Supplementary Material

1. Additional Dataset Details

1.1. Dataset Construction and Labeling

1.1.1. Static Scene Construction

Episode History Collection. To build the static-scene subset of EQA-Decision, we collect episode histories H from two large-scale 3D environments: ScanNet and HM3D. For ScanNet, we use the first 600 frames (approximately 30 seconds) of the original human-captured RGB-D video trajectories, which naturally provide rich coverage of indoor scenes and object configurations. These segments serve as episode histories for QA construction.

Unlike ScanNet, the HM3D dataset does not include pre-recorded exploration trajectories. We therefore generate episode histories using a semi-automated procedure. For each environment, we select a start location x_{src} and a destination x_{dst} such that the geodesic distance exceeds 10 meters and the path contains significant curvature. This encourages the trajectories to span multiple rooms and diverse viewpoints. The agent follows the shortest geodesic path and performs panoramic scans every one meter by rotating up to 180°, mimicking human-like exploration behavior. All trajectories are manually inspected to ensure scene coverage and to filter out cases with unproductive segments, such as long intervals facing walls.

We additionally extract episode histories from the ALFRED environment to enrich task-oriented static-scene observations. ALFRED provides interactive household scenes together with a symbolic planner and a set of atomic actions. For each successful demonstration, we replay the complete execution using the CAPEAM agent and record RGB frames at a regular interval of ten atomic actions. This sampling strategy preserves key stages of the task while avoiding redundancy. During replay, we also query the simulator for ground-truth metadata. These include the set of objects currently visible to the agent, their semantic states, and other environment attributes that remain stable across short temporal spans. The resulting trajectories capture both the visual layout of the environment and the underlying scene semantics, producing reliable episode histories for static-scene QA construction.

Prompt for Scene Summary

You are assisting in constructing an embodied question–answering dataset. Your task is to summarize only the verifiable visual facts from the provided frames.
Strictly follow these rules:
- Only list objects that are clearly visible. Do not guess or hallucinate.
- Include directly observable attributes such as color, shape, material, and simple open/closed states.
- Provide coarse spatial relations only when visually obvious (e.g., “on”, “next to”, “left of”).
- Keep all descriptions concise and factual.

Output format:
- visible_objects:
- attributes:
- spatial_relations:

Figure 1. Prompt used for static-scene summarization.

Question Generation. To construct the static-scene subset of EQA-Decision, we adopt a two-stage pipeline that combines trajectory summarization with prompt-based QA generation. For each episode history H, we first select representative RGB frames and use Gemini to produce a structured scene summary as shown in Figure 1. The model is instructed to list only verifiable elements in the scene, including visible objects, observable attributes, and coarse spatial relations, while hallucinated or uncertain information is strictly prohibited. These summaries serve as the factual grounding for subsequent QA construction.

Based on the generated scene summaries, we then apply a controlled few-shot prompting scheme to produce question–answer pairs, as illustrated in Figure 2. The prompt specifies the annotation intent and emphasizes the need to generate natural and visually grounded questions that an everyday user might ask, along with concise answers supported solely by the scene summary. Generated outputs are automatically screened to remove unverifiable or ambiguous content before undergoing manual review to ensure accuracy and category balance.

Prompt for QA Generation

You are assisting in generating static-scene question–answer pairs. Your task is to create natural and visually grounded QAs based solely on the provided scene summary.
All questions should reflect what a household user might naturally ask, and all answers must be directly supported by the summary. Avoid hallucinated content or external knowledge.
Scene summary:
SCENE SUMMARY
Below are examples of valid question types. They demonstrate the expected style but do not restrict the diversity of generated QAs.
State
question: Did I leave the cabinet door open?
answer: Yes, the cabinet door is open.
Location
question: Where is the mug in the kitchen?
answer: It is on the right side of the counter.
Attribute
question: Is the chair in the living room made of wood?
answer: Yes, it is wooden.
Counting
question: How many pillows are on the sofa?
answer: Three.
Existence
question: Is there a mirror in this room?
answer: No.
Based on the scene summary above, generate a diverse set of grounded and unambiguous QA pairs across these categories.

Figure 2. Prompt used for static-scene QA generation.

1.1.2. Spatial Understanding

Depth Map Construction and Object-Level Geometry. To equip our dataset with spatial reasoning signals, we augment all RGB-only images with metric depth predictions. Following practices from Bunny-695k, each image is processed using a high-quality monocular depth estimator ZoeDepth to generate a dense depth map D. To preserve both millimeter-level indoor precision and large outdoor depth ranges, the predicted metric depth is encoded using a reversible multi-channel format, either as a single-channel uint24 map or a three-channel uint8 representation with different quantization multipliers. This encoding retains fine-grained geometric cues while maintaining compatibility with standard image encoders.

To obtain object-level geometric information, we pair bounding-box annotations with segmentation masks. For each annotated region, we apply SAM with both the bounding box and its center point as prompts to produce a set of candidate masks. We filter out masks that significantly exceed the bounding-box extent and select the one with the highest confidence. When segmentation is unreliable, we fallback to sampling the depth at the region's center, ensuring stable depth extraction across all cases. Using the resulting mask M, we compute a set of robust depth descriptors—including minimum depth, maximum depth, mean depth, and center depth—to characterize each object. To suppress noise, the extremal depths are computed using the 5th and 95th percentiles of depth values within M. These descriptors capture reliable 3D relationships such as relative distance, occlusion ordering, and coarse surface layout, forming the geometric foundation for downstream spatial question answering.

Prompt for Depthmap Understanding

Design a conversation between you and a human talking about the depth map. The human asks you to describe the depth map. You should focus on depth value predictions. The colors only represent depth values. Do not directly mention colors on the image in your response; instead, mention the depth distribution they correspond to.
By looking at the depth map, you should also infer what may exist in the RGB image. If something truly exists in the RGB image and can be inferred from the depth map, you may mention it. If possible, pay attention to spatial relationships. When referring to spatial relationships such as left and right, always use real-world orientation rather than the image coordinate system.

Figure 3. Prompt used for depthmap understanding.

Question Generation. Based on the object-level depth statistics and scene geometry, we generate spatial QA pairs using a controlled prompting scheme. As shown in Figure 3 and Figure 5, the prompt instructs the model to rely solely on verifiable depth cues and observed spatial relations, producing grounded questions that an everyday user might ask (e.g., near–far comparison, left/right ordering, occlusion reasoning). All answers are constrained to be concise, factual, and fully supported by the extracted geometric information, ensuring that the generated QA pairs reflect reliable spatial understanding.

1.1.3. Task Dynamics Reasoning

Construction. The task dynamics module captures temporal and causal relationships in embodied tasks and is composed of three sub-components: sub-task planning, state tracking with causal judgment, and progress estimation.

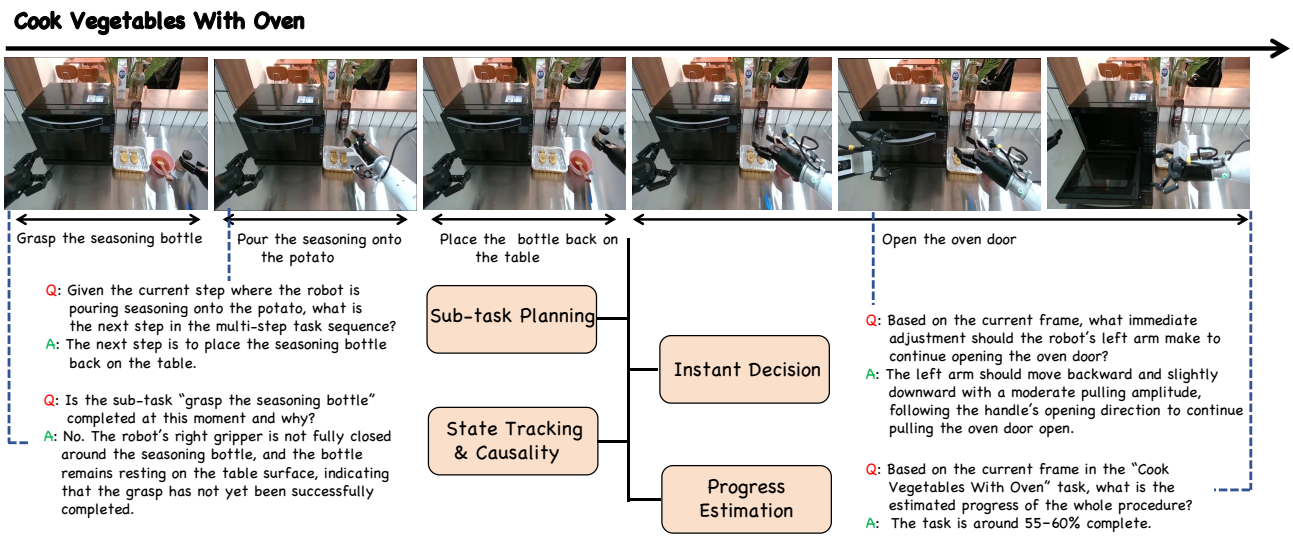


Figure 4. Annotated example for the "Cook Vegetables With Oven" task.

Prompt for Spatial Understanding

Design a conversation of no more than three Question–Answer pairs between you and a person asking about the image. The conversation should be logically connected. Think about the spatial relationships of objects in the image, and generate the conversation based on these relationships.
Spatial relationships may include, but are not limited to: positional (left/right, below/above, behind/front), distance (further/closer to the camera or to another object), size (big/small, tall/short, wide/thin), or reach (whether object A touches or reaches object B).
When describing spatial relationships, always use real-world orientation, as if you were standing in the scene. For example, "right side of the object" should refer to the object's right side in the real world, not the right side of the image. Only describe relationships you are certain about.

Figure 5. Prompt used for spatial understanding.

We start from temporally annotated egocentric trajectories, where each frame is associated with an action label and a task identifier.

For sub-task planning, we build a semi-automatic pipeline that converts raw action annotations into structured multi-step tasks. For each video, temporally contiguous actions that belong to the same high-level task are grouped into a coherent multi-step segment. Each atomic action inside the segment is treated as a step-level observation. We record the key frame at the action timestamp together with several context frames before and after the step. These step boundaries and their surrounding frames serve as the basic units for subsequent temporal reasoning. To enrich linguistic diversity while preserving the underlying temporal structure, we later use Gemini-2.5-pro to rewrite and augment the textual descriptions of these steps.

For state tracking and causality, we focus on determining which sub-task the agent is currently executing, whether that sub-task has been successfully completed, and why. We reuse the step boundaries obtained above and uniformly sample frames immediately before and after each annotated boundary. Pairs of frames in which the queried sub-task has already been completed are treated as positive examples, whereas frames captured before completion form negative examples. Gemini-2.5-pro is then employed to verbalize the causal justification and to highlight image-grounded evidence that support each judgement.

For progress estimation, we follow a motion-phase-based segmentation procedure. Long-horizon trajectories from sources such as AgiBot and Open X-Embodiment are decomposed into motion phases according to velocity and directional variations. Given the phase boundaries, we compute normalized progress ratios by mapping the current frame index onto $[0, 1]$ along the task timeline. These progress labels provide dense temporal supervision for learning progress-related reasoning skills.

Question Generation. Based on the structured temporal annotations above, we adopt a controlled prompting scheme to generate question–answer pairs for all three subcomponents. For sub-task planning, questions focus on future-step prediction, remaining-step reasoning, and con-

Prompt for State Tracking and Causality

You are generating question–answer pairs for determining the current sub-task state and its causal justification. Each sample includes frames before and after a sub-task boundary.
Follow these rules:
- Decide whether the sub-task is ongoing, completed, or not yet started.
- Provide a short justification grounded in visual evidence.
- Do not introduce any unobservable outcomes or external knowledge.
- Ensure that the reasoning reflects causal consistency across frames.

Output format:

```
Q: <question>
A: <answer>
```

Figure 6. Prompt used for state tracking and causal reasoning QA generation.

textual multi-step planning, conditioned on the observed step sequence and its visual context. For state tracking and causality, prompts shown in Figure 6 guide the model to decide the current sub-task state and to articulate image-grounded evidence explaining the decision. For progress estimation, questions target the estimation of normalized progress through a task and the identification of motion-phase transitions. We additionally visualize in Figure 4 an annotated example from the "Cook Vegetables With Oven" task, illustrating how the task-dynamics module is constructed.

1.1.4. Instant Decision

Construction. The instant-decision module focuses on modeling the agent's real-time decision-making process in dynamic embodied environments. We begin with densely annotated trajectories collected from diverse embodied datasets. For each trajectory, we extract continuous action segments such that the temporal gap between adjacent annotations does not exceed a fixed threshold (typically three to five seconds), ensuring fine-grained temporal continuity and preserving transient interaction cues.

Between every two consecutive action annotations, we uniformly sample intermediate frames that capture short-term transitional decision states. For each sampled frame, Gemini-2.5-pro is provided with the neighboring annotations together with a compact description of the ongoing task. This enables the model to summarize key contextual details, including the agent's current intention, the partially completed step, and the short-horizon spatial relations between the agent and relevant objects. These summaries are then paired with the corresponding visual inputs to construct question–answer samples that reflect context-dependent decisions and next-action prediction under uncertainty.

Prompt for Instant Decision

You are generating question–answer pairs for the *instant decision* task. Each sample includes two frames: the earliest frame of the current high-level step and the current frame (NOW). All reasoning must be based **only** on the visible evidence from the current frame.
Task:

```
task
```

now sub-task:

```
sub-task
```

Follow these rules:
- Identify the key visible facts: target object, robot arm, spatial relations, and contact state (contact / close / far).
- Decide whether the current high-level step is completed.
- If the step is completed, provide a short transition-style reasoning and justification grounded in visible evidence.
- If the step is not completed, generate a concise question asking what action the robot should take next.
- Answers must be imperative, physically executable low-level actions (*e.g.*, "move slightly downward toward the bottle") without hallucinated details.

Output format:

```
Q: <question>
A: <answer>
```

Figure 7. Prompt used for instant-decision QA generation.

To ensure annotation quality, we conduct both automated consistency checks and manual verification on sampled subsets. Ambiguous or contradictory cases are removed through this mixed filtering process. The resulting corpus provides a dense set of instant-decision examples that capture real-time perception–action coupling and short-horizon adaptivity in embodied tasks.

Question Generation. Based on the intermediate decision summaries, we design prompts shown in Figure 7 that elicit natural questions about imminent actions, likely next steps, and short-term intentions. Gemini-2.5-pro generates question–answer pairs grounded solely on the sampled visual frames and their adjacent annotations. Questions tar-

Statistic	Value
Question Length	
Average length	92.58
Median length	86
Minimum length	12
Maximum length	1302
Answer Length	
Average length	66.53
Median length	35
Minimum length	2
Maximum length	3730
Image Resolution	
Average resolution	557×444
Min resolution	45×49
Max resolution	2667×2428
Most common:	640×512

Table 1. Summary statistics of questions, answers, and image resolutions in the EQA-Decision dataset.

get the agent's immediate decision rationale, while answers remain concise, factual, and supported by observable evidence. This pipeline produces a large set of fine-grained, visually grounded instant-decision questions suitable for evaluating real-time reasoning and responsiveness.

1.2. Additional Dataset Statistics

Question and Answer Length Distribution. As shown in Table 1, we provide detailed statistics for all textual components in the EQA-Decision dataset. In total, the dataset contains 4.69 million questions and answers. The average question length is 92.58 characters, and the median length is 86 characters. The questions span a wide range of styles, from brief command-like prompts with a minimum length of 12 characters to long, temporally conditioned instructions that reach up to 1302 characters. The most concise questions resemble short user commands, while the longest ones come from task trajectories that include summarized histories of the previous twenty steps.

Answers are generally more compact. The mean answer length is 66.53 characters, and the median length is 35 characters. The shortest answers contain only two characters, for example the response "no", which often appears in binary decision tasks or yes–no annotations inherited from earlier datasets. At the other extreme, the longest answers reach 3730 characters, typically produced in trajectory-level reasoning scenarios or inherited from verbose upstream outputs. These statistics reveal a substantial range of linguistic complexity across different modules. The dataset covers everything from concise scene descriptions and simple object

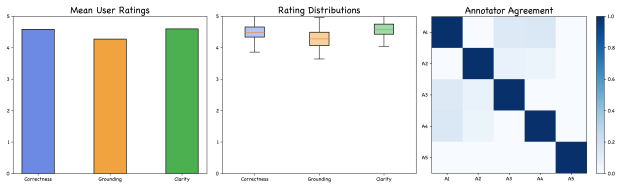


Figure 8. User study results on annotation quality. The figure shows (left to right): mean ratings across correctness, grounding, and clarity; rating distributions for each dimension; and inter-annotator agreement across all five evaluators.

attributes to long-horizon temporal reasoning questions that require understanding of multi-step histories. This diversity reflects the breadth of the dataset and introduces meaningful challenges for language modeling and temporal grounding.

Image Resolution Distribution. We analyze the resolution statistics of all images in the dataset. The average image size is approximately 557 pixels in width and 444 pixels in height, but the range is highly diverse. Resolutions span from very low-resolution frames of 45×49 pixels to ultra-high-resolution images reaching 2667×2428 pixels. The most frequent resolutions include 640×512, 500×375, 500×333, and 640×480, reflecting contributions from simulated environments, web imagery, egocentric videos, and robot-captured observations.

The substantial resolution variability indicates that models trained on the dataset must be robust to non-uniform image scales and aspect ratios, benefiting architectures that incorporate strong visual normalization or multi-scale processing.

1.3. Human Review Process.

To ensure the reliability of the automatically generated annotations, we conducted a dedicated human-review stage after the initial generation. First, we randomly sampled approximately 1% of all generated QA pairs across all modules for manual inspection. Each sampled item was examined for clarity of the question, correctness of the answer, consistency with the visual data, and the absence of hallucinated or unobservable content. A second reviewer then performed a cross-check on a smaller subset—about 8% of the sampled items—to compute inter-annotator agreement, which reached a Cohen's Kappa of approximately 0.70. We classified flagged issues into three major categories: answer incorrect or unsupported by evidence; question unclear or ambiguous; visual or temporal mismatch. Based on the feedback, we removed approximately 2% of the sampled items and corrected around 6% as needed. The error analysis outcomes were used to refine the generation prompts and annotation guidelines as part of a continuous feedback loop. This human-in-the-loop step ensures that our dataset meets high quality standards for downstream model training and

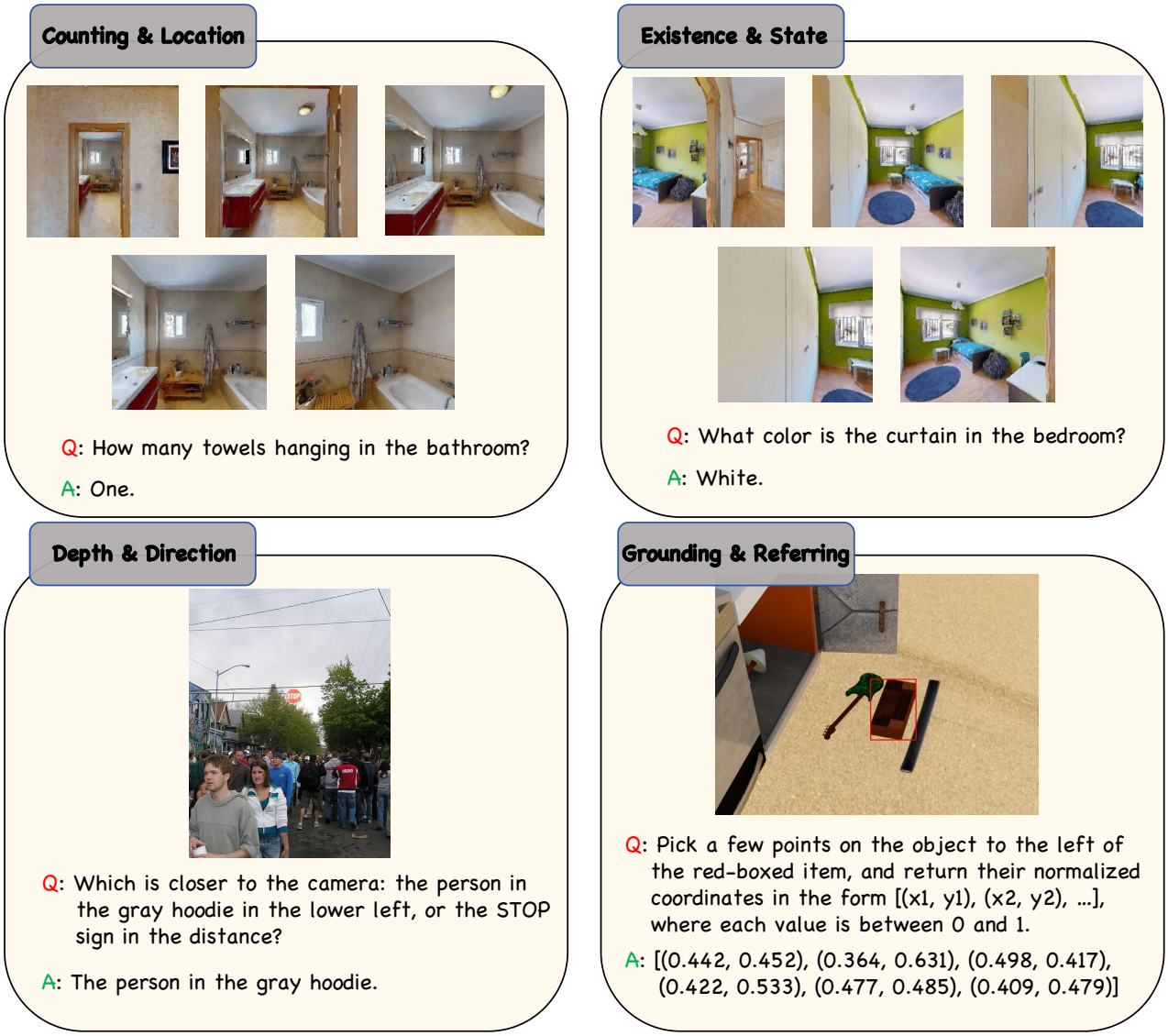


Figure 9. EQA-Decision visualization results

evaluation.

User Study on Annotation Quality. Beyond internal consistency checks, we additionally conducted a small user study shown in Figure 8 to assess the perceived quality of our automatically generated QA pairs. We randomly sampled 400 items from the final dataset, stratified across all reasoning modules. Each item was presented with its associated visual or temporal context together with the corresponding question–answer pair. Five undergraduate participants with basic background in computer vision or NLP were asked to rate each item along three dimensions on a 1–5 Likert scale: (i) Correctness of the answer, (ii) Grounding of the QA in the provided visual/temporal evidence, and (iii) Clarity of the question formulation.

Across all samples, the mean scores were 4.5 for Correctness, 4.3 for Grounding, and 4.6 for Clarity, with more than 90% of the items receiving a rating of at least 4 on all three dimensions. The ratings from different participants were highly consistent, with an average pairwise correlation of 0.72. These results suggest that users perceive the majority of our QA pairs as accurate, well grounded, and clearly phrased, providing further evidence for the overall quality of the released dataset.

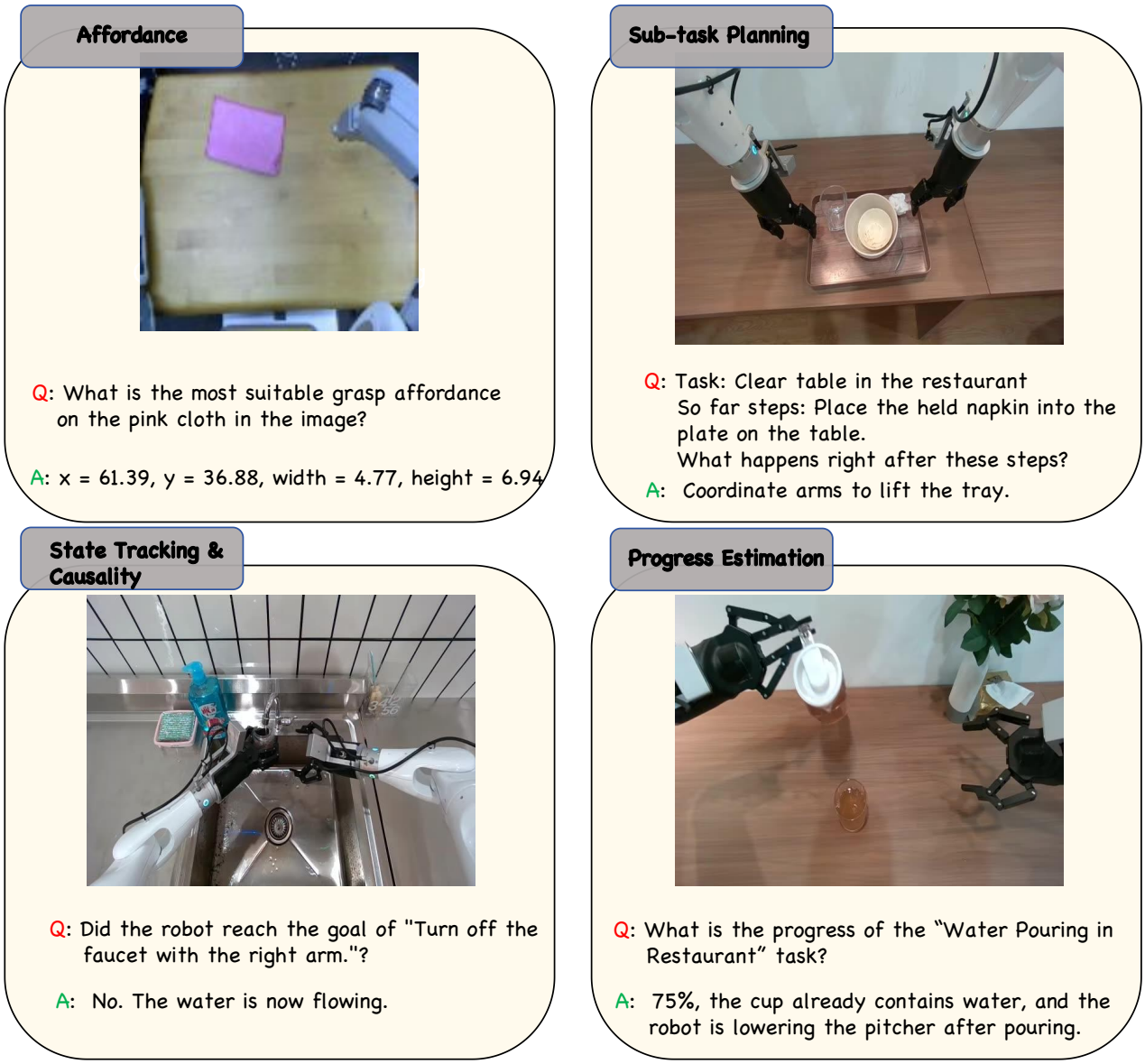


Figure 10. EQA-Decision visualization results

2. Training Details

Both Stage 1 (Supervised Fine-Tuning) and Stage 2 (Chain-of-Thought Supervised Fine-Tuning) adopt a unified optimization setup that enables stable domain adaptation of Qwen3-VL-8B-Instruct to embodied-reasoning tasks. Training is executed with DeepSpeed ZeRO-3 and a global batch size of 128 distributed across eight RTX PRO 6000(96GB) GPUs. Each device processes sixteen samples per step, combined with gradient accumulation to match the global batch. We use AdamW with learning rate 2×10^{-4}, cosine decay scheduling, a warm-up ratio of 3%, momentum parameters ($\beta_1 = 0.9,\ \beta_2 = 0.95$), and weight decay of 0.1. This configuration provides a stable optimization profile for long-sequence multimodal training.

To keep training parameter-efficient, we apply LoRA only to the language-fusion components while keeping the vision encoder, the cross-modal merger, and all normalization layers frozen. The LoRA configuration uses rank 64, $\alpha = 64$, and a dropout rate of 0.05, allowing the model to acquire new reasoning and instruction-following behaviors without overfitting perception modules. In Stage 2, the model continues from the Stage 1 checkpoint with the same LoRA structure, but the training data now includes chain-of-thought explanations generated by Gemini-2.5, providing richer supervisory signals and helping stabilize the

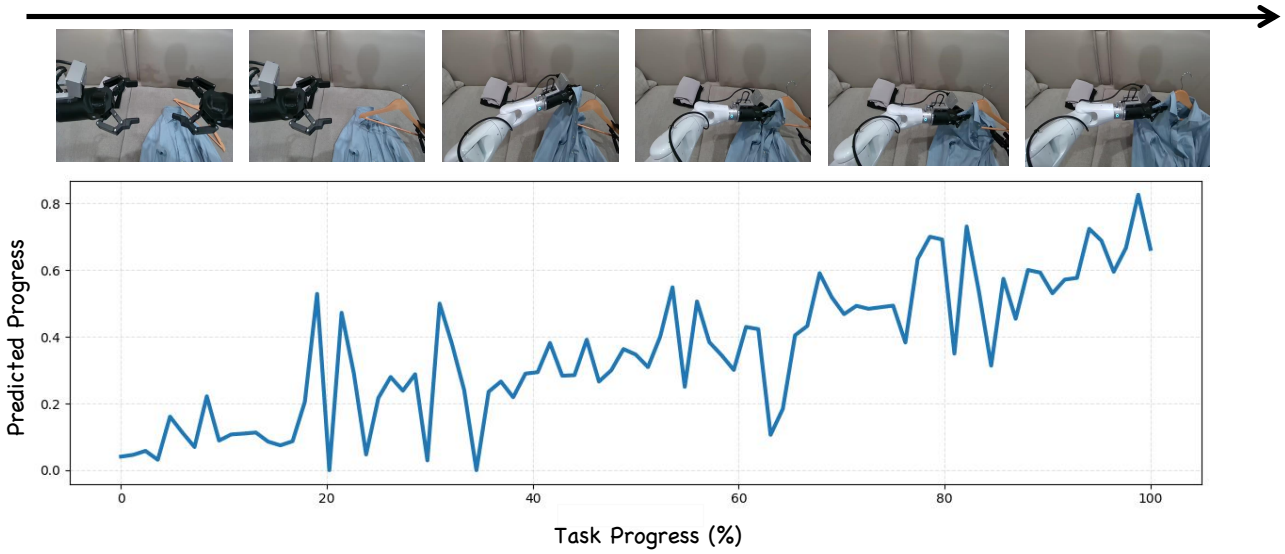


Figure 11. Visualization results of RoboDecision-8B on task progress estimation.

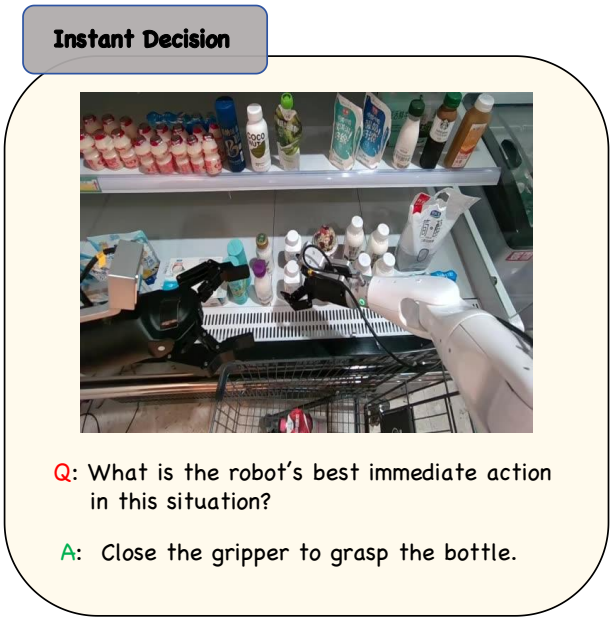


Figure 12. EQA-Decision visualization results

reinforcement-learning stage that follows.

After completing the two supervised fine-tuning stages, we further refine the model using a reinforcement-learning phase based on Group Relative Policy Optimization. This stage starts from the Stage 2 checkpoint while keeping the same parameter-efficient adapters; all perception-related components, including the vision encoder, the cross-modal merger, and normalization layers, remain frozen so that optimization focuses entirely on reasoning, grounding, and action selection. Training is performed with DeepSpeed ZeRO-3 for one epoch over the reinforcement dataset. Because reinforcement learning requires sampling multiple response candidates, each device processes a batch of one sample and accumulates gradients over eight steps to match the effective global batch size used previously. For every input prompt, the model generates eight candidate trajectories, with a maximum prompt length of 2048 tokens and a maximum completion length of 2048 tokens. We use a learning rate of 5×10^{-6} with cosine decay, a warm-up ratio of three percent, and AdamW with weight decay of 0.1. FlashAttention is enabled to maintain efficient attention computation over long multimodal sequences, and training is conducted entirely in `bfloat16` precision. Checkpoints are saved at regular intervals throughout training.

Reward computation follows a unified reward function that jointly evaluates the correctness of the final answer and the coherence of the produced reasoning trace. In practice, we combine the three reward components using fixed weights,

$$\alpha = 0.5, \quad \beta = 1.0, \quad \gamma = 0.2,$$

where α controls the contribution of reasoning quality, β emphasizes factual answer accuracy, and γ provides a lightweight routing signal that stabilizes transitions between "thinking" and "answering" modes. This formulation encourages accurate predictions, structured multi-step reasoning, and visually grounded decision-making. Through this reinforcement stage, the model strengthens its visual grounding, becomes more robust in long-horizon reasoning, and exhibits improved decision-making reliability across di-

verse embodied scenarios.

3. Additional Visualizations

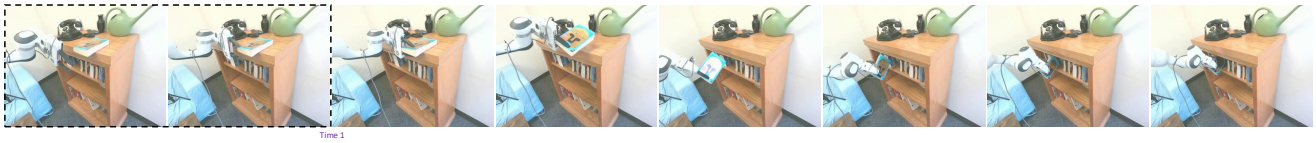
In this section, we provide additional visualization results to further illustrate both the structure of the EQA-Decision dataset and the embodied reasoning capabilities of our RoboDecision-8B model. As introduced in the main paper, the dataset is organized into four major reasoning modules—*Static Scene Construction*, *Spatial Understanding*, *Task Dynamics Reasoning*, and *Instant Decision*—which together decompose into nine fine-grained sub-tasks.

To offer a clearer view of the reasoning granularity covered by the dataset, Figures 9, 10, and 12 visualize one representative example for each of the nine sub-tasks. These examples are taken directly from the EQA-Decision dataset and reflect the diversity of scene understanding, spatial inference, temporal reasoning, and multimodal decision cues that arise across tasks.

Beyond dataset visualizations, we also include two qualitative examples demonstrating the embodied decision-making behavior of our RoboDecision-8B model. Figure 11 shows the model's progress estimation capabilities, predicting task completion percentages along long-horizon manipulation sequences. Figure 13 highlights the model's planning and instant decision-making behavior, illustrating how it selects the next best action from the current visual observation during task execution.

Together, these visualizations provide a comprehensive view of both the dataset's reasoning coverage and the model's perception-to-decision performance across diverse embodied scenarios.

Robot Planning

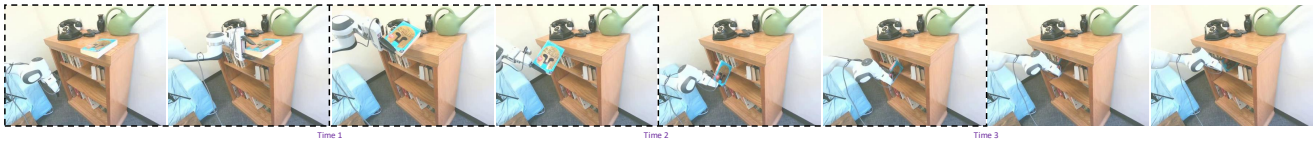


Question at time 1: The robot arm is organizing the shelf. List the next few steps based on the video segment.
Qwen3-8B:
1. Pick up the book 2. Move to a different location 3. Place the book down 4. Repeat for other items 5. Adjust or reposition
RoboDecision:
<reason>The robot arm is holding a book and moving it towards the bookshelf. The next steps will be to place the book on the shelf and then move to pick up the other book from the top of the shelf.</reason>
<answer> 1. pick the book from the top of the table 2. place the book in the shelf 3. move the book backwards 4. release the book 5. move your arm up</answer>

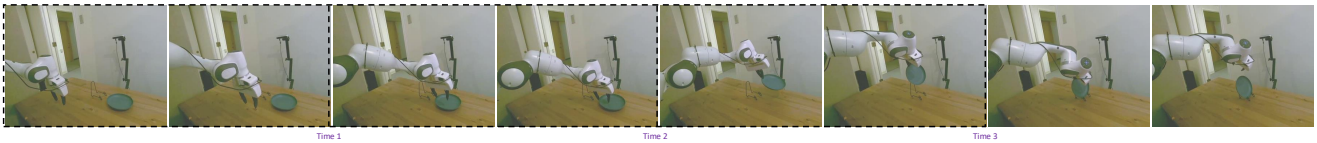
Question at time 1: The robot arm is flipping the mug upright. List the next few steps based on the video segment.
Qwen3-8B:
The robot arm is flipping the mug upright. The next steps are: 1. The robot arm will release the mug on the table 2. The robot arm will move away from the mug.
RoboDecision:
<reason>The robot arm is positioned above the blue mug. To flip it upright, the arm needs to move down to grasp the mug, lift it, rotate it, and then place it back on the table.</reason>
<answer>1. Pick up the blue cup 2. Rotate the blue cup 3. Place the blue cup on the table 4. done</answer>

Question at time 1: The robot arm is loading the plate. List the next few steps based on the video segment.
Qwen3-8B:
The robot arm is moving towards the plate.
RoboDecision:
<reason>The robot arm moves down to grasp the plate, then lifts it up from the table.</reason>
<answer>1. pick up the plate 2. place the plate in the stand 3. move away from the stand</answer>

Robot Decision



Question: The robot arm is organizing the shelf. List the next step based on the video segment.
At Time 1:
Qwen3-8B:
The robot arm is organizing the shelf by moving the book.
RoboDecision:
<reason>The robot arm is holding a book and moving it towards the top shelf of the bookcase.</reason>
<answer>Pick the book</answer>
At Time 2:
Qwen3-8B:
Place the book on the table,
RoboDecision:
<reason>The robot arm has grasped the book from the top of the shelf. The next logical step to organize the shelf is to place the book on one of the empty shelves below.</reason>
<answer>Place the held book in the empty shelf space</answer>
At Time 3:
Qwen3-8B:
Place the book in the shelf,
RoboDecision:
<reason>The robot arm has picked up the book from the top of the shelf and is now holding it in front of the shelf. The next logical step is to place the book onto one of the empty spaces on the shelf.</reason>
<answer>put the book in the shelf</answer>



Q: The robot arm is loading the plate. List the next step based on the video segment.
At Time 1:
Qwen3-8B:
Place the plate on the table,
RoboDecision:
<reason>The robot arm's gripper is positioned directly above the plate, indicating the next step is to close the gripper to grasp it.</reason>
<answer>move down to the plate</answer>
At Time 2:
Qwen3-8B:
The robot arm is holding the plate.
RoboDecision:
<reason>The robot arm has lowered the plate onto the table. The next logical step is to close the gripper to grasp the plate.</reason>
<answer>pick up the plate</answer>
At Time 3:
Qwen3-8B:
Place the plate on the stand,
RoboDecision:
<reason>The robot arm has successfully grasped and lifted the plate from the table. The next logical step is to move the plate to a new location.</reason>
<answer>place the plate on the dish rack</answer>

Figure 13. Visualization results of RoboDecision-8B on Instant Decision.