

# **RoboAnnotatorX: A Comprehensive and Universal Annotation Framework for Accurate Understanding of Long-horizon Robot Demonstration**

Supplementary Material

## A. Details of Model Architecture

Our model architecture implements a hierarchical approach to video understanding through a multi-scale temporal processing framework, as detailed in Tab. 1. At its foundation lies a powerful ViT-G/14 visual encoder that extracts frame-level visual features with a hidden dimension of 1408 and patch size of 14. For clip-level and global temporal understanding, we employ a Q-Former-based time-aware encoder that transforms raw visual features into structured representations with a hidden dimension of 768. As shown in Tab. 1, the encoder utilizes 32 queries and 8 attention heads. The encoder takes both visual features and temporal position embeddings  $p_t$  as input. Following TimeChat [22], we implement  $p_t$  as the embedding of text descriptions like "This is frame t" using the language model's tokenizer, rather than using numerical timesteps directly, which provides better alignment between visual and temporal information. The clip-level stream further incorporates a temporal Q-Former with identical specifications for fine-grained motion dynamics, while the global stream utilizes a 4-layer transformer encoder with 8 attention heads to capture long-range dependencies across the entire sequence. Each stream employs a specialized projector to transform the visual features (see Tab. 1): the scene-level projector is an MLP2X\_GELU that transforms 1408-dimensional features into 4096-dimensional representations through grid pooling, the clip-level projector is a linear layer that projects 768-dimensional temporal features to 4096 dimensions, and the video-level projector is a linear layer that projects 768-dimensional features to 4096 dimensions representation. Finally, the multi-scale features are concatenated and fed into a Vicuna-7B-v1.5 with a hidden dimension of 4096, which generates comprehensive video descriptions and responses by integrating information across all temporal scales. As detailed in Tab. 1, the language model does not utilize image start/end tokens or image patch tokens. This carefully orchestrated architecture enables a robust understanding of complex video sequences by effectively capturing both fine-grained temporal dynamics and global semantic context.

Table 1. **RoboAnnotatorX architecture details.** The architecture processes visual information through three complementary streams: (1) A scene-level stream using ViT for visual feature extraction, (2) A clip-level stream with Time-aware Encoder and Temporal Q-Former for modeling local dynamics, and (3) A video-level stream with Time-aware Encoder and Global Transformer for capturing long-range dependencies. Each stream employs a dedicated projector to align feature dimensions before integration with the language model.

Component	Parameter	Value	Description
Visual Encoder	Model Type	ViT-G/14 [7]	Base vision transformer for frame-level encoding
	Hidden Dimension	1408	-
	Patch Size	14	-
Time-aware Encoder	Model Type	Q-Former [5]	Generates query-based visual representations
	Hidden Dimension	768	-
	Number of Queries	32	-
	Number of Heads	8	-
Temporal Q-Former	Model Type	Q-Former [5]	Processes temporal information within clips
	Hidden Dimension	768	-
	Number of Queries	32	-
	Number of Heads	8	-
Global Transformer	Model Type	Transformer Encoder [5]	Capture global temporal dependencies
	Hidden Dimension	768	-
	Number of Layers	4	-
	Number of Heads	8	-
Scene-level Projector	Model Type	MLP2X_GELU	Projects scene-frame features after grid pooling
	Input Dimension	1408	-
	Hidden Dimension	4096	-
	Output Dimension	4096	-
Clip-level Projector	Model Type	Linear	Projects clip-level temporal features
	Input Dimension	768	-
	Output Dimension	4096	-
Video-level Projector	Model Type	Linear	Projects global dynamic features
	Input Dimension	768	-
	Output Dimension	4096	-
Language Model	Model Type	Vicuna-7B-v1.5 [32]	Processes concatenated features
	Hidden Dimension	4096	-
	Use Image Start/End	False	-
	Use Image Patch Token	False	-

## B. Human Validation & LLM Evaluation Bias Analysis

To ensure the reliability and robustness of our evaluation protocol, we conduct a comprehensive analysis comparing LLM-based automatic scoring with human judgments. This analysis reveals two key insights regarding the consistency and limitations of LLM-based evaluation.

**First**, we observe strong scoring consistency across different large language models (LLMs), as illustrated in Fig. 1. For instance, the average *Pearson correlation coefficient* between GPT-4o and other LLMs reaches **0.91**, indicating high alignment in scoring trends. More importantly, human validation confirms that the **relative ranking of models remains stable**, regardless of which LLM is used for evaluation. These findings support the conclusion that LLM-based metrics can serve as a reliable proxy for model comparison, and our comparative analysis remains robust across evaluator choices.

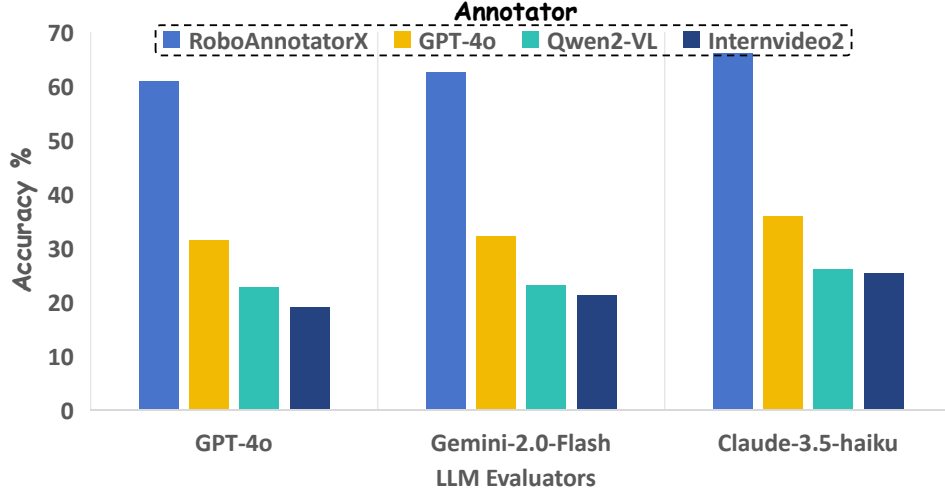


Figure 1. **Evaluation score consistency across LLM evaluators.** Bar chart showing the average scores assigned to each model by different LLM evaluators. The consistent ranking patterns across evaluators indicate strong agreement.

**Second**, we identify a small but meaningful gap between LLM-based and human evaluations. As shown in Tab. 2, our model consistently receives higher scores from human evaluators. This discrepancy stems from a limitation in the common VQA LLM-based evaluation, where the model’s predicted answer is compared against a small set of reference text answers, rather than evaluated based on visual information. Consequently, visually correct answers that deviate in wording or detail from the reference texts may be wrongly marked as incorrect. In contrast, human evaluation acts as the *gold standard* by assessing answers directly against visual evidence, rather than relying on text matching. This allows for fair recognition of alternative but equally valid expressions. To properly quantify this effect, we propose the **Alternative Answer Impact (AAI)**—the relative change between LLM-based and human evaluation. This evaluation analysis reveals that our model achieves higher scores under human evaluation. The potential reason is that our model avoids overfitting to specific formats with different description styles or varying detail granularity—common to humans as well, enabled by our diverse dataset and curriculum training. Moreover, it stands out in interpreting temporal-spatial relationships from multiple perspectives (e.g., describing actions at different stages or positions using different reference points), thanks to multi-scale temporal encoder tailored for complex long-horizon demonstrations. More failure case details and demos please see Fig. 19.

Table 2. **Evaluation comparison across different evaluators.** Human evaluations yield consistently higher scores than LLM-based ones, especially for RoboAnnotatorX.

Model	LLM Avg.		Human		AAI	
	Acc (%) ↑	Score (0-5) ↑	Acc (%) ↑	Score (0-5) ↑	Acc (%) ↑	Score (%) ↑
GPT-4o	33.09	2.14	36.13	2.37	+1.33	+0.13
Qwen2-VL	23.95	1.84	24.51	1.87	+0.56	+0.03
Internvideo2	21.86	1.65	22.46	1.70	+0.60	+0.05
RoboAnnotatorX	63.14	3.59	68.79	3.85	+5.65	+0.26

## C. Details of Baselines & More Experimental Results

### C.1. General-domain MLLM

General-domain MLLMs are pre-trained on broad visual-language datasets, equipping them with robust general understanding capabilities across diverse domains. While these models demonstrate strong performance in tasks like visual recognition, scene understanding, and natural language interaction, they may lack specialized comprehension of robotics-specific concepts, terminology, and spatial-temporal relationships. This limitation can affect their ability to interpret robotic actions, understand mechanical constraints, or reason about physical interactions in robotic contexts.

- **Qwen2-VL** [24] advances the Qwen-VL [1] architecture with two key innovations for enhanced video processing: Naive Dynamic Resolution and Multimodal Rotary Position Embedding (M-ROPE). The former enables the processing of arbitrary image resolutions through dynamic visual token mapping, while M-ROPE’s temporal-spatial decomposition integrates 1D textual, 2D visual, and 3D video positional information. These improvements enable analysis of videos exceeding 20 minutes and enhance performance across video-based tasks, visual reasoning, and multilingual text recognition.
- **PLLaVA** [27] is an efficient video-language model that adapts image-language pre-trained models for video tasks through a simple yet effective pooling strategy. The model processes videos by sampling 16 frames and applies a pooling operation along the temporal dimension to smooth feature distribution, focusing on reducing the spatial dimension while maintaining a larger temporal dimension. This approach addresses the challenges of direct fine-tuning on video datasets, including performance saturation and prompt sensitivity.
- **LLaVA-NEXT-Video** [31] is an advanced Large Multimodal Model designed for video understanding tasks, building upon the image-trained LLaVA-NEXT [14] model. It processes videos using the AnyRes technique, which represents each video frame as a 12x12 token grid. The model can handle up to 56 frames per video, thanks to a linear scaling technique that expands the maximum token length from 4096 to 8192. This approach allows the model to effectively process and understand videos of varying lengths and resolutions.
- **InternVideo2** [26] introduces a comprehensive video foundation model family, powered by a 6-billion parameter video encoder and unified training that combines masked video modeling, cross-modal contrastive learning, and next token prediction. Its video processing pipeline converts input videos into spatiotemporal tokens by sampling 8 frames with 14x14 spatial downsampling, which are then processed through a ViT architecture with attention pooling and 3D position embeddings. The integration of audio and text encoders enables robust cross-modal capabilities across video recognition, video-text tasks, and video dialogue.
- **MiniCPM-V** [28] presents an efficient approach to multimodal large language models (MLLMs) designed specifically for end-device deployment. The model employs a three-module architecture: a visual encoder for image processing, a compression layer with a single-layer cross-attention perceiver resampler for token compression, and an LLM backbone for text generation. This architecture enables MiniCPM-V to handle high-resolution images up to 1.8M pixels at any aspect ratio while maintaining strong performance across various tasks.
- **VideoGPT+** [15] features a dual-encoder design combining CLIP-L/14 for spatial details and InternVideo-v2 for temporal context, unified through Vision-Language projection layers and processed by a Phi-3-Mini language model. Using segment-wise sampling and adaptive pooling, it demonstrates strong performance across benchmarks while contributing a 112K video-instruction dataset and the VCGBench-Diverse benchmark spanning 18 video categories.
- **VideoLLaMA2** [4] advances video understanding with a dual-branch architecture. Its video pipeline processes 16 frames at 336x336 resolution using a CLIP backbone [20] while introducing a Spatial-Temporal Convolution connector that improves upon its predecessor’s Q-former [29]. An integrated Audio Branch processes log mel spectrograms, enabling strong performance across visual and audio-visual tasks including VQA, captioning, and question answering.
- **LLaMA-VID** [13] introduces an efficient Vision Language Model for processing long videos through a dual-token strategy. Each frame is encoded using two tokens: a context token derived from instruction-guided queries, and a content token generated by average pooling visual features from a pre-trained vision transformer. This compact representation enables the efficient processing of long videos while maintaining strong understanding capabilities through the effective transfer of long-context comprehension from language to vision domains.
- **TimeChat** [22] is a time-sensitive multimodal language model that combines a timestamp-aware frame encoder, sliding video Q-Former, and large language model. Processing up to 96 timestamped frames, it employs a sliding window approach to generate video tokens, which are integrated with optional speech transcripts and user queries. Instruction-tuned for temporal understanding, TimeChat enables precise event summarization, timestamp localization, and highlight detection in long videos.



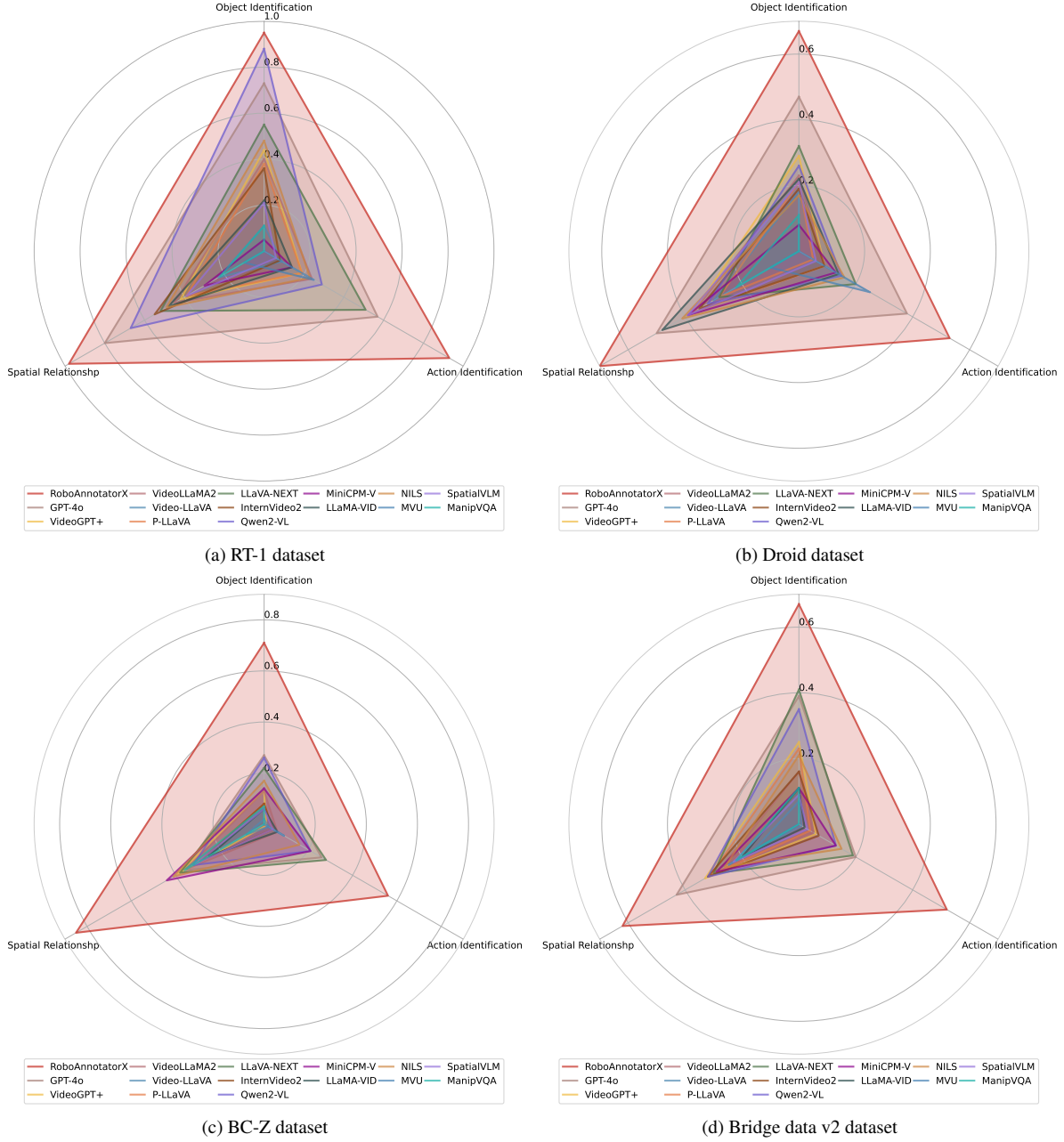


Figure 2. **Evaluation results on short-horizon datasets.** Radar charts comparing performance across three key dimensions: Object Identification, Spatial Relationship, and Action Identification.

## C.2. Robotic-related MLLM

- **MVU** [21] introduces an efficient framework for long-video understanding, featuring a novel Likelihood Selection technique for single-pass inference in multiple-choice tasks. The framework operates in three variants: pure LLM world knowledge, single-frame VLM processing, and full multimodal fusion, where object-centric information from pre-trained models is integrated through natural language. Using temporal downsampling to 16 frames and likelihood-based selection of 8 frames, MVU achieves state-of-the-art performance without video-level training while maintaining interpretability through language-based operations.
- **NILS** [2] presents a zero-shot framework for automatically labeling long-horizon robot data through a three-stage process:

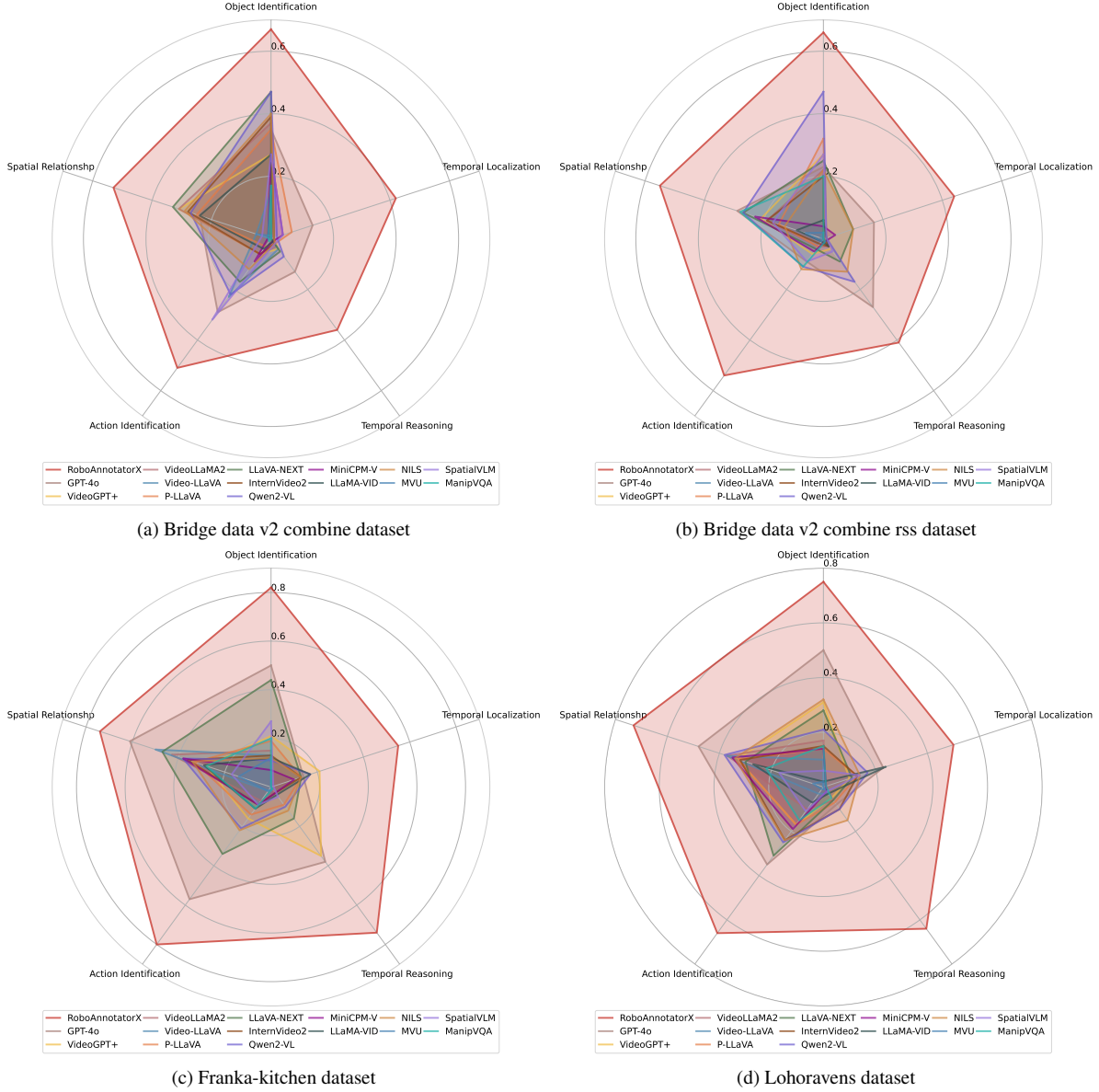


Figure 3. **Evaluation results on long-horizon datasets.** Radar charts comparing performance across five key dimensions: Object Identification, Spatial Relationship, Action Identification, Temporal Reasoning, and Temporal Localization.

object detection, object-centric change detection, and key state-based language label generation using LLMs. By leveraging pre-trained vision-language models and incorporating proprioceptive information, NILS outperforms existing approaches in key state detection and successfully annotated over 115k trajectories from 430+ hours of robot data across diverse environments.

- **SpatialVLM** [3] enhances vision language models’ 3D spatial reasoning capabilities through an Internet-scale data generation framework that creates 2 billion VQA examples from 10 million real-world images. The system processes CLIP-filtered scene images through pre-trained models to extract object-centric segmentation, depth, and captions, lifting 2D images into 3D point clouds for spatial property extraction. By training on this comprehensive spatial dataset and leveraging LLMs’ commonsense reasoning, the model achieves strong performance in both qualitative and quantitative spatial reasoning tasks, enabling novel applications in chain-of-thought spatial reasoning and robotics.
- **ManipVQA** [11] enhances MLLMs with manipulation-centric knowledge through a unified VQA framework, building

upon SPHINX architecture with LLaMA2 as its language backbone. The model combines CLIP’s local semantic features with Q-Former’s visual summarization and employs a sub-images patching strategy (448×448 image into four 224×224 patches) for detailed visual perception. Fine-tuned on a comprehensive dataset of interactive objects using a unified cross-entropy loss, ManipVQA integrates tool detection, affordance recognition, and physical concept understanding while maintaining general visual reasoning capabilities, demonstrating robust performance in both robotic simulators and vision benchmarks.

## D. Details of RoboX-VQA

### D.1. The Collection of Real World Demonstration

To train RoboAnnotatorX, we curate a comprehensive collection of real-world robotic manipulation datasets, as shown in Fig. 4. Our curation strategy prioritizes five critical dimensions: **video quality**, **robot diversity**, **object variety**, **task complexity**, and **annotation fidelity**, with particular emphasis on tabletop manipulation scenarios. The foundation of our dataset builds upon Open X-Embodiment [19, 33], which encompasses over 1 million real robot trajectories spanning 22 distinct robot platforms and demonstrating 527 unique skills across 160,266 task instances. To extend this foundation, we developed two additional long-horizon manipulation datasets: `bridge_data_v2.combine` (2K trajectories) and `bridge_data_v2.combine.rss` (1K trajectories), created through a systematic three-phase approach, as shown in Algorithm 1:

- **Data Concatenation:** Drawing inspiration from Bridge Data V2’s continuous demonstration collection methodology, we developed an advanced trajectory concatenation algorithm to construct extended, more sophisticated action sequences.
- **Continuity Verification:** Our algorithm implements rigorous continuity checks by computing visual similarity between consecutive trajectories. Specifically, it evaluates the visual correspondence between the terminal frame of the current sequence and the initial frame of the candidate sequence, employing a stringent similarity threshold of 0.9.
- **Adaptive Sequencing:** Based on the similarity analysis, the algorithm either: (a) integrates the new trajectory to form an extended, coherent demonstration when continuity is confirmed, or (b) finalizes the current sequence and initializes a new one when significant scene changes or task transitions are detected.

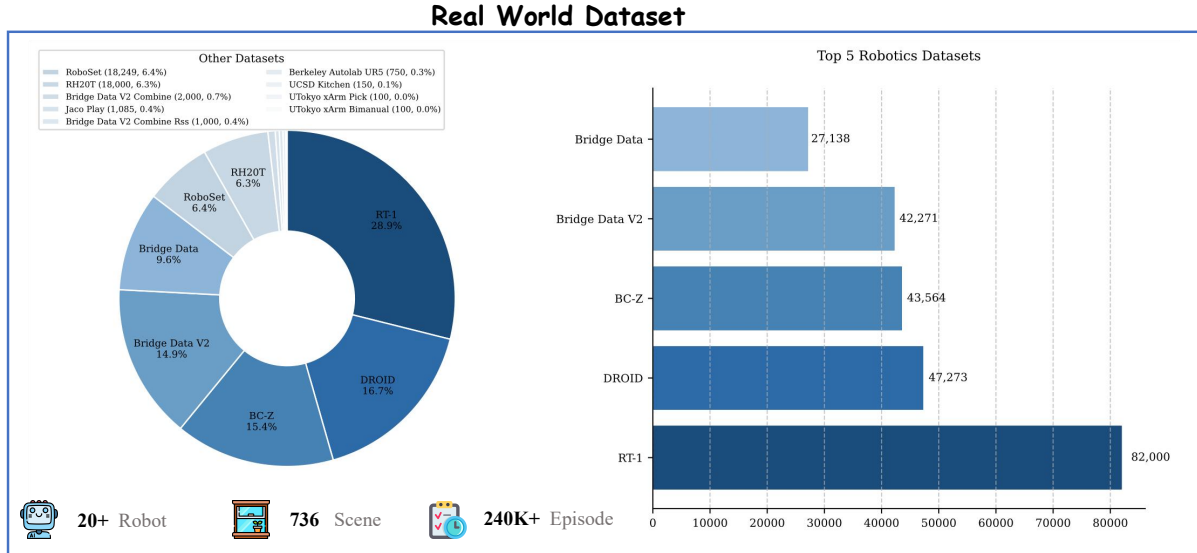


Figure 4. **Overview of real-world datasets used in our work.** All datasets focus on robotic manipulation tasks but vary in scale, setup, and specific task domains.

In addition to Open X-Embodiment, we incorporate RH20T [6], a large-scale multi-modal dataset containing over 110,000 robot manipulation sequences. We select approximately 20,000 trajectories across 20 representative tasks. While this combined dataset provides a robust foundation for our research, it has certain limitations. The data primarily consists of short-term sequences, which may not fully capture extended robotic operations. Additionally, the lack of fine-grained annotations limits our understanding of specific actions and sub-tasks. To address these constraints and enhance our model’s capabilities in un-

---

**Algorithm 1** Long-horizon Trajectory Concatenation

---

```
function CONCATENATELONGHORIZONTRAJECTORIES(trajectories, similarity_threshold = 0.9)
    ▷ Phase 1: Data Concatenation
    for each trajectory_group in Dataset do
        Sort trajectories by temporal order
        Initialize empty lists: concatenated_frames, instructions
        for each trajectory in sorted_trajectories do
            Extract current_frames and current_instruction

            ▷ Phase 2: Continuity Verification
            if concatenated_frames is not empty then
                similarity = ComputeVisualSimilarity(concatenated_frames[-1], current_frames[0])
                if similarity < similarity_threshold then
                    ▷ Check for scene discontinuity
                    if length(concatenated_frames) > 1 then
                        SaveLongHorizonSequence(concatenated_frames, instructions)
                    end if
                    Reset concatenated_frames and instructions
                end if
            end if

            ▷ Phase 3: Adaptive Sequencing
            if IsValidTrajectory(trajectory) then
                Extend concatenated_frames with current_frames
                Append current_instruction to instructions
            end if
        end for

        ▷ Process final sequence if valid
        if length(concatenated_frames) > 1 then
            SaveLongHorizonSequence(concatenated_frames, instructions)
        end if
    end for
end function
```

---

derstanding long-horizon tasks, we complement this dataset with simulated environments and advanced generative methods, which will be discussed in subsequent sections.

## D.2. The Collection of Simulator Demonstration

In this work, we address the critical challenge of data scarcity in long-horizon robotic learning through a comprehensive dual strategy. Our approach combines high-fidelity simulated environments [9, 16, 30] with advanced generative methods based on large language models to create a rich, diverse dataset. These simulated benchmarks provide a wide spectrum of activities, from elementary switch operations to intricate object manipulations, while the LLM-based task generation methods [10, 17, 25] procedurally create novel tasks and scenarios. This synergistic approach not only expands our task space but also introduces variations that challenge and improve model generalization, ultimately yielding a dataset suitable for training robust, generalizable models for complex, long-horizon robotic tasks.

- **Franka Kitchen** [9] presents a sophisticated simulation environment for evaluating robotic manipulation in household settings. As shown in Fig. 5, The benchmark features a 7-DoF Franka robot performing common kitchen operations, supported by 580 demonstration trajectories across 7 distinct sub-tasks.

As illustrated in Tab. 3, these tasks range from basic switch operations to complex object manipulations. This comprehensive setup evaluates the robot’s capacity to execute extended action sequences in realistic scenarios, testing its dexterity, spatial awareness, and interaction capabilities with various kitchen fixtures. The benchmark’s multi-task nature enables a thorough assessment of robotic performance in long-horizon, goal-oriented household environments.

- **CALVIN** [16] is an advanced benchmark for evaluating language-conditioned robotic manipulation in long-horizon sce-



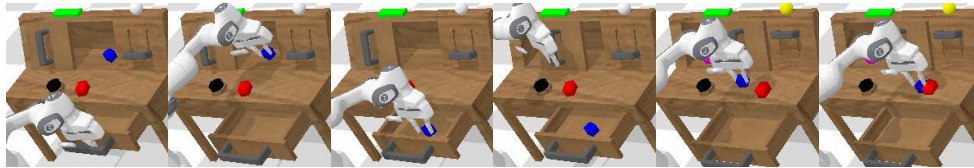
🔊 1)Open microwave->2)Move kettle to top left burner->3)Turn on oven knob->4)Open slide cabinet

Figure 5. **The example in Franka Kitchen.** Illustration of sequential tasks performed by the Franka robot, including opening microwave, moving kettle, turning on oven knob, and opening slide cabinet. These demonstrations showcase common household manipulation tasks in a simulated kitchen environment.

Table 3. **Franka Kitchen sub-tasks and descriptions.** A comprehensive list of manipulation tasks in the Franka Kitchen environment, detailing seven distinct sub-tasks for robot interaction with common kitchen appliances and objects. Each task represents a fundamental household operation requiring precise robot control and manipulation.

Task	Description
Bottom burner	Turn the oven knob that activates the bottom burner
Top burner	Turn the oven knob that activates the top burner
Light switch	Turn on the light switch
Slide cabinet	Open the slide cabinet
Hinge cabinet	Open the left hinge cabinet
Microwave	Open the microwave door
Kettle	Move the kettle to the top left burner

narios. As shown in Fig. 6, the environment features a simulated Franka robot arm operating at a desk equipped with interactive objects, including drawers, cabinets, a light switch, and colored blocks. CALVIN’s distinctive feature lies in its ability to chain multiple language instructions into complex, multi-step tasks.



🔊 1)Open drawer->2)Place in drawer->3)Open cabinet->4)Lift block->5)Push block right

Figure 6. **The example in CALVIN.** Demonstration of a multi-step manipulation sequence performed by the Franka robot arm, including: (1) opening a drawer, (2) placing an object in the drawer, (3) opening a cabinet, (4) lifting a block, and (5) pushing the block right. This sequence illustrates CALVIN’s capability to execute complex, language-guided tasks through sequential manipulations in a desk environment.

As detailed in Tab. 4, CALVIN encompasses diverse atomic actions ranging from object manipulation (rotation, pushing, lifting, placing) to fixture interaction (drawers, sliders, switches). These fundamental operations can be combined to form complex, long-horizon trajectories, enabling the evaluation of language-guided agents in realistic, multi-step household scenarios.

- **LoHoRavens [30]** is a comprehensive benchmark dataset designed to evaluate the capabilities of language-conditioned robots in long-horizon tasks. Built upon the Raven’s robot simulator, this dataset presents a diverse array of challenges that test the reasoning and planning abilities of embodied agents, particularly those leveraging large language models (LLMs). As shown in Fig. 7 The benchmark comprises ten meticulously crafted tasks, each designed to assess different aspects of robotic manipulation and decision-making. These tasks are broadly categorized into three main types: *Move*, *Sort*, and *Matching*.

- **Move:** Robots are required to position blocks in specific locations, which may be defined in absolute terms or relative to other objects.

Table 4. **Overview of tasks in CALVIN.** A comprehensive list of manipulation primitives supported in the CALVIN environment, encompassing basic object interactions (rotating, pushing, lifting), container operations (drawer, cabinet, slider), and device controls (lightbulb, LED). Each task represents a fundamental action that can be combined to form complex, multi-step sequences.

Task	Description
Rotate block right/left	Rotate block 90 degrees right or left
Push block right/left	Push block to the right or left
Move slider right/left	Slide door to the right or left
Open/Close drawer/cabinet	Open or close the drawer or cabinet
Lift block from table/slider	Lift block from table or slider
Place in slider/drawer	Put grasped object in slider or drawer
Push into drawer	Push object into the drawer
Stack/Unstack block	Stack blocks or remove top block from stack
Turn on/off lightbulb	Toggle switch to turn light on or off
Turn on/off LED	Push button to turn green light on or off

- \* *MoveBlocktoArea*: Move all the blocks to the {abs\_area}.
- \* *MoveColorBlocktoArea*: Move all the {color} blocks to the {abs\_area}.
- \* *MoveBlockinAreatoArea*: Move all blocks in {abs\_area} to {abs\_area}.
- \* *MoveSizeBlocktoCorner*: Move all {size} blocks to {position} corner.
- **Sort**: Robot must demonstrate categorization and grouping skills. The challenge involves sorting various objects, such as blocks or bowls, based on shared attributes.
  - \* *StackAllBlocks*: Stack all the blocks together.
  - \* *StackBlocksofSameSize*: Stack all the blocks of the same size.
  - \* *StackBlocksofSameColor*: Stack all the blocks of the same color.
  - \* *StackColorBlockstoArea*: Stack all blocks of primary color on the left side.
- **Matching**: Robots need to pair blocks with appropriate bowls based on certain criteria.
  - \* *PutBlockInMatchingBowl*: Stack all the blocks together.
  - \* *PutBlockInMismatchingBowl*: Stack all the blocks of the same size.
  - \* *PutBlockinZonewithMatchingColor*: Put blocks of the same color in the zone with the matching color.

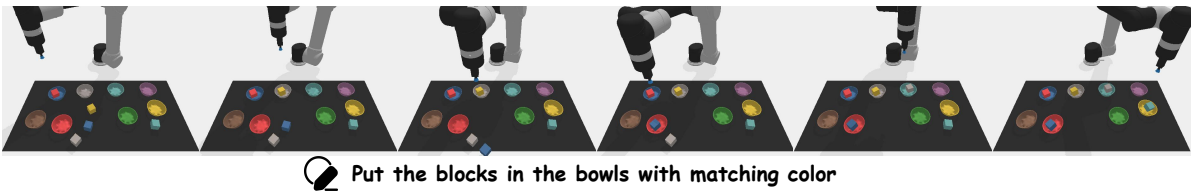


Figure 7. **The example in LoHoRavens.** Demonstration of a matching task where the robot needs to place blocks into bowls with corresponding colors, showcases the system’s ability to understand color-based relationships and execute precise manipulation actions.

- **Generative Simulation** Recent approaches like RoboGen [25], RoboCasa [17], and GenSim2 [10] address the data scarcity problem in robotic skill learning while expanding the diversity of long-horizon tasks. These methods employ LLMs to create complex, multi-step instructions that combine multiple subtasks, significantly expanding the diversity of long-horizon scenarios. By integrating predefined simulated environments like Maniskill [8, 23] and Robosuite [34] with generative techniques, these approaches utilize simulator oracle engines or pre-trained policies to efficiently collect large-scale demonstrations. The collected data includes rich ground truth metadata, such as precise state information and action primitives, which not only enhances model robustness but also facilitates subsequent question-answering construction. This innovative process effectively bridges the gap between simulated environments and practical applications while enabling more diverse and comprehensive robotic learning scenarios.



### D.3. The Meta-Information of Demonstration

We propose a novel approach to enhance the visual understanding and reasoning capabilities of models by leveraging meta-information from demonstrations. Meta-information refers to supplementary descriptive information associated with video data, providing richer context and details that facilitate deeper learning and reasoning. We categorize meta-information into two types based on different data sources. For the Coarse-labeled Real World Dataset, we employ Macro-Scene Descriptions (MSD) as meta-information. MSD are coarse-grained language annotations inherent to real-world robotic demonstrations, describing the actions occurring in the video and providing a high-level overview of real-world semantic information. For the Versatile Long-horizon Simulation Dataset, we introduce Micro-Temporal Instructions (MTI) as meta-information. The generation of MTI is closely integrated with the initialization of the simulated environment: the system automatically records the assets used and annotates their corresponding attributes. This method allows us to precisely capture the attributes (such as color, size) and spatial relationships of objects manipulated in each subtask, without the need for additional manual annotation. This dual structure ingeniously combines the complexity of real robotic interactions with the precision of simulated data, establishing a robust and comprehensive foundation for model training. The MSD provides authentic robotic manipulation scenarios with natural variations, while the MTI offers detailed and accurate annotations across long-horizon tasks in simulation. Please see Appendix J for more details.

### D.4. Automatic Question Answering Generation

To maximize the value of collected meta-information and further enhance model capabilities, we introduce an automated question-answering (QA) generation framework that transforms unstructured video demonstrations into structured learning signals. As depicted in Fig. 8, our framework leverages GPT-4o to automatically generate diverse and semantically meaningful QA pairs by integrating meta-information from both real-world scenarios (through Macro-Scene Descriptions, MSD) and simulated environments (via Micro-Temporal Instructions, MTI). This automated process not only deepens the model’s comprehension of key information and implicit relationships in videos but also strengthens its reasoning and generalization abilities across multiple dimensions of understanding.

The core of our framework lies in its carefully designed prompting strategy and comprehensive question category system. Our prompting approach guides GPT-4o to generate contextually relevant QA pairs by combining rich meta-information with predefined question categories, ensuring the generated questions effectively probe different aspects of understanding while maintaining strong alignment with the learning objectives. In the following sections, we first detail our prompt design for QA generation, which enables structured and consistent outputs. We then present the RoboX-VQA Categories Distribution, which

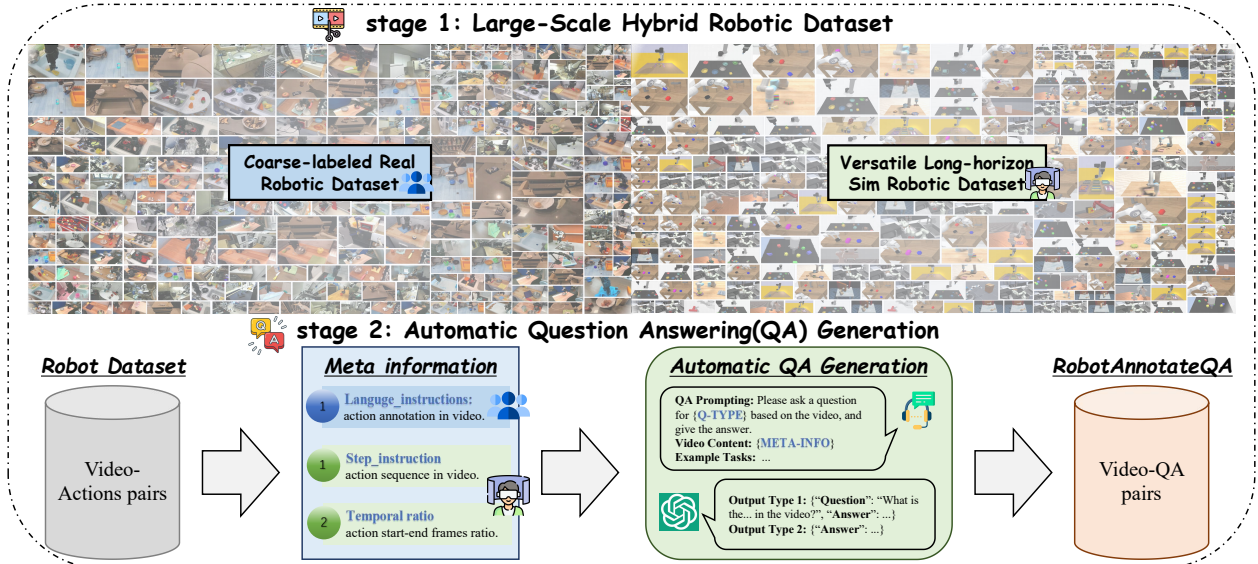


Figure 8. **Overview framework for automatic RoboX-VQA generation.** Our framework leverages dual-source meta-information to generate comprehensive question-answer pairs. These meta-information sources are fed into GPT-4o, which, guided by carefully designed prompts, generates diverse and semantically meaningful QA pairs. This automated process creates structured learning signals that enhance models’ visual understanding and reasoning capabilities across multiple dimensions.

systematically covers various dimensions of video understanding, from scene understanding to temporal understanding.

**RoboX-VQA Categories Distribution** The generated RoboX-VQA is systematically organized into two main categories: Scene Understanding and Temporal Understanding, each meticulously designed to evaluate specific aspects of model comprehension, as illustrated in Fig. 9:

- **Scene Understanding:** encompasses two fundamental aspects:
  - **Object Identification:** This component focuses on type recognition, status assessment, and function analysis of objects. Examples include "What type of object is being manipulated?" or "What is the current state of the target object?" Such questions evaluate the model's ability to comprehend object properties and their functional roles.
  - **Spatial Relationship:** This category evaluates both relative position and position change between objects. Questions like "Where is the object positioned relative to the table?" or "How has the object's position changed after the action?" assess the model's spatial reasoning capabilities.
- **Temporal Understanding:** comprises three key components:
  - **Action Understanding:** Evaluates action identification and action-object interactions. Questions such as "How does the robot interact with the object?" and "What is the main action being performed?" help assess the model's comprehension of action patterns and their effects.
  - **Temporal Localization:** Focuses on boundary detection, and dense video captioning. Examples include "At which timestamp does the manipulation action begin?" or "What is happening throughout this video segment?"
  - **Temporal Reasoning:** Addresses segment summarization, task abstraction and action ordering. Questions like "What is the sequence of actions performed?" or "How can this task be abstracted into higher-level steps?" evaluate the model's ability to understand temporal dynamics.

By incorporating these diverse question types, our QA generation method significantly enhances the model's capabilities in several ways. First, it forces the model to engage in **multi-level reasoning**, improving its ability to handle complex queries. Second, it helps bridge the gap between visual perception and language understanding, crucial for **human-robot interaction**. Third, the structured nature of QA pairs allows for **more efficient learning** compared to raw video data, enabling the model to quickly grasp key concepts and relationships. Lastly, by covering a wide range of scenarios and question types, this

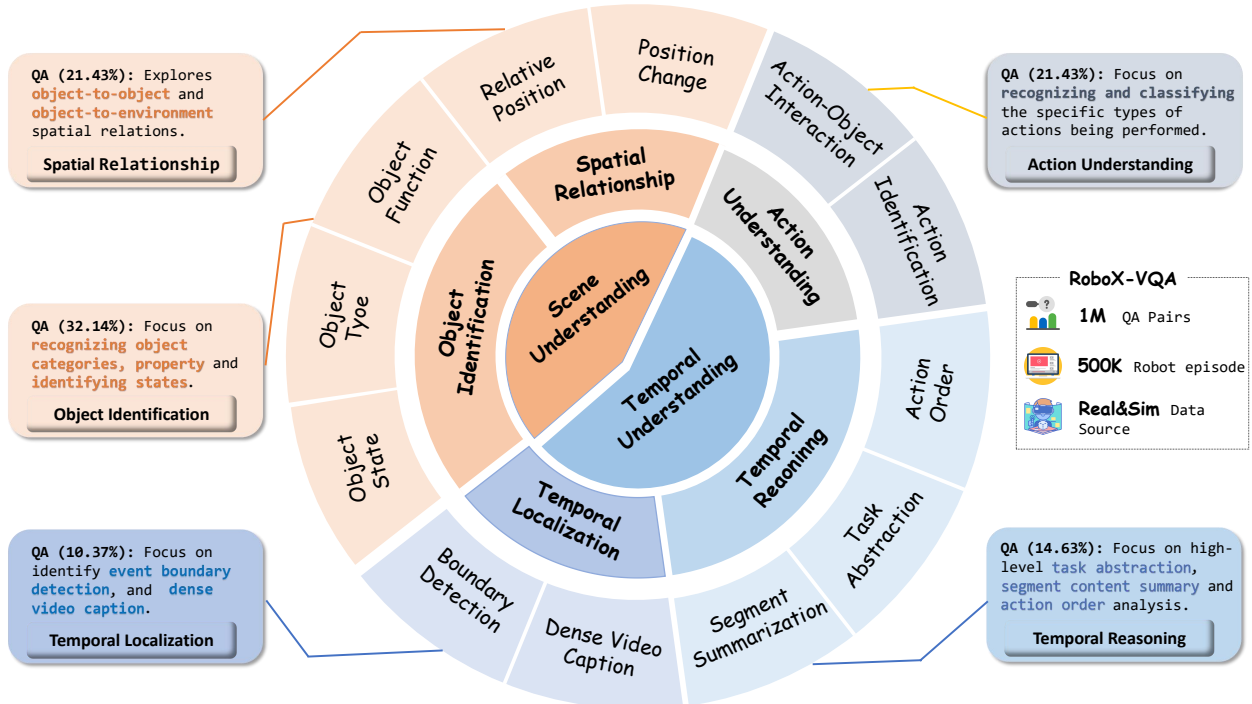


Figure 9. **RoboX-VQA distribution overview.** Hierarchical categorization of our dataset's question-answer pairs across two main dimensions. Scene Understanding evaluates object identification and spatial relationships, while Temporal Understanding assesses action understanding, temporal localization, and temporal reasoning. This comprehensive framework ensures a thorough evaluation of models' visual-temporal comprehension capabilities.



approach enhances the model’s generalization ability, making it more robust in handling diverse real-world situations.

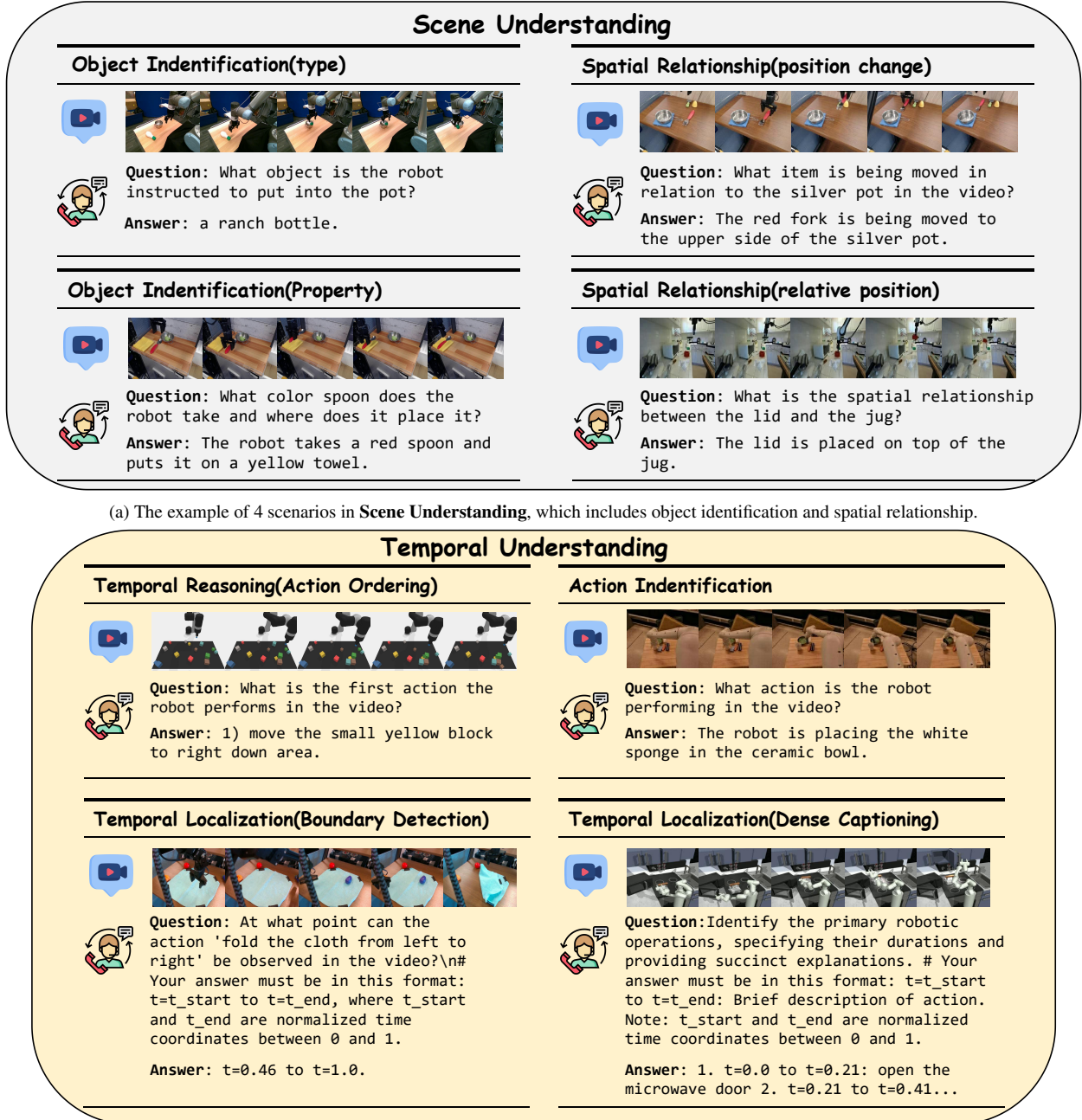


Figure 10. **Examples of different understanding scenarios.** Visualization of our dataset’s two main evaluation categories through representative examples. Each example is accompanied by question-answer pairs that assess specific aspects of model comprehension.

**Prompt for QA Generation** Below we present the systematic prompt design for QA generation using GPT-4o, which enables the creation of diverse and meaningful question-answer pairs from robotics demonstrations. As illustrated in Fig. 11, our carefully crafted prompt template incorporates three key components: the question type specification that guides the model to focus on specific aspects of video understanding, the structured meta-information that provides rich contextual details about the video content, and the standardized JSON format that ensures consistent and machine-readable outputs.

You are an expert in robotics, video understanding, and data labeling. Now I want you to help me write question-and-answer pairs based on the information provided.

Please ask a question for {Question Type} in the field of video understanding based on the provided {Video Content}. Then you need to give the answer.

# The video depicts a robot performing tasks on a tabletop.

# Only describe what you are certain about, and avoid providing descriptions that may be ambiguous or inaccurate.

Input Format:

1. Question Type: Question type specification

2. Video Content: Meta information about the video

Output Format:

Your response must be in JSON format, with the keys question and answer, like this: {question: your question here, answer: your answer here}

Figure 11. **Prompt template for question-answering generation.** The template consists of three main components: (1) Question type specification, (2) Video content description through meta-information, and (3) Response format requirements in JSON.

## E. Details of Training

Robotic demonstration understanding requires a gradual progression from basic scene awareness and coarse-grained action identification to fine-grained spatiotemporal reasoning. Direct training often struggles with complex temporal dependencies and task-specific knowledge acquisition. To address this, we introduce a curriculum-inspired three-stage training paradigm that progressively develops the capabilities through carefully designed data composition and module configurations, as shown in Algorithm 2. Building upon these considerations and our previously described automated question-answering generation method, we designed a systematic training pipeline aimed at progressively enhancing the model’s vision-language understanding capabilities. This training pipeline adopts a curriculum learning approach, starting from basic cross-modal alignment and gradually transitioning to complex long-horizon robotic manipulation comprehension. As shown in Tab. 5 and Tab. 6, we carefully designed the module training configurations and hyperparameters across different stages to ensure effective knowledge integration. The training process utilized an NVIDIA A800 GPU with 80GB of VRAM. The model training was conducted over a total of 300 hours, including all pre-training and fine-tuning stages.

### E.1. Stage 1: General Pretraining

We conduct vision-language alignments in stage 1 using captioning-based pertaining datasets, consisting of 558K image-text pairs and 510K video-text pairs from robotic manipulation scenarios. To ensure data diversity and representativeness, we carefully curated high-quality datasets including large-scale collections such as Open X-Embodiment, along with other widely-adopted datasets like DROID and RH20T. For the model training strategy, we adopt a selective module training approach: keeping the visual encoder, time-aware encoder, global transformer, temporal Q-former, and LLM frozen, while only training the scene-level, clip-level, and video-level projectors to efficiently establish cross-modal mapping relationships. To optimize the training process, we employ DeepSpeed ZeRO-2 for training acceleration, utilizing the AdamW optimizer with an initial learning rate of  $1e-3$  and a warmup ratio of 0.03. In terms of training configuration, we use a batch size of 32, train for 1 epoch, and set the maximum token length to 2048. These carefully calibrated training parameters ensure efficient and stable learning of vision-language representations. The model training was conducted over a total of 109 hours.

### E.2. Stage 2: Short-Horizon Fine-tuning

In the second stage, we conduct short-horizon instruction fine-tuning leveraging 227K image-instruction pairs and 886K short video clips containing atomic robotic actions. Building upon the general visual understanding foundation from stage 1, we advance from basic robotics scene understanding to specific action primitive comprehension, emphasizing precise recognition of manipulation primitives and detailed robot-object interaction patterns. Regarding the model training strategy, we adopted

Table 5. **Module training configuration across different stages.** Detailed overview of module training states (Freeze/Open) during three sequential stages: general pretraining, short-horizon fine-tuning, and long-horizon fine-tuning. The configuration shows which components remain frozen and which are trainable across different training phases, with LoRA adaptation introduced in the fine-tuning stages.

	General Pretraining	Short-Horizon Fine-tuning	Long-Horizon Fine-tuning
Visual Encoder	Freeze❄	Freeze❄	Freeze❄
Time-aware Encoder	Freeze❄	Trainable🔥	Trainable🔥
Global Transformer	Freeze❄	Trainable🔥	Trainable🔥
Temporal Q-former	Freeze❄	Trainable🔥	Trainable🔥
Scene-level Projector	Trainable🔥	Trainable🔥	Trainable🔥
Clip-level Projector	Trainable🔥	Trainable🔥	Trainable🔥
Video-level Projector	Trainable🔥	Trainable🔥	Trainable🔥
LLM	Freeze❄	Freeze❄	Freeze❄
LoRA	None	Trainable🔥	Trainable🔥

Table 6. **Hyperparameter settings across different training stages.** We adopt AdamW optimizer for all stages with DeepSpeed ZeRO -2. LoRA is enabled for both short-horizon and long-horizon fine-tuning stages with rank 64 and alpha 16. The model is trained on 4 NVIDIA A800 GPUs with a maximum sequence length of 2048 tokens.

	General Pretraining	Short-Horizon Fine-tuning	Long-Horizon Fine-tuning
DeepSpeed Stage	2	2	2
Optimizer	AdamW	AdamW	AdamW
Epoch	1	1	1
Batch Size	32	32	16
Learning Rate	1e-3	2e-5	2e-5
Warmup Ratio	0.03	0.03	0.03
Weight Decay	0	0	0
Max Token	2048	2048	2048
LoRA Enable	False	True	True
LoRA R	-	64	64
LoRA Alpha	-	16	16
LoRA Dropout	-	0.05	0.05
LoRA Bias	-	None	None
Training Device	A800 × 4	A800 × 4	A800 × 4
Training Duration	109H	223H	46H

a more open-module training approach. While keeping the visual encoder and LLM frozen, we enabled the training of the time-aware encoder, global transformer, temporal Q-former, scene-level projector, clip-level projector, and video-level projector. Additionally, we introduced LoRA technology for efficient fine-tuning to enhance the model’s ability to specific action primitive comprehension. For training configuration, we continued using DeepSpeed ZeRO-2 for acceleration with the AdamW optimizer but reduced the learning rate to 2e-5 to ensure fine-tuning stability. The training process continued for 1 epoch, maintaining a batch size of 32 and a maximum token length of 2048, with a warmup ratio of 0.03. This refined parameter configuration enabled the model to effectively learn fine-grained dynamic features in short-horizon scenarios while maintaining continuity with the foundational knowledge acquired during pretraining. The model training was conducted over a total of 223 hours.

### E.3. Stage 3: Long-Horizon Fine-tuning

In the third stage, we conduct long-horizon instruction fine-tuning leveraging 86K extended robotic demonstrations of complex, multi-step operations, as shown in Fig. 12. Building upon the atomic action recognition capabilities developed in stage 2, we advance from single-action comprehension to sophisticated temporal understanding. Following the same module training strategy as stage 2, we keep the visual encoder and LLM frozen while enabling the training of the time-aware encoder, global transformer, temporal Q-former, scene-level projector, clip-level projector, and video-level projector. We continued

utilizing LoRA technology for parameter-efficient fine-tuning to accommodate the specific requirements of long-horizon scenarios. Considering the complexity and memory consumption of long-sequence data, we adjusted the batch size to 16 while maintaining other key hyperparameters, including DeepSpeed ZeRO-2 acceleration, AdamW optimizer, learning rate of  $2e-5$ , warmup ratio of 0.03, and maximum token length of 2048. The training continued for 1 epoch, ensuring thorough learning of long-term dependencies. This carefully calibrated training strategy enables the model to effectively handle more complex multi-step task sequences while maintaining training stability. The model training was conducted over a total of 46 hours.

Through this progressive training strategy, our model has demonstrated significant performance improvements. From establishing fundamental vision-language mappings through large-scale pretraining to mastering basic action understanding through short-horizon instruction fine-tuning, and finally achieving complex task sequence interpretation through long-horizon instruction fine-tuning, each stage has made crucial contributions to the model’s capability enhancement. This training pipeline not only ensures stable and effective learning but also enables the model to gradually adapt to more challenging tasks through carefully designed module training strategies and parameter configurations while maintaining foundational knowledge.

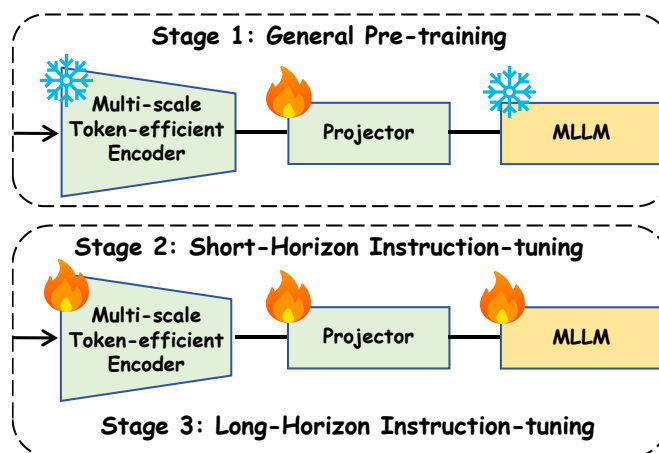


Figure 12. **Module training configuration.** Diagram showing module freeze/open status across three stages: General Pretraining, Short-Horizon Fine-tuning, and Long-Horizon Fine-tuning.

## F. Details of Robotic Downstream Task

### F.1. Evaluation Details

Our experimental evaluation was conducted on two platforms: the simulated LohoRavens and a real-world robotic setup using the xArm 6-DOF manipulator. To ensure a rigorous and fair comparison, we maintained consistent policy architecture across all methods within each environment. In the LohoRavens simulation, all methods leveraged OpenVLA [12] as the underlying policy model, benefiting from its robust visual-language capabilities. For real-world experiments with the xArm manipulator, all methods utilized Octo [18], chosen for its proven effectiveness in translating language instructions into physical actions in unstructured environments. We employed three comprehensive metrics to thoroughly evaluate performance:

- **Planning Accuracy:** Measures the robot’s ability to generate correct and coherent action plans that align with given instructions, reflecting the system’s reasoning capabilities.
- **Grounding Accuracy:** Evaluates whether the policy correctly understands language instructions by associating them with appropriate objects, spatial relationships, and temporal sequences of actions in the environment.
- **Success Rate:** Represents the percentage of tasks completed successfully from start to finish, serving as the ultimate measure of end-to-end performance in realistic scenarios.

### F.2. LohoRavens Experiment

For the LohoRavens experiments, we collected demonstrations spanning multiple task types across the benchmark’s three main categories: Move, Sort, and Matching. Our dataset encompassed diverse challenges including object rearrangement, container interaction, multi-step manipulation, and conditional tasks. All demonstrations were annotated using both our proposed method and established baseline approaches under identical conditions. Our algorithm’s fine-grained annotations proved particularly advantageous for complex long-horizon tasks, where decomposing intricate demonstrations into executable sub-instructions significantly enhanced policy learning efficiency. This hierarchical approach effectively bridged the gap between high-level planning and low-level execution, which was especially beneficial for the more complex tasks within the LohoRavens benchmark that require sophisticated reasoning, temporal sequencing, and spatial planning capabilities.

### F.3. Real Robot Experiment

For the real robot experiments with the xArm manipulator, We collected 150 high-quality demonstrations across 15 household manipulation tasks, carefully designed to represent everyday scenarios including pick and place, container manipulations, open and close, and sequential multi-step manipulation. Representative task examples are visualized in Fig. 13. The real robot experiments presented additional challenges due to physical constraints, perception noise, and variations in object appearance. Despite these challenges, our method demonstrated significantly superior performance in generating contextually grounded annotations with enhanced spatial-temporal understanding. This translated to measurable improvements in both grounding accuracy and task success rates compared to the best-performing baseline methods.

The robust performance across both simulated and real-world environments confirms that our annotation approach effectively enables more efficient and effective policy training for language-conditioned robotic manipulation tasks.

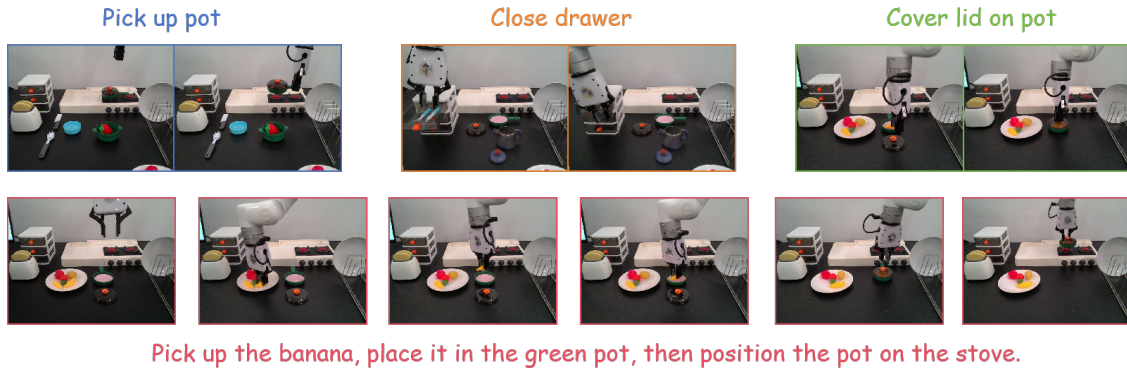


Figure 13. **Task used for evaluation in real-world.** Representative household manipulation tasks performed by the xArm manipulator in our real-world experiments. The tasks showcase various manipulation scenarios including pick-and-place operations, container interactions, and sequential multi-step manipulations that require accurate spatial-temporal understanding.

## G. Quick Guideline of Usage

Our framework enables the generation of **comprehensive annotations** through **diverse prompts**, as illustrated in the example below as shown in Fig. 14. RoboAnnotatorX supports a wide spectrum of annotation capabilities, from fundamental scene understanding to sophisticated temporal reasoning across long-horizon demonstrations. The framework is designed to be **extensible**, allowing researchers to easily incorporate new prompting templates and annotation schemes as needed. This **flexibility** is particularly valuable given the current challenges in robotics dataset annotation, where traditional methods often struggle with scalability and consistency across complex demonstrations. By leveraging multimodal large language models with our multi-scale token-efficient encoder, RoboAnnotatorX provides a reliable foundation for generating rich, context-aware annotations that can help unlock the full potential of existing robotic demonstration datasets.

```
from Roboannotatorx import load_roboannotatorx

# Example Prompt Library
prompt_library = [
# 1. Scene Understanding
## 1.1 Object Identification
"What objects are involved in the tasks performed by the robot in the video?",
"What is the first item that the robot interacts with in the video?",
"What object does the robot take and where does it put it according to the video?"
...

## 1.2 Spatial Relationship
"What is the spatial relationship between the big spoon and the tray during the robot's task?",
"Where is the orange towel moved to in relation to the table?",
"Where is the yellow object moved to in the video?",
...

#2. Temporal Understanding
##2.1 Action Identification
"What specific action is the robot performing in the video?",
"What action is the robot performing in the video?",
"What action is the robot performing in relation to the carrot on the plate?",
...

## 2.2 Temporal Localization
"Can you identify the time in the video that matches the action 'move slider right'?",
"Outline the main robotic tasks performed in the video, including their temporal boundaries and summaries.",
"Analyze the video to extract key robotic maneuvers, detailing their time ranges and basic characteristics.",
"Analyze the video to extract key robotic maneuvers, detailing their time ranges and basic characteristics.",
...

## 2.3 Temporal Reasoning
"What is the first action performed by the robot in the video?",
"What is a short, precise description of the action occurring from t=0.64 to t=0.7?",
"What is the final location of the blue fork after all the actions are performed?",
"What is the high-level task being described in the video?",
...
]

roboannotatorx = load_roboannotatorx(llm_backbone)
video= load_video('video_path')
prompt = load_prompt(prompt_list, 'question_type')
annotation = roboannotatorx(video=video, prompt=prompt)
```

Figure 14. **Quick usage guideline.** Example Python script demonstrating how to use RoboAnnotatorX for comprehensive video-based annotation. The prompt library includes two main categories: (1) Scene Understanding with subcategories for object identification and spatial relationships, and (2) Temporal Understanding covering action identification, temporal localization, and temporal reasoning. The code snippet shows the straightforward API for loading the model and generating annotations using diverse prompts.



## H. Pesudocodes of Framework

---

### Algorithm 2 The training of RoboAnnotatorX

---

**Input:** Training datasets  $\mathcal{D}_{pre}, \mathcal{D}_{short}, \mathcal{D}_{long}$

**Parameters:** Visual Encoder  $E_{img}$ , Time-aware Encoder  $TE$ , Temporal Q-Former  $TQ$ , Global Transformer  $GT$ , Projectors  $P_s, P_c, P_v$ , Language Model  $L$

#### Stage 1: General Pretraining

```

for epoch  $e = 1$  to  $E_{pre}$  do
  for each batch  $(V, t) \in \mathcal{D}_{pre}$  do                                 $\triangleright V \in \mathbb{R}^{T \times H \times W \times 3}$  is a video
    // Scene-level token stream
     $F_{keyframe} = \{F_0\} \cup \{F_i \mid i \in [1, 2, \dots, K-2] \times I\} \cup \{F_{T-1}\}$ 
     $S_{scene} = \text{GridPool}(\text{Encoder}_{img}(F_{keyframe}))$                  $\triangleright S_{scene} \in \mathbb{R}^{(K \times N_s) \times d}$ 
    // Clip-level token stream
    for each clip  $C_i$  extracted from  $V$  do
       $F_{clip}^t = \text{TE}(C_i)$                                            $\triangleright F_{clip}^t \in \mathbb{R}^{l_i \times N_i \times d}$ 
       $S_{clip}^i = \text{QFormer}(F_{clip}^t)$                                  $\triangleright S_{clip}^i \in \mathbb{R}^{N_c \times d}$ 
    end for
    // Video-level token stream
     $V^t = \text{AvgPool}_{N_i}(\text{TE}(V))$                                    $\triangleright V^t \in \mathbb{R}^{T \times d}$ 
     $S_{video} = \text{GlobalTransformer}(V^t)$                              $\triangleright S_{video} \in \mathbb{R}^{T \times d}$ 
    // Combine token streams for LLM input
     $V_{LLM} = \text{Concat}(S_{scene}, \{S_{clip}^i\}_{i=1}^I, S_{video})$          $\triangleright V_{LLM} \in \mathbb{R}^{L \times d}$ 
    // Update model parameters
    Update parameters using loss  $\mathcal{L}_{pre}(L(V_{LLM}), t)$ 
  end for
end for

```

#### Stage 2: Short-Horizon Instruction Fine-tuning

```

for epoch  $e = 1$  to  $E_{short}$  do
  for each batch  $(V, I, q, a) \in \mathcal{D}_{short}$  do                     $\triangleright I$ : instruction,  $q$ : question
    Extract hierarchical features as in Stage 1:  $V_{LLM} = \text{Concat}(S_{scene}, \{S_{clip}^i\}_{i=1}^I, S_{video})$ 
     $\hat{a} \leftarrow L(V_{LLM}, I, q)$                                      $\triangleright$  Generate answer
    Update parameters with loss  $\mathcal{L}_{short}(L(V_{LLM}, I, q), a)$ 
  end for
  Evaluate on validation set  $\mathcal{D}_{short}^{val}$  and save checkpoints
end for

```

#### Stage 3: Long-Horizon Instruction Fine-tuning

```

for epoch  $e = 1$  to  $E_{long}$  do
  for each batch  $(V, I, \{q_i\}, \{a_i\}) \in \mathcal{D}_{long}$  do           $\triangleright$  Multiple QA pairs
    Extract hierarchical features as in Stage 1:  $V_{LLM} = \text{Concat}(S_{scene}, \{S_{clip}^i\}_{i=1}^I, S_{video})$ 
    for each question-answer pair  $(q_i, a_i)$  in instruction  $I$  do
       $\hat{a}_i \leftarrow L(V_{LLM}, I, q_i)$                              $\triangleright$  Generate answer
      Accumulate loss  $\mathcal{L}_{long}(L(V_{LLM}, I, q_i), a_i)$ 
    end for
    Update parameters with accumulated loss
  end for
  Evaluate on validation set  $\mathcal{D}_{long}^{val}$  and save checkpoints
end for

```

**Output:** Trained model parameters for hierarchical video understanding

---

## I. More Visualization










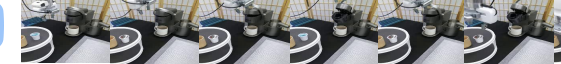

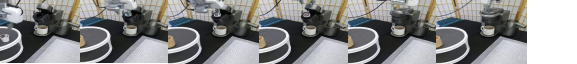
  <p><b>Question:</b> What object is the robot instructed to put into the pot? <b>Answer:</b> a ranch bottle.</p> <p><b>Pred:</b> a ranch bottle.</p> <p><b>Pred:</b> The robot is instructed to put a bottle with a green cap into the pot.</p> <p><b>Pred:</b> The robot is instructed to put the green and white object into the pot.</p>	  <p><b>Question:</b> What action is the robot performing in the video? <b>Answer:</b> The robot is placing the white sponge in ceramic bowl.</p> <p><b>Pred:</b> The robot put the white sponge into a bowl.</p> <p><b>Pred:</b> The robot appears to be pouring liquid from a bottle into a bowl or container.</p> <p><b>Pred:</b> The robot arm picks up the white cup from the table and puts it into the green bowl.</p>
  <p><b>Question:</b> In the video, where is the egg placed before being put on the stove? <b>Answer:</b> The egg is placed in a pot or pan.</p> <p><b>Pred:</b> The egg is placed in the pot.</p> <p><b>Pred:</b> The egg is initially positioned directly on the burner.</p> <p><b>Pred:</b> The egg is placed on the stove before being put on the stove.</p>	  <p><b>Question:</b> Which time segment of the video shows the robot engaged in 'put the green figure on the top edge of the cloth'? <b>Answer:</b> t=0.51 to t=1.0.</p> <p><b>Pred:</b> From t=0.51 to t=1.0.</p> <p><b>Pred:</b> The robot engaged in "put the green figure on the top edge of the cloth" is from t=-0.333 to t=-0.667.</p> <p><b>Pred:</b> t=0.00 to t=0.33.</p>
  <p><b>Question:</b> Identify separate robotic tasks, including timestamps and brief descriptions. <b>Answer:</b> 1. t=0.0 to t=0.16: Open machine lid 2. t=0.16 to t=0.57: Insert coffee pod into machine 3. t=0.57 to t=0.72: Close machine lid 4. t=0.72 to t=1.0: Press brew button</p> <p><b>Pred:</b> 1. t=0.0 to t=0.13: Open machine lid 2. t=0.13 to t=0.52: Insert coffee pod into machine 3. t=0.52 to t=0.62: Close machine lid 4. t=0.62 to t=1.0: Press brew button</p> <p><b>Pred:</b> t=0.0 to t=0.4: The robotic arm picks up a coffee pod. t=0.4 to t=0.5: The coffee machine lid is opened. t=0.5 to t=0.7: The robotic arm positions the coffee pod above the machine. t=0.7 to t=1.0: The coffee machine lid is closed and the robotic arm moves away.</p> <p><b>Pred:</b> t=0 to t=0.25: A robotic arm is seen moving a cup towards a coffee machine. t=0.25 to t=0.75: A robotic arm is seen moving a cup towards a coffee machine. t=0.75 to t=0.95: The arm is seen moving a coffee cup towards a coffee machine.</p> <p><b>Question:</b> What is the first action the robot performs in the video? <b>Answer:</b> Open machine lid.</p> <p><b>Pred:</b> Open machine lid.</p> <p><b>Pred:</b> The robotic arm first picks up a coffee pod.</p> <p><b>Pred:</b> Moving a cup towards a coffee machine.</p>	  <p><b>Question:</b> What is the high-level task being described in the video? <b>Answer:</b> make coffee.</p> <p><b>Pred:</b> The robot's sequential arrangement of a coffee matching, and coffee pod suggests making coffee.</p> <p><b>Pred:</b> Preparing for brewing coffee.</p> <p><b>Pred:</b> Moving a cup towards a coffee machine.</p>

Figure 15. **More visualization.** Comparative analysis of scene understanding and temporal reasoning capabilities across RoboAnnotatorX, GPT-4o, and Qwen2-VL models. The comparison showcases how different models interpret and reason about visual inputs, temporal relationships, and action sequences.








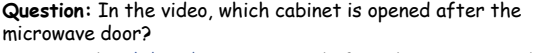







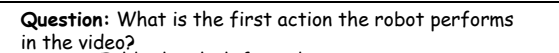

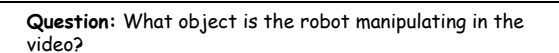
  <p><b>Question:</b> What is the sequence of actions performed by the robot in the video?</p> <p><b>Answer:</b> The robot performs the following actions in sequence: 1) opens the <b>microwave door</b>, 2) turns the <b>oven knob</b> that activates the bottom burner, 3) opens the <b>slide cabinet</b>, and 4) opens the <b>left hinge cabinet</b>.</p> <p><b>Pred:</b> The robot first opens the <b>microwave door</b>, then turns the <b>oven knob</b>, followed by opening <b>slide cabinet</b>, and finally opens the <b>left hinge cabinet</b>.</p> <p><b>Pred:</b> The robot appears to be preparing to <b>open a cupboard</b>, then opens the <b>microwave door</b> and <b>places the blue object inside the microwave</b>.</p> <p><b>Pred:</b> The robot in the video performs a sequence of actions, including opening the <b>oven door</b>, <b>removing the bread from the oven</b>, and <b>placing it on the counter</b>.</p>	  <p><b>Question:</b> What task is the robot performing in the video involving the white object?</p> <p><b>Answer:</b> The robot is picking up the <b>white object</b> from the table and <b>hanging it to the metallic object</b>.</p> <p><b>Pred:</b> The robot is <b>hanging the white object</b> on the <b>stand</b>.</p> <p><b>Pred:</b> The robot is performing a manipulation task involving a <b>white object</b>. It appears to be picking up the object and <b>moving it from one position to another on the table</b>.</p> <p><b>Pred:</b> The robot is seen picking up the <b>white object</b> and <b>moving it to another location in the video</b>.</p>
  <p><b>Question:</b> In the video, which cabinet is opened after the microwave door?</p> <p><b>Answer:</b> The <b>slide cabinet</b> is opened after the microwave door.</p> <p><b>Pred:</b> The <b>slide cabinet</b> and the <b>left hinge cabinet</b> is open after the microwave door.</p> <p><b>Pred:</b> The <b>cupboard</b> is open after the microwave door.</p> <p><b>Pred:</b> The <b>cabinet</b> that is opened after the microwave door is the one on the <b>right side</b> of the oven.</p>	  <p><b>Question:</b> What action is the robot performing in the video related to the file cabinet?</p> <p><b>Answer:</b> The robot is opening the <b>bottom drawer</b> of the <b>file cabinet</b>.</p> <p><b>Pred:</b> The robot is opening the <b>bottom drawer of a cabinet</b>.</p> <p><b>Pred:</b> The robot is opening a <b>drawer</b> of the <b>file cabinet</b>.</p> <p><b>Pred:</b> The robot is seen opening the <b>file cabinet</b> in the video.</p>
  <p><b>Question:</b> Segment the video into distinct robotic operations, providing a concise description and time range for each.</p> <p><b>Answer:</b> 1. <math>t=0.0</math> to <math>t=0.58</math>: <b>fold cloth from down to top</b> 2. <math>t=0.58</math> to <math>t=1.0</math>: <b>fold cloth from left to right</b></p> <p><b>Pred:</b> 1. <math>t=0.0</math> to <math>t=0.41</math>: <b>fold the cloth from bottom left to top right</b> 2. <math>t=0.41</math> to <math>t=1.0</math>: <b>fold the cloth from bottom left to top right</b></p> <p><b>Pred:</b> Here are the steps the robot likely follows to fold the cloth: 1. Initial Positioning: The cloth is laid flat on a surface. 2. First Fold: The robot grasps one side of the cloth and folds it halfway over itself. 3. Second Fold: It then folds the opposite side over the first fold, aligning the edges. 4. Final Fold: The robot folds the cloth in half again, creating a compact rectangle shape. This sequence results in a neatly folded cloth.</p> <p><b>Pred:</b> <del><math>t=0.000</math> to <math>t=0.125</math>: Fold the towel in half.</del> <del><math>t=0.125</math> to <math>t=0.250</math>: Fold the towel in half again.</del> <del><math>t=0.250</math> to <math>t=0.375</math>: Fold the towel in half again.</del> <del><math>t=0.375</math> to <math>t=0.500</math>: Fold the towel in half again.</del> <del><math>t=0.500</math> to <math>t=0.625</math>: Fold the towel in half again.</del> <del><math>t=0.625</math> to <math>t=0.75</math>.</del></p>	  <p><b>Question:</b> What is the first action the robot performs in the video?</p> <p><b>Answer:</b> Folds the cloth from down to top.</p> <p><b>Pred:</b> The human folds the cloth <b>from bottom left to top right</b>.</p> <p><b>Pred:</b> <del>The human first picks up the cloth.</del></p> <p><b>Pred:</b> The human folds the cloth in half.</p>
  <p><b>Question:</b> What is the first action the robot performs in the video?</p> <p><b>Answer:</b> Folds the cloth from down to top.</p> <p><b>Pred:</b> The human folds the cloth <b>from bottom left to top right</b>.</p> <p><b>Pred:</b> <del>The human first picks up the cloth.</del></p> <p><b>Pred:</b> The human folds the cloth in half.</p>	  <p><b>Question:</b> What object is the robot manipulating in the video?</p> <p><b>Answer:</b> <b>a cloth</b>.</p> <p><b>Pred:</b> The human is manipulating a <b>cloth</b>.</p> <p><b>Pred:</b> The human is manipulating a <b>towel</b> in the video.</p> <p><b>Pred:</b> The human is manipulating a <b>blue towel</b> with a white design on it.</p>

Figure 16. **Model Visualization analysis.** We present a detailed comparison of RoboAnnotatorX against state-of-the-art models, focusing on their abilities to understand and interpret robotic demonstrations. The comparison encompasses two critical aspects: scene understanding (e.g., object recognition and spatial relationships) and temporal reasoning (e.g., action sequences and temporal dependencies). Through this analysis, we demonstrate how each model processes visual inputs and reasons about temporal relationships, highlighting the unique strengths and capabilities of comprehending complex robotic tasks.

## J. Prompt Design for QA Generation and Evaluation

Our method leverages GPT-4o’s capabilities to automatically generate comprehensive question-answer pairs from demonstration meta-information and to evaluate model predictions against ground truth. This section details our prompt engineering approach and provides examples of the meta-information structure that forms the foundation of our dataset.

### J.1. Meta-Information Examples

The dual-source meta-information structure provides rich contextual data for QA generation, as shown in Fig. 17:

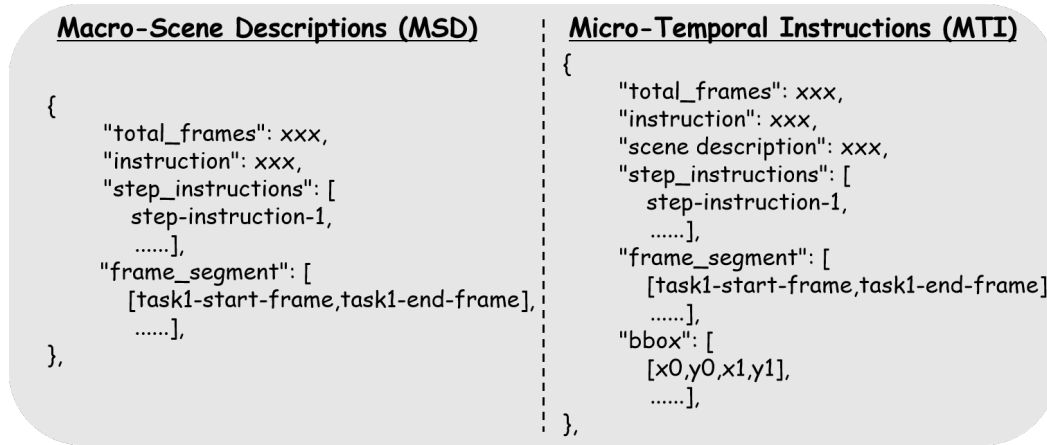


Figure 17. **Example of Meta-Information.** **Left:** Macro-Scene Descriptions (MSD) provide coarse-grained natural language annotations for real-world robotic demonstrations, capturing high-level action sequences and semantic context. **Right:** Micro-Temporal Instructions (MTI) offer fine-grained, time-stamped annotations from simulated environments, precisely documenting scene description, spatial relationships, and temporal sequences of manipulation subtasks.

### J.2. QA Generation Prompts

We designed specialized prompts to guide GPT-4o in generating diverse and semantically meaningful question-answer pairs based on the meta-information extracted from demonstrations, as illustrated in Fig. 11. Our prompt engineering follows a structured approach with several key components:

- **Meta-Information Integration:** Each prompt begins by providing the available meta-information (either MSD or MTI) as context, ensuring GPT-4o has access to the complete temporal and spatial details of the demonstration.
- **Task Type Specification:** The prompt explicitly defines the type of understanding to be probed (e.g., scene understanding, temporal understanding) and further specifies subtypes (e.g., object identification, spatial relationships, action understanding, temporal localization).
- **Question Format Guidelines:** We provide clear instructions on question formatting, including complexity level parameters and requirements for unambiguous answers that can be objectively evaluated.
- **Answer Constraint Specifications:** To maintain evaluation consistency, we define answer format requirements tailored to each question type (e.g., object names for identification questions, specific temporal markers for localization questions).

### J.3. Evaluation Prompts

For evaluating model performance, we developed systematic evaluation prompts, as shown in Fig. 18 that leverage GPT-4o’s text-processing capabilities and transform GPT-4o into a consistent evaluator and grader across our diverse question categories:

- **Text-Only Evaluation Protocol:** Our approach critically relies on a pure text-based assessment methodology. The evaluation compares only textual model predictions against textual ground-truth answers, without requiring GPT-4o to process or understand any visual information from the original demonstrations. This text-only approach leverages GPT-4o’s superior language understanding while entirely bypassing its limitations in video comprehension.
- **Scoring Framework:** We implemented a 0-5 scoring system with precisely defined rubrics for each level of scores.

This text-based evaluation approach is particularly advantageous for our task, as it isolates linguistic comprehension from visual interpretation, allowing for focused assessment of how well models convert visual demonstrations into accurate textual

You are an intelligent chatbot designed for evaluating the correctness of generative outputs for question-answer pairs.

Your task is to compare the predicted answer with the correct answer and determine if they match meaningfully. Here's how you can accomplish the task:

##INSTRUCTIONS:

- Focus on the meaningful match between the predicted answer and the correct answer.
- Consider synonyms or paraphrases as valid matches.
- Evaluate the correctness of the prediction compared to the answer.

Please evaluate the following video-based question-answer pair:

Question: question

Correct Answer: answer

Predicted Answer: pred

Provide your evaluation only as a yes/no and score where the score is an integer value between 0 and 5, with 5 indicating the highest meaningful match. Consider the following guidelines:

1. Minor differences in object descriptions should not necessarily result in a negative evaluation if the overall action is correct.
2. Consider the potential for slight misinterpretations in visual details, especially for similar objects or surfaces.

Please generate the response in the form of a Python dictionary string with keys 'pred' and 'score', where value of 'pred' is a string of 'yes' or 'no' and value of 'score' is in INTEGER, not STRING.

DO NOT PROVIDE ANY OTHER OUTPUT TEXT OR EXPLANATION. Only provide the Python dictionary string.

Figure 18. **Prompt template for text-based evaluation.** Our evaluation framework relies on GPT-4o's sophisticated text processing capabilities to compare model outputs against ground truth answers. The template includes precise instructions for semantic matching, evaluation criteria, and standardized scoring, enabling consistent assessment without requiring any visual information. This text-only approach leverages GPT-4o's language understanding strengths while avoiding its limitations in video comprehension.

descriptions. While GPT-4o's visual understanding capabilities may be limited, its exceptional language processing abilities make it an ideal evaluator for comparing textual descriptions derived from robotic demonstrations, ensuring both reliability and validity in our experimental results.

## K. Failure Cases Analysis

Through detailed analysis of our algorithm’s failure cases, as shown in Fig. 19, we observe that the discrepancies between model predictions and ground truth annotations can be categorized into two main types: alternative answers and genuine errors. Here we present a comprehensive analysis of these cases and their underlying causes.

### K.1. Alternative Answers

Our analysis reveals that certain discrepancies, while technically classified as failures by strict evaluation metrics, actually represent valid alternative interpretations. These cases can be categorized into three main types:

- **Alternative References:** We observe variations in how the model refers to scene elements:
  - Object References: Different but valid terms for the same object (e.g., "soda can" vs. "beverage container", "orange bottle" vs. "juice container")
  - Spatial References: When describing positions among multiple objects, different anchoring choices can yield varying but equally valid descriptions, such as describing an object X between A and B as either "right of A" or "left of B".
- **Perspective Variations:** The model may generate descriptions that differ from GT annotations in their reference frame or viewpoint while maintaining semantic accuracy. For instance, when GT states "The robot is positioned above the white bowl," the model might output "The robot is positioned adjacent to the white bowl." Such variations reflect different but equally valid spatial interpretations of the same scene configuration.
- **Granularity Differences:** The model sometimes provides more detailed descriptions than GT. E.g., GT: "grape toy" vs. Model: "purple plush toy". Such cases demonstrate the model’s capability to capture and articulate fine-grained object attributes while preserving the fundamental semantic meaning.

alternative object description

What object is being instructed to be placed upright in the video?

**Output:** An orange can.

**GT:** A water bottle.

alternative object description

What object is being placed into another object in the video?

**Output:** The purple plush toy is being placed into the beige bowl.

**GT:** The grape toy is being placed into the beige toy.

alternative spatial relationship

What is the spatial relationship between the robot and the white bowl from which it is picking the apple?

**Output:** The robot is positioned above the white bowl while performing the action of picking the apple.

**GT:** The robot is positioned adjacent to the white bowl.

positional error

What action is the robot performing in the video?

**Output:** The robot is opening the bottom drawer.

**GT:** The robot is opening the middle drawer.

object error

What action is the robot performing in the video?

**Output:** The robot is picking a coke can from the top shelf of the fridge.

**GT:** The robot is picking a blue plastic bottle from the top shelf of the fridge.

action error

Identify and briefly describe the key robotic actions in this video, specifying the start and end times for each action.\n

**Output:** 1. t=0.0 to t=0.52: Move the yellow cloth to the left of the pot. 2. t=0.52 to t=1.0: Move the can to the left of the fork.

**GT:** 1. t=0.0 to t=0.48: Place the can in front of the blue fork and against the wall. 2. t=0.48 to t=1.0: Place the yellow towel to the left of the blue fork

Figure 19. **Failure case examples.** Visualization of discrepancies between model predictions and ground truth annotations, highlighting alternative answers and genuine errors.



## K.2. Genuine Errors

In contrast to alternative answers, we identify several types of genuine recognition failures that impact model performance, as can be seen from Figs. 20 and 21:

- **Visual Similarity Confusion:** Misidentification due to similar appearance attributes:
  - Shape and Color similarity (e.g., white cup vs. white bowl)
  - Material confusion (e.g., plastic vs. glass containers)
- **Multi-object Interference:** Confusion in scenes with multiple objects
- **Context-induced Confusion:** Context-induced Confusion (e.g., assuming a container near coffee is a coffee cup)
- **Absolute Spatial Limitation:** Difficulty in accurately estimating absolute distances and object sizes, mainly because two-dimensional video frames lack depth perception cues.
- **Weak Intention Prediction:** Challenges in accurately capturing and interpreting the sequential nature of actions and events in videos and inferring the underlying intention behind observed actions, primarily due to the model’s limited capability in causal reasoning.
- **Training Data-Induced Limitations:** Certain errors stem from biases and gaps in the training data.
  - **Biased Toward Success:** The model shows stronger performance on successful tasks, while its ability to detect failures is limited, likely due to fewer failure cases in the training data.
  - **Viewpoint Variation Weakness:** The model struggles with understanding navigation videos involving continuous viewpoint changes, likely because its training data predominantly consists of fixed-view manipulation videos and its architectural design emphasizes action recognition rather than handling large viewpoint shifts.

Our analysis reveals several crucial implications for improving video understanding systems, particularly in the context of long-horizon robotic demonstrations. Given our focus on generating high-quality annotations for complex robotic tasks, we identify three key areas for advancement: First, evaluation metrics need fundamental revision to better accommodate valid alternative descriptions while maintaining discriminative power for genuine errors - this is especially crucial for robotic demonstrations where actions and spatial relationships can have multiple valid interpretations. Second, model improvements

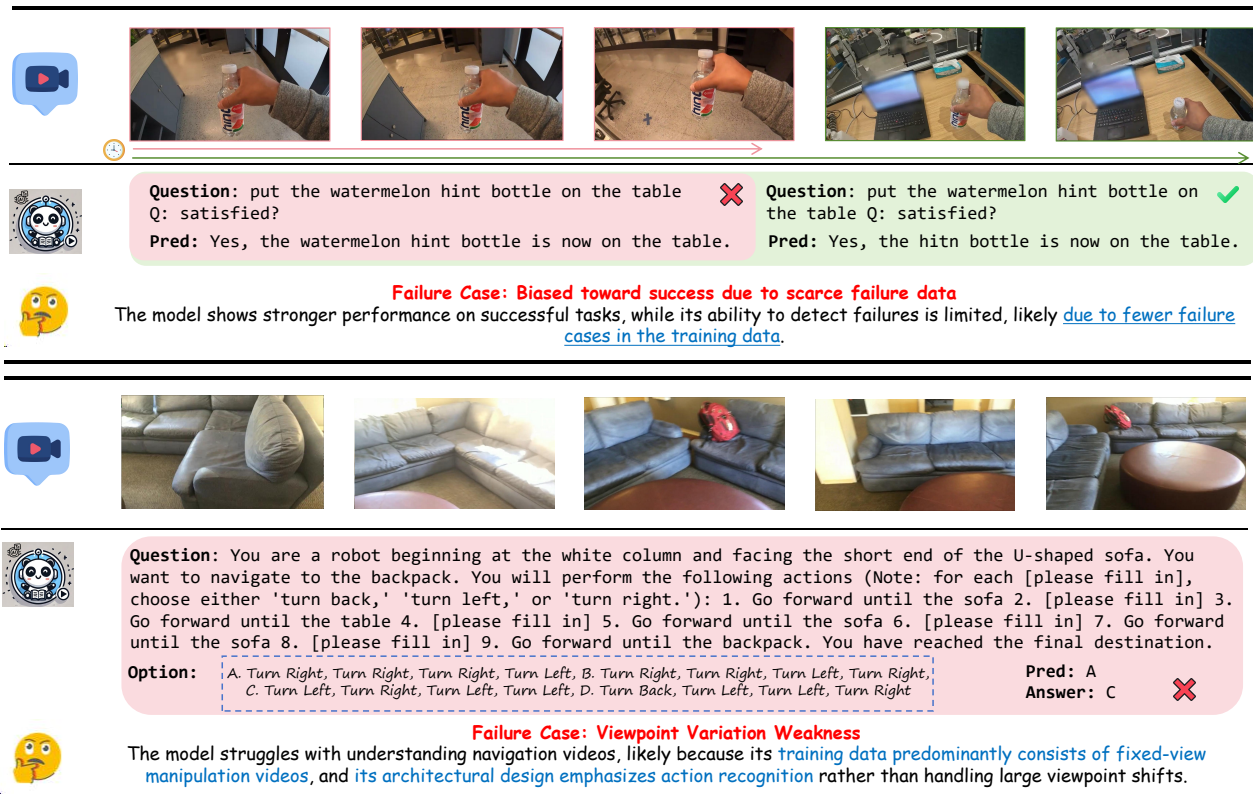


Figure 20. **Failure examples of training data-induced errors.** Visualizations demonstrate cases where the model fails due to (i) bias toward successful task recognition and (ii) difficulties in handling navigation videos with continuous viewpoint changes.

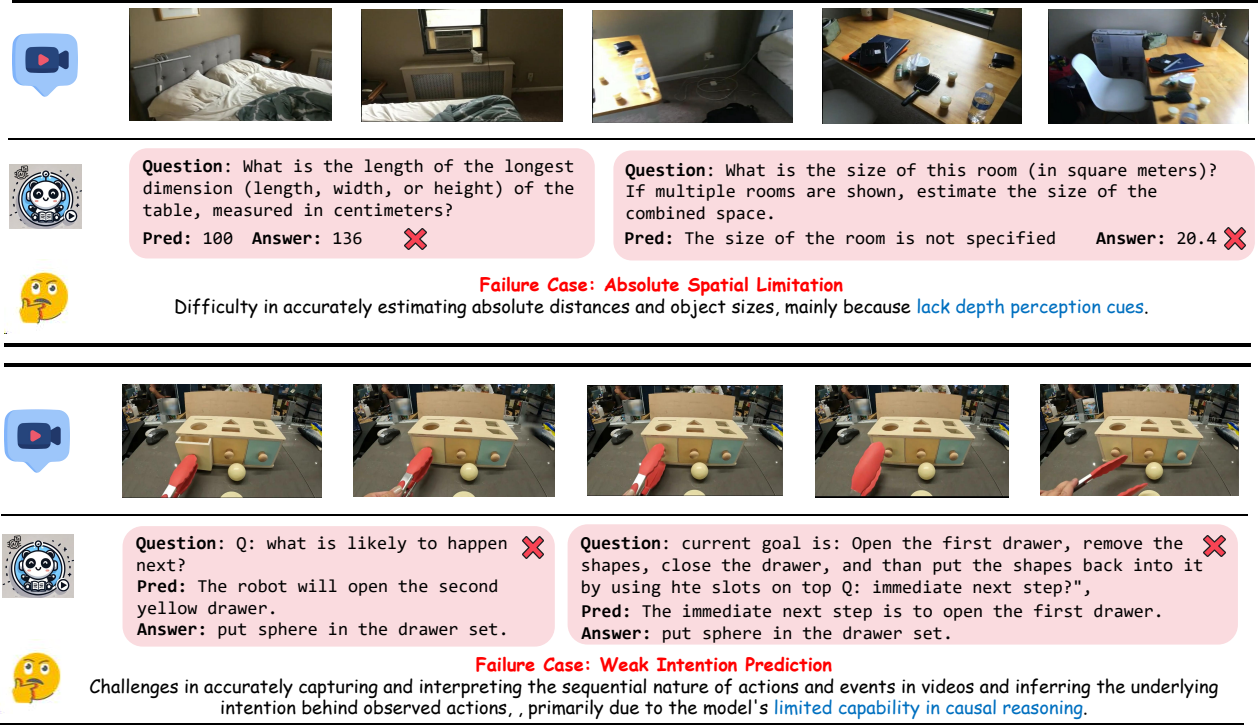


Figure 21. **Failure Examples.** Visualizations highlight the model’s limitations in Absolute Spatial Estimation, where 2D video frames lack depth cues, and Weak Intention Prediction, stemming from limited causal reasoning for sequential actions.

should focus on enhanced feature discrimination for similar objects, improved spatial reasoning, and better temporal context integration, directly addressing the challenges of maintaining semantic coherence across extended demonstrations. Third, the incorporation of 3D information appears critical for resolving many spatial ambiguities and object relationships in complex scenes, particularly for robotic manipulation tasks where precise spatial understanding is essential. These insights align with our multi-scale approach and curriculum training strategy, potentially enabling more robust and generalizable automated annotation systems for complex robotic demonstrations.

## L. Limitation & Future Work

While RoboAnnotatorX advances automated annotation of robotic demonstrations through its multi-scale token-efficient encoder and curriculum-based training, several fundamental challenges and opportunities for improvement remain:

- **3D Understanding:** While our current framework effectively processes 2D visual information, it exhibits limitations in robust 3D spatial reasoning, particularly for complex manipulation scenarios requiring precise object relationships. We plan to address this by integrating depth information and exploring neural implicit representations to better capture 3D scene dynamics and spatial relationships in robotic interactions. Potential directions include leveraging multi-view geometry and 3D scene graphs to enhance the framework’s spatial understanding capabilities.
- **Multi-modal Annotation Generation:** Although our framework excels at natural language descriptions, it currently lacks the capability to generate diverse structured annotations (e.g., bounding boxes, motion trajectories, keypoints) that are crucial for robotic learning. Future work will focus on extending the model architecture with specialized decoders for different annotation modalities while maintaining computational efficiency.
- **Real-time Processing:** The current framework has limited capability for real-time annotation generation, constraining its applications in active learning and dynamic environments. We are exploring model compression techniques and efficient inference strategies to reduce latency while preserving annotation quality. Integration with streaming architectures could enable online processing of continuous demonstrations.

Moving forward, our research will prioritize several key directions: developing more sophisticated evaluation metrics that better accommodate semantic variations, integrating multi-sensor spatial information for enhanced 3D understanding, and optimizing the model architecture for improved scalability. Through these improvements and our established foundation in multi-scale temporal modeling and curriculum training, we aim to advance RoboAnnotatorX as a reliable annotation tool

that can help unlock the full potential of robotic demonstrations, while RoboX-VQA continues to serve as a comprehensive benchmark for developing and evaluating long-horizon robot video understanding capabilities.

## References

- [1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023. [3](#)
- [2] Nils Blank, Moritz Reuss, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Wenzel, Oier Mees, and Rudolf Lioutikov. Scaling robot policy learning via zero-shot labeling with foundation models. In *8th Annual Conference on Robot Learning*, 2024. [4](#)
- [3] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14455–14465, 2024. [5](#)
- [4] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. [3](#)
- [5] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. [1](#)
- [6] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A robotic dataset for learning diverse skills in one-shot. In *RSS 2023 Workshop on Learning for Task and Motion Planning*, 2023. [6](#)
- [7] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023. [1](#)
- [8] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. Maniskill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations*, 2023. [9](#)
- [9] Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*, 2019. [7](#)
- [10] Pu Hua, Minghuan Liu, Annabella Macaluso, Yunfeng Lin, Weinan Zhang, Huazhe Xu, and Lirui Wang. Gensim2: Scaling robot data generation with multi-modal and reasoning llms, 2024. [7](#), [9](#)
- [11] Siyuan Huang, Iaroslav Ponomarenko, Zhengkai Jiang, Xiaoqi Li, Xiaobin Hu, Peng Gao, Hongsheng Li, and Hao Dong. Manipvqa: Injecting robotic affordance and physically grounded information into multi-modal large language models. *arXiv preprint arXiv:2403.11289*, 2024. [5](#)
- [12] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. [16](#)
- [13] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. 2024. [3](#)
- [14] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. [3](#)
- [15] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Videogpt+: Integrating image and video encoders for enhanced video understanding. *arxiv*, 2024. [3](#)
- [16] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):7327–7334, 2022. [7](#)
- [17] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems*, 2024. [7](#), [9](#)
- [18] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024. [16](#)
- [19] Abby O’Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. [6](#)
- [20] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021. [3](#)
- [21] Kanchana Ranasinghe, Xiang Li, Kumara Kahatapitiya, and Michael Ryoo. Understanding long videos in one multimodal language model pass, 2024. [4](#)

- [22] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. *ArXiv*, abs/2312.02051, 2023. 1, 3
- [23] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2024. 9
- [24] 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. 3
- [25] Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation. *arXiv preprint arXiv:2311.01455*, 2023. 7, 9
- [26] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Chenting Wang, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, et al. Internvideo2: Scaling video foundation models for multimodal video understanding. *arXiv preprint arXiv:2403.15377*, 2024. 3
- [27] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava : Parameter-free llava extension from images to videos for video dense captioning, 2024. 3
- [28] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024. 3
- [29] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 3
- [30] Shengqiang Zhang, Philipp Wicke, Lütfi Kerem Şenel, Luis Figueredo, Abdeldjalil Naceri, Sami Haddadin, Barbara Plank, and Hinrich Schütze. Lohoravens: A long-horizon language-conditioned benchmark for robotic tabletop manipulation. *arXiv preprint arXiv:2310.12020*, 2023. 7, 8
- [31] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, 2024. 3
- [32] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. 1
- [33] Yan Zheng, Xiaofei Xie, Ting Su, Lei Ma, Jianye Hao, Zhaopeng Meng, Yang Liu, Ruimin Shen, Yingfeng Chen, and Changjie Fan. Wuji: Automatic online combat game testing using evolutionary deep reinforcement learning. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 772–784, 2019. 6
- [34] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020. 9