process model-generated answers that contain free-form text, such as in CoT or long-form reasoning outputs. It instructs the model (or another lightweight parser model) to extract the final answer option—typically a letter (A, B, C, D, or E)—from the full response. This prompt plays a critical role in decoupling reasoning quality from answer accuracy, allowing us to evaluate whether the model reaches a correct conclusion after potentially verbose reasoning. The design includes an illustrative example to make the extraction instruction explicit and reduce hallucination of unexpected formats.

Example: Verifier Prompt

Instructions: Please read the example below and extract the final answer from the model response.
Hint: Your output should be a single letter (e.g., A, B, C, or D) indicating the correct option.

Question: What fraction of the shape is blue?
Choices: (A) 3/11 (B) 8/11 (C) 6/11 (D) 3/5

Model response: The correct answer is (B) 8/11.

Extracted answer: `B`

D.2 Prompt for Zero-Shot

The zero-shot prompt is optimized for direct evaluation of pretrained VLMs without any in-context demonstrations. It instructs the model to select one option from a multiple-choice question using only the corresponding letter. The prompt avoids any reasoning cues or explanations, forcing the model to rely entirely on its pretrained visual and language priors. This prompt setting allows us to assess the model's default grounding and answer formulation capabilities, free from inductive biases introduced by reasoning scaffolds.

Prompt: Zero-Shot Inference

Instructions: Answer the following multiple-choice question by selecting the correct option letter only.
Hint: Do not include any explanation—your response should only contain one of the letters: A, B, C, D, or E.
{Question}

The {question} can be found in Table 1.

D.3 Prompt for Chain-of-Thoughts

To improve performance on questions that benefit from intermediate reasoning steps (e.g., spatial inference, task planning, or temporal prediction), we adopt a CoT prompt that encourages step-by-step explanation before committing to a final answer. The CoT prompt explicitly requests both reasoning and a conclusive answer in a standard format, helping the model avoid trailing off or omitting a definitive choice. This setting is particularly useful for analyzing the internal decision-making process of large language models in complex manipulation scenarios.

Prompt: Chain-of-Thought Reasoning

Instructions: Answer the following multiple-choice question by reasoning step by step. Show your work for each step before concluding.
Hint: After completing your reasoning, output only the final answer option letter (A, B, C, D, or E) at the end.
{Question}

The {question} can be found in Table 1.

Table 1: Question Prompt Templates for VQA Functions

VQA Function	Question Prompt Prototype
robot_gripper_open	Is the robot's gripper open?
object_reachable	Is there any obstacle blocking the robot from reaching {object}?
relative_direction	In the image from {camera} at step {step}, which direction is the {object} relative to the robot's end effector?
relative_depth	In the image from {camera}, which colored point is closest/farthest from the camera?
view_correspondence	In the left image ({camera1}), a red dot is marked. Which point in the right image ({camera2}) corresponds to the same location?
task_success_state	The robot is to {instruction}. Has the robot successfully completed the task?
is_stable_grasp	Is the robot's grasp of the {object} stable?
goal_configuration	The robot's task is to {instruction}. Which configuration shows the goal state?
action_understanding	The robot is tasked to {instruction}. Which phase of the grasp action is shown?
next_action	After {current phase}, what will be the robot's NEXT action phase?
trajectory_understanding	Which language instruction best describes the robot's trajectory shown in the image?
action_direction	Which colored arrow correctly shows the direction the robot will move next?
temporal_sequence	What is the correct sequence of action phases shown in the images?

E Fine-Tuning and Evaluation Details

E.1 Fine-Tuning Details

Model Configuration The model utilized for vision-language tasks is based on `meta-llama/Llama-3.2-11B-Vision`, configured for optimal performance. Key settings include gradient checkpointing with the "unsloth" method, a LoRA (Low-Rank Adaptation) rank of 128, an alpha parameter of 256, and no dropout for LoRA modules. Model fine-tuning is selectively enabled for language layers, attention modules, and MLP modules while keeping vision layers fixed. The maximum sequence length is set to 2048 tokens to accommodate complex vision-language interactions.

Training Setup Training utilizes the dataset `keplerccc/ManipulationVQA-60k` with a dedicated train split and a validation ratio of 5%. Batch size is carefully controlled at 4 samples per device, enhanced by gradient accumulation over 4 steps. The training process involves linear scheduling of the learning rate, starting at , and includes a weight decay of 0.01. The training is configured to run for one epoch with frequent checkpoints every 1000 steps, evaluation intervals at 5000 steps, and logging every 10 steps.

Evaluation Protocol Evaluation is conducted using a maximum of 10,000 test samples, with explicit configuration for generating visualizations and fallback strategies in case of missing test splits. Generation settings include sampling with a temperature of 0.7 and allowance of up to 50 new tokens per generation. The evaluation setup includes assessing both base and fine-tuned model versions, each clearly delineated within the configuration.

Distributed Training and Precision The system leverages distributed training techniques, exploiting high-performance computational resources for scalable training. It utilizes Brain Floating Point (BF16) precision to balance computational efficiency and numerical stability, eschewing FP16 for better performance stability.

E.2 Evaluation Details

Experimental Setup We conducted evaluations using a vision-language model (VLM) pipeline configured specifically for Visual Question Answering (VQA) tasks. The evaluation utilizes the Hugging Face dataset named `keplerccc/ManipulationVQA`, specifically the `test` split, enabling standardized comparisons. To maintain computational efficiency and manage GPU resources effectively, the evaluation employs adaptive batch processing strategies.

Model Configuration The evaluation primarily considers two large-scale multimodal models: `llava-hf/llava-v1.6-34b-hf` and `llava-hf/llava-next-72b-hf`. These models leverage tensor parallelism set to 4, harnessing the full computational power of four A100 GPUs to optimize throughput. The models were initialized with a GPU memory utilization parameter set to 0.9, ensuring efficient memory usage without exceeding GPU capacity.

Prompt and Response Extraction Each evaluation prompt explicitly instructs the models to select from multiple-choice answers (options A, B, C, D, E). Responses are subsequently processed using a secondary extraction model (`meta-llama/Llama-3.2-3B-Instruct`), designed to deterministically extract the selected letter-answer from the models' verbose outputs. This extraction leverages zero-temperature sampling to guarantee reproducibility and consistency across evaluations.

Dataset and Evaluation Metrics The dataset comprises a randomly shuffled subset of test questions, limited by a configurable maximum sample parameter. Accuracy metrics are computed overall and further broken down by tags to provide granular insights into model performance across different question categories. Detailed timing information for responses is recorded to assess computational efficiency, reporting average response times alongside accuracy metrics.

F Human Expert Instruction and Feedback

To improve the quality and answerability of automatically generated questions, we ask a human expert to improve the data generation process. We provided an initial set of 200 question-image pairs generated by the Robo2VLM pipeline to a human expert for review. The expert was instructed to identify unanswerable or ambiguous cases and annotate the reasons, which were then used to iteratively refine the prompt and generation pipeline. The human expert takes two hours to complete the evaluation. We then follow the revised questions to generate the whole dataset.

F.1 Evaluation Protocol

The human expert was asked to assess whether each question could be reliably answered based solely on the visual input and accompanying instruction. For cases deemed unanswerable, the expert selected from predefined failure modes including: (1) insufficient or unclear visual context, (2) ambiguous or underspecified language in the prompt, and (3) other task-specific issues. This structured feedback guided the refinement of question templates, robot state annotations, and visual preprocessing steps.

F.2 Feedback-Driven Refinement of Auto-Curation

Table 2 summarizes the key issues uncovered through human evaluation and the corresponding solutions incorporated into the Robo2VLM pipeline. These challenges fall into four main categories: (i) *Context and Task Definition*, addressing missing goal descriptions and task phase awareness; (ii) *Visual Information and Camera Limitations*, such as limited visibility or poor resolution, which

Table 2: Problems Identified by Human Experts and Corresponding Solutions Implemented in Robo2VLM Pipeline

Problem Category	Implemented Solution
Context and Task Definition	
Image understanding issue without task context	Enhanced question prompt with task context
Lack of goal specificity	Enhanced question prompt with goal descriptions
Assumed implicit knowledge of robotic tasks	Added description of the robot's current phase
Visual Information and Camera Limitations	
Limited wrist camera view and object visibility	Integrated multi-view images
Invisible gripper state from certain angles	Added gripper state verification and filtering
Insufficient image resolution for detailed object identification	Filtered out images with resolution lower than 100×100 pixels
Question Formulation and Consistency	
Ambiguous or complex question phrasing	Standardized linguistic templates
Inconsistent task completion criteria	Unified success state definitions
Redundant or confusing phrasing	Applied phrase filtering and clarity scoring
Conflicting answers across questions for same image	Added consistency validation checks
Category-Specific Issues	
Multiple viewpoints needed for configuration selection	Added multi-angle verification to configuration questions
Spatial reasoning depends on object boundaries and color	Improved spatial questions with object detection and color validation
Direction prediction depends on task goal	Integrated goal-aware motion prediction

were mitigated through multi-view integration and filtering heuristics; (iii) *Question Formulation and Consistency*, where we standardized linguistic structures, unified success criteria, and added consistency validation checks; and (iv) *Category-Specific Issues*, including configuration reasoning, spatial alignment, and directional prediction, which were resolved using goal-aware and multi-perspective analysis. Together, these improvements enhance the reliability, interpretability, and generalization of vision-language evaluations in robotic settings.

G Key dataset statistics

We analyzed a total of 60,000 samples in the dataset. On average, questions are 108.69 characters long, with a median length of 113 characters. The shortest question contains 28 characters, while the longest reaches 378 characters. Each question includes an average of 4.65 answer choices, with most having either 4 or 5 options. The typical choice is 14.22 characters long on average, though lengths vary widely—from as short as 1 character to as long as 271 characters. The combined length of all choices per question averages 66.09 characters, with a median of 44 characters and a range from 5 to 687 characters.

In terms of correct answer distribution, the dataset is relatively balanced among options A to D: 22.03% of correct answers are 'D', 21.86% are 'B', 21.74% are 'C', and 21.53% are 'A'. Option 'E' appears less frequently, making up 12.84% of correct responses.

Regarding image data, the average image width is 520.66 pixels, with a median of 640 pixels, while heights average 292.99 pixels, with a median of 256 pixels. Image widths range from 84 to 640 pixels, and heights from 84 to 480 pixels. The most common image resolutions are 640x360 (39.61%), 320x256 (21.14%), 640x240 (8.46%), 640x180 (5.81%), and 448x224 (4.14%). Across the dataset, there are 19 unique image resolutions.

Table 3: Dataset Statistics Summary for 60,000 Samples

Category	**Metric**	**Value**
Questions	Avg. length (chars)	108.69
	Median length (chars)	113.00
	Min length (chars)	28
	Max length (chars)	378
Choices	Avg. # choices per question	4.65
	Median # choices	5.00
	Min # choices	4
	Max # choices	5
	Avg. length of a choice (chars)	14.22
	Median length of a choice (chars)	6.00
	Min/Max choice length (chars)	1 / 271
Choices (total per question)	Avg. total length (chars)	66.09
	Median total length (chars)	44.00
	Min/Max total length (chars)	5 / 687
Answer Distribution	A	12,918 (21.53%)
	B	13,115 (21.86%)
	C	13,046 (21.74%)
	D	13,216 (22.03%)
	E	7,705 (12.84%)
Image Width (px)	Avg.	520.66
	Median	640.00
	Min/Max	84 / 640
Image Height (px)	Avg.	292.99
	Median	256.00
	Min/Max	84 / 480
Top-5 Resolutions	640x360	23,768 (39.61%)
	320x256	12,683 (21.14%)
	640x240	5,075 (8.46%)
	640x180	3,484 (5.81%)
	448x224	2,482 (4.14%)
	Unique resolutions	19

H Distractor Choice Design

This section outlines the design and evaluation of distractor choices in our VQA dataset, which play a critical role in determining question difficulty and diagnostic value. We begin by examining the impact of introducing a "None of the Above" (NAB%) option, which systematically increases task ambiguity and reduces model performance across the board (Fig. 12). We then detail the principles and heuristics used to generate diverse and context-aware distractors for different question types. These include binary negations, categorical sampling, spatial reasoning perturbations, and content-aware language distractors. Special emphasis is placed on generating plausible incorrect choices that reflect partial knowledge, ambiguity, or visually confusable elements. Finally, we describe how randomized shuffling and probabilistic replacement with NAB options further strengthen the challenge by discouraging rote pattern matching. Together, these strategies enhance the dataset's ability to probe fine-grained reasoning, visual grounding, and robustness to uncertainty in large vision-language models.

H.1 None of the Above Proportion

This section shows experiment of adding ‘None of the Above’ selection Ratio (NAB%) choice increase the difficulty of the dataset and model accuracy decrease for all the models. We show the result in the line plot in Fig. 12.

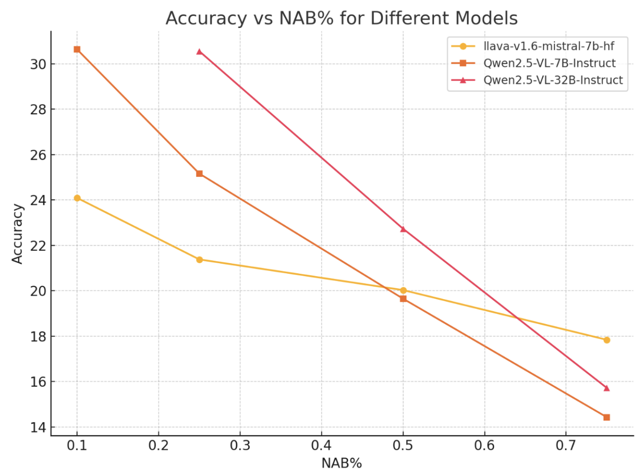


Figure 12: Accuracy vs. ‘None of the Above’ Selection Ratio (NAB%) for Three Vision-Language Models

The plot reveals that all three models experience a decline in accuracy as NAB% increases, indicating reduced confidence or higher prediction difficulty when a greater proportion of questions are considered potentially unanswerable. Qwen2.5-VL-32B-Instruct consistently outperforms the other two models when data is available, achieving the highest accuracy of 30.55% at NAB% = 0.25. Interestingly, the 7B Qwen2.5-VL variant initially performs well (30.63% at NAB% = 0.1) but degrades more sharply than the 32B version. The llava-v1.6-mistral-7b-hf model maintains the lowest accuracy across all NAB% levels, suggesting it is less robust under ambiguity. These trends highlight the importance of model scale and training data in handling tasks with varying uncertainty.

H.2 Distractors

The design of distractor choices is crucial for creating challenging and meaningful Visual Question Answering (VQA) instances. The provided Python codebase employs several strategies to generate plausible yet incorrect options, aiming to test nuanced understanding rather than simple pattern recognition.

Binary and Generic Distractors For questions anticipating a binary response (e.g., Yes/No), the primary distractor is often the direct negation of the correct answer. This is evident in functions like `vqa_robot_gripper_open` and `vqa_object_reachable`. These are typically supplemented by generic distractors such as “Cannot be determined” or context-specific but still general alternatives like “Partially open” or “Partially reachable”. The `_validate` method ensures that binary questions have exactly four choices, accommodating these patterns.

Categorical and Permutation-Based Distractors Many VQA generation functions define a set of possible categories and select distractors from those not matching the correct answer. **Relative Directions:** In `vqa_relative_direction`, a comprehensive list of possible spatial relations (e.g., “Upper Left”, “Lower Forward”) is generated. After identifying the correct direction, incorrect choices are drawn from this list, with a preference for those sharing some component (e.g., the same vertical component) with the correct answer to increase plausibility. **Action Phases:** For `vqa_action_understanding` and `vqa_next_action`, distractors are chosen from a defined set of robot action phase descriptions (e.g., “Approaching the object with open gripper”, “Firmly grasping the object”). The incorrect choices are the descriptions of other valid phases. **Temporal Sequences:** `vqa_temporal_sequence` generates distractors by creating incorrect orderings (permutations) of the actual sequence of events or phases if the question is about the sequence itself. **Color/Label-Based Choices:** In `vqa_relative_depth` and `generate_action_direction_selection_vqa`, distinct colors (e.g., "Red", "Green", "Blue") are assigned to different points or arrows in the image. The choices are then these color names, with one corresponding to the correct visual marker. Similarly, `vqa_multi_view_correspondence` uses letter labels ("A", "B", "C", "D", "E") for choices corresponding to marked points.

Spatially Derived Distractors For tasks involving spatial reasoning, distractors are often generated to be distinct in the image space. In `vqa_multi_view_correspondence`, distractor points are generated in different quadrants of the image from the correct corresponding point, ensuring a minimum pixel distance from each other and the correct point. `generate_action_direction_selection_vqa` creates incorrect directional arrows by ensuring their angles are meaningfully different from the correct action direction, with a minimum angular separation.

Content-Based Distractors from External Knowledge The `vqa_trajectory_understanding` function generates distractor language instructions by using templates (e.g., "Pick up the {} from the {}") and filling them with common objects and locations, which may or may not be present in the current scene, thus testing a deeper understanding of the visualized trajectory against plausible alternative tasks.

Strategic Shuffling and "None of the above" The `_shuffle_choices` method is systematically called after initial VQA construction. This method randomizes the order of the correct answer and the initially formulated incorrect choices. Furthermore, for non-binary questions (typically those with five choices), there is a 20% chance to replace the actual correct answer with "None of the above", and the original correct answer text is then discarded for that instance, making "None of the above" the correct choice. This adds another layer of complexity, requiring the system to not only identify the correct option but also to recognize when none of the substantive options are correct.

The combination of these strategies ensures a diverse set of distractors, tailored to the specific type of question being posed and the visual information presented.