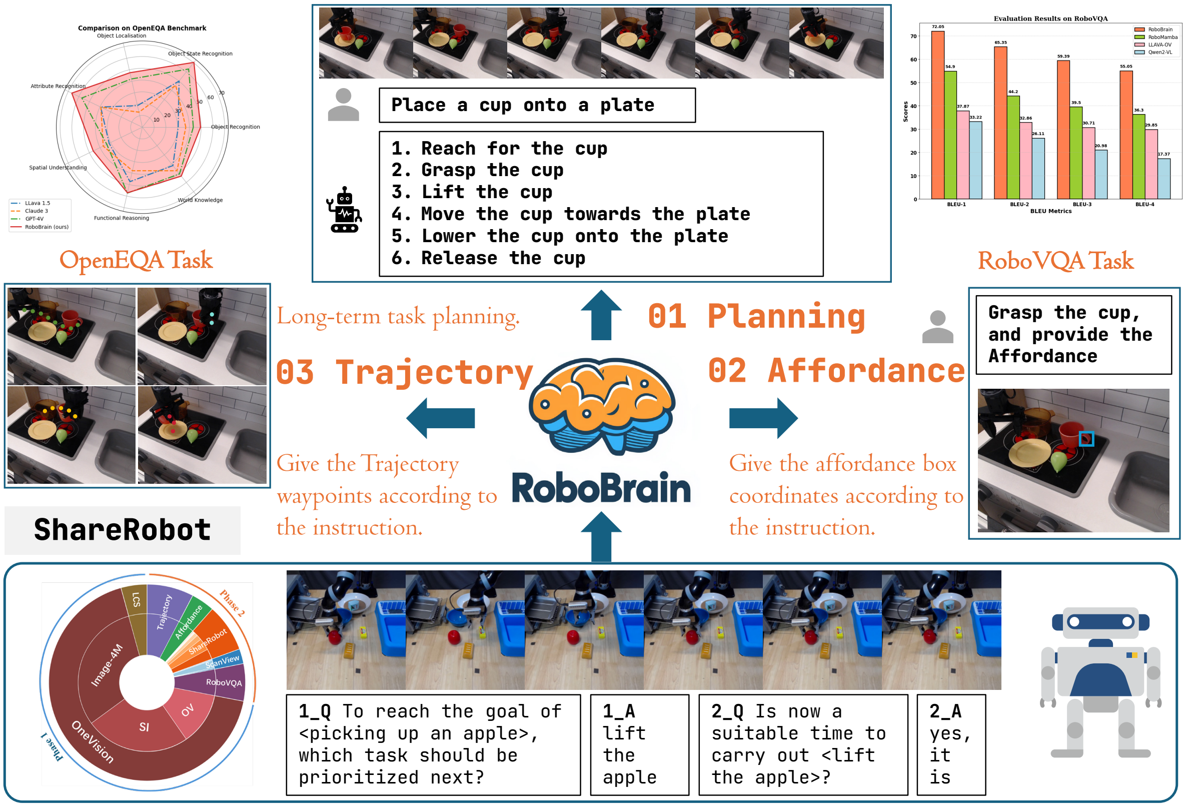
RoboBrain: A Unified Brain Model for Robotic Manipulation from Abstract to Concrete

Yuheng Ji[2,3,6,*], Huajie Tan[1,2,*], Jiayu Shi[1,2,*], Xiaoshuai Hao[2,*,†], Yuan Zhang[1,2], Hengyuan Zhang[1,2]
Pengwei Wang[2,†], Mengdi Zhao[2], Yao Mu[5], Pengju An[1,2], Xinda Xue[1,2], Qinghang Su[2,4], Huaihai Lyu[2,3,6]
Xiaolong Zheng[3,6], Jiaming Liu[1,2], Zhongyuan Wang[2], Shanghang Zhang[1,2,✉]
[1] State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University
[2] Beijing Academy of Artificial Intelligence [3] Institute of Automation, Chinese Academy of Sciences
[4] Institute of Information Engineering, Chinese Academy of Sciences [5] The University of Hong Kong
[6] School of Artificial Intelligence, University of Chinese Academy of Sciences



Place a cup onto a plate

1. Reach for the cup
2. Grasp the cup
3. Lift the cup
4. Move the cup towards the plate
5. Lower the cup onto the plate
6. Release the cup

Long-term task planning.

Grasp the cup, and provide the Affordance

Give the affordance box coordinates according to the instruction.

Give the Trajectory waypoints according to the instruction.

1_Q To reach the goal of <picking up an apple>, which task should be prioritized next?

1_A lift the apple

2_Q Is now a suitable time to carry out <lift the apple>?

2_A yes, it is

Figure 1. **Overview of RoboBrain.** RoboBrain consists of three key robotic capabilities: planning capability, affordance perception, and trajectory prediction. RoboBrain outperforms previous MLLMs in robotics tasks. The bottom part shows the composition of RoboBrain's training data and provides a specific example of visual question answering from our proposed ShareRobot. Best viewed on screen.

Abstract

Recent advancements in Multimodal Large Language Models (MLLMs) have shown remarkable capabilities across various multimodal contexts. However, their application in robotic scenarios, particularly for long-horizon manipulation tasks, reveals significant limitations. These limitations arise from the current MLLMs lacking three essential robotic brain capabilities: ***Planning Capability****, which involves decomposing complex manipulation instructions into manageable sub-tasks;* ***Affordance Per-***

[*] Equal contribution.
[†] Project leaders.
[✉] Corresponding author.

ception*, the ability to recognize and interpret the affordances of interactive objects; and* ***Trajectory Prediction****, the foresight to anticipate the complete manipulation trajectory necessary for successful execution. To enhance the robotic brain's core capabilities from abstract to concrete, we introduce* ***ShareRobot****, a high-quality heterogeneous dataset that labels multi-dimensional information such as task planning, object affordance, and end-effector trajectory. ShareRobot's diversity and accuracy have been meticulously refined by three human annotators. Building on this dataset, we developed* ***RoboBrain****, an MLLM-based model that combines robotic and general multi-modal data, utilizes a multi-stage training strategy, and incorporates long videos and high-resolution images to improve its robotic manipulation capabilities. Extensive experiments demonstrate that RoboBrain achieves state-of-the-art performance across various robotic tasks, highlighting its potential to advance robotic brain capabilities. Project website: RoboBrain.*

1. Introduction

Recent advancements in Multimodal Large Language Models (MLLMs) have significantly advanced the pursuit of Artificial General Intelligence (AGI). By leveraging extensive multimodal datasets sourced from the internet and employing self-supervised learning techniques, MLLMs demonstrate exceptional capabilities in visual perception and understanding human language instructions, excelling in tasks such as visual question answering [3, 14, 15], image captioning [35, 37], and sentiment analysis [17]. Despite significant progress in MLLMs, the exploration of their application in robotics remains in its early stages, highlighting a crucial area for further research and innovation.

Recent studies have examined the application of MLLMs in robotics, focusing on planning and subgoal decomposition [6, 25], action sequencing [8, 9], and replanning and feedback [41, 46]. However, their effectiveness in robotic scenarios—particularly for long-horizon manipulation tasks—reveals significant limitations. These limitations stem from the current MLLMs' lack of three critical robotic capabilities: planning, affordance perception, and trajectory prediction, as illustrated in Fig. 1. For instance, consider a robotic arm tasked with lifting a teapot and pouring water into a cup. The MLLM should be capable of decomposing this task into sub-tasks, such as "approach the teapot and lift it", "move the teapot until the spout is positioned over the cup", and "tilt the teapot to pour". For each sub-task, such as "approach and grasp the teapot", the MLLM must utilize affordance perception to accurately identify the graspable regions of the teapot. Additionally, trajectory prediction is essential for determining the complete path from the starting point to the graspable part of the teapot. This challenge for existing MLLMs primarily arises from the scarcity of large-scale, fine-grained datasets specifically designed for robotic operation tasks.

To empower the RoboBrain's core capabilities from abstract to concrete, we first introduce ***ShareRobot***, a large-scale, fine-grained dataset specifically designed for robotic operation tasks. Specifically, we label multi-dimensional information such as task planning, object affordance, and end-effector trajectory. Building upon ShareRobot, we developed ***RoboBrain***, an MLLM model based on the LLaVA [40] architecture, aimed at enhancing the perception and planning capabilities of robots in complex tasks. In the process of training RoboBrain, we meticulously designed the ratio of robotic data to general multi-modal data, implemented a multi-stage training strategy, and incorporated long videos and high-resolution images. This approach endowed RoboBrain with powerful visual information perception capabilities in robotic scenarios, supporting historical frame memory and high-definition image input, thereby further enhancing the ability in robotic manipulation planning. Extensive experimental results demonstrate that RoboBrain outperforms existing models across multiple robotic benchmarks, including RoboVQA [60] and OpenEQA [49], achieving state-of-the-art performance. Additionally, it shows competitive results in trajectory and affordance prediction accuracy. These findings validate the effectiveness of the proposed dataset and framework in enhancing robotic brain capabilities. In summary, the main contributions of this paper are as follows:

- We propose ***RoboBrain***, a unified multimodal large language model designed for robotic manipulation, which facilitates more efficient task execution by transforming abstract concepts into concrete actions.
- We meticulously designed the ratio of robotic data to general multi-modal data, implemented a multi-stage training strategy, and incorporated long videos and high-resolution images. This approach provided ***RoboBrain*** with historical frame memory and high-resolution image input, thereby further enhancing its capabilities in robotic manipulation planning.
- We introduce ***ShareRobot***, a high-quality heterogeneous dataset that labels multi-dimensional information, including task planning, object affordance, and end-effector trajectory, effectively enhancing various robotic capabilities.
- Comprehensive experimental results demonstrate that ***RoboBrain*** achieves state-of-the-art performance across various embodied benchmarks, highlighting its potential for real-world applications in robotics.

2. Related Work

MLLM for Robotic Manipulation Planning Existing studies mostly utilize MLLMs primarily focus on understanding natural language and visual observation tasks [6–

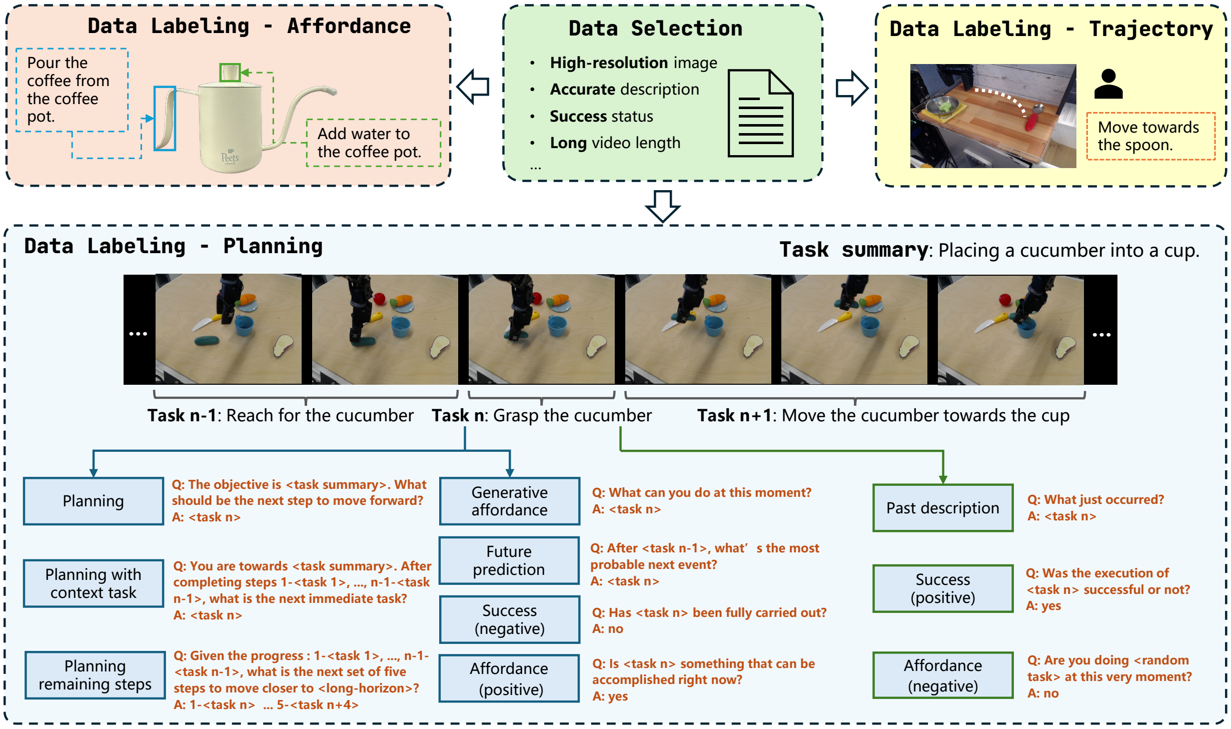

Data Labeling - Affordance
Pour the coffee from the coffee pot.
Add water to the coffee pot.
Data Selection
High-resolution image
Accurate description
Success status
Long video length
...
Data Labeling - Trajectory
Move towards the spoon.
Data Labeling - Planning
Task summary: Placing a cucumber into a cup.
Task n-1: Reach for the cucumber
Task n: Grasp the cucumber
Task n+1: Move the cucumber towards the cup
Planning
Q: The objective is <task summary>. What should be the next step to move forward?
A: <task n>
Planning with context task
Q: You are towards <task summary>. After completing steps 1-<task 1>, ..., n-1-<task n-1>, what is the next immediate task?
A: <task n>
Planning remaining steps
Q: Given the progress : 1-<task 1>, ..., n-1-<task n-1>, what is the next set of five steps to move closer to <long-horizon>?
A: 1-<task n> ... 5-<task n+4>
Generative affordance
Q: What can you do at this moment?
A: <task n>
Future prediction
Q: After <task n-1>, what' s the most probable next event?
A: <task n>
Success (negative)
Q: Has <task n> been fully carried out?
A: no
Affordance (positive)
Q: Is <task n> something that can be accomplished right now?
A: yes
Past description
Q: What just occurred?
A: <task n>
Success (positive)
Q: Was the execution of <task n> successful or not?
A: yes
Affordance (negative)
Q: Are you doing <random task> at this very moment?
A: no

Figure 2. **The generation procession of our ShareRobot dataset.** Our dataset labels multi-dimensional information, including task planning, object affordance, and end-effector trajectories. The task planning is first annotated by atomic tasks and then augmented by constructing question-answer pairs. The affordance and trajectory are labeled on the images according to the specific instructions.

8, 30], with fewer addressing the decomposition of high-level task instructions into actionable steps. PaLM-E [19] generates multimodal inputs by mapping real-world observations into the language embedding space. RT-H [6] and Robomamba [42] generate reasoning results along with robot actions obtained from an additional policy head. However, while these models generate planning texts and actions, they still lack adequate mechanisms for executing complex atomic tasks, highlighting the need for enhanced affordance perception and trajectory prediction.

Datasets for Manipulation Planning Early datasets for Manipulation [11, 22, 31, 44, 62] mainly comprise annotated images and videos that highlight fundamental hand-object interactions, including grasping and pushing. Recent advancements [18, 60] in robotic manipulation emphasize multi-modal and cross-embodiment datasets for enhanced generalization. Datasets such as RH20T [20], BridgeDataV2 [68], and DROID [28] enhance scene diversity, broadening the range of manipulation scenarios. Notably, RT-X [54] compiles data from 60 datasets across 22 embodiments into the Open X-Embodiment (OXE) repository. In this work, we extract high-quality data from OXE, decompose high-level descriptions into low-level planning instructions, and adapt these into a question-answer format to enhance model training.

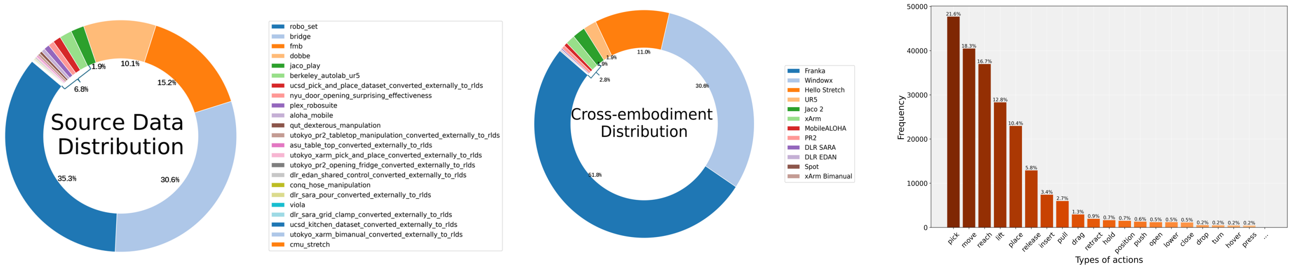
3. ShareRobot Dataset

To enhance the RoboBrain's capability of planning, affordance perception, and trajectory prediction, we develop a dataset called ShareRobot–a large-scale, fine-grained dataset specifically designed for robotic operation tasks. The generation procession of our dataset is shown as Fig. 2 The details are described in the following sections.

3.1. Overview

ShareRobot is a comprehensive dataset, facilitates more efficient task execution by transforming abstract concepts into concrete actions. The main features of the ***ShareRobot*** dataset include:

- **Fine-grained** Unlike the Open X-Embodiment dataset[53], which only offers generalized high-level task descriptions, each data point in ShareRobot includes detailed low-level planning instructions linked to individual frames. This specificity enhances the model's ability to execute tasks accurately at the right moment.
- **Multi-dimensional** To enhance RoboBrain's capabilities from abstract to concrete, we label task planning, object affordances, and end-effector trajectories, allowing for greater flexibility and precision in task processing.
- **High quality** We establish rigorous criteria for selecting



(a) Source Data Distribution **(b) Cross-embodiment Distribution** **(c) Statics of types of atomic tasks**

Figure 3. **The diversity of our ShareRobot dataset.** Our dataset involves (a) 23 original datasets, (b) 12 embodiments and (c) 107 types of atomic tasks. The distribution of the top 20 most frequent atomic actions within our ShareRobot dataset is presented in (c).

data from the Open-X-Embodiment dataset[53], focusing on high resolution, accurate descriptions, successful task execution, visible affordance, and clear motion trajectories. Based on these criteria, we validate 51,403 instances to ensure high quality, forming the foundation for RoboBrain's core capabilities.

- **Large scale** With 1,028,060 question-answer pairs, ShareRobot is the largest open-source dataset for task planning, affordance prediction, and trajectory prediction, enabling deeper understanding of complex relationships from abstract to concrete.
- **Rich diversity** In contrast to the RoboVQA[60] dataset's limited scenes, ShareRobot features 102 scenes across 12 embodiments and 107 types of atomic tasks, as shown in Fig. 3. This diversity allows MLLMs to learn from varied real-world contexts, enhancing robustness in complex, multi-step planning.
- **Easy scalability** Our data generation pipeline is designed for high scalability, facilitating expansion as new robotic embodiments, task types, and environments develop. This adaptability ensures the ShareRobot dataset can support increasingly complex manipulation tasks.

3.2. Data Selection

Based on the Open X-embodiment dataset [53], we carefully selected 51,403 instances, mainly focusing on image quality, description accuracy and success status. Our data collection process adheres to the following principles:

- **High-resolution image** We eliminate videos lacking images or those with low resolution. Any video with a resolution below 128 pixels is removed.
- **Accurate description** Videos without descriptions or with vague descriptions are filtered out to avoid affecting the planning capability of the model.
- **Success status** We discard videos conducting failed tasks, as unsuccessful demonstrations affect the model's learning.
- **Long video length** Videos with fewer than 30 frames are excluded, as they contain only atomic tasks.
- **Object not covered** We remove any videos where the target object or end-effector is covered by other objects, as our model has to accurately identify the positions of end-effectors and the object's affordance.
- **Clear Trajectories** We exclude the demonstrations with unclear or incomplete trajectories, as trajectory recognition is one of our RoboBrain's capabilities.

3.3. Data Labeling

Planning Labeling We extract 30 frames from each robotic operation demonstration. We use these frames along with their high-level descriptions to decompose them into low-level planning instructions using Gemini [63]. Three annotators then review and refine these instructions to ensure the precision of labeling. The low-level planning data is formatted to align with the RoboVQA [60] structure for model training, employing question templates for the 10 question types in RoboVQA. This process transforms 51,403 low-level planning entries into 1,028,060 question-answer pairs, with annotators monitoring data generation to maintain the dataset's integrity.

Affordance Labeling We filter 8,511 images from the dataset and annotate each with affordance areas. For each 30-frame demonstration, we label the affordance in the first frame, corresponding to the contact regions between the end-effectors and the objects. We identify the contact frame, where the end-effectors first contact the object, and label the ground truth bounding box in the first frame as $\{l^{(x)}, l^{(y)}, r^{(x)}, r^{(y)}\}$, where $\{l^{(x)}, l^{(y)}\}$ are the top left coordinates and $\{r^{(x)}, r^{(y)}\}$ are the bottom right corner coordinates.

Trajectory Labeling We annotate 8,511 images with bounding boxes for the gripper, maintaining consistency with the affordance bounding box format. Each end-effector is labeled with three parts: the entire gripper, the left finger, and the right finger. This data serves to calculate trajectory positions and train a gripper detector. The trajectory position is determined by averaging the bounding boxes of the left and right fingers, allowing for efficient labeling of additional data.

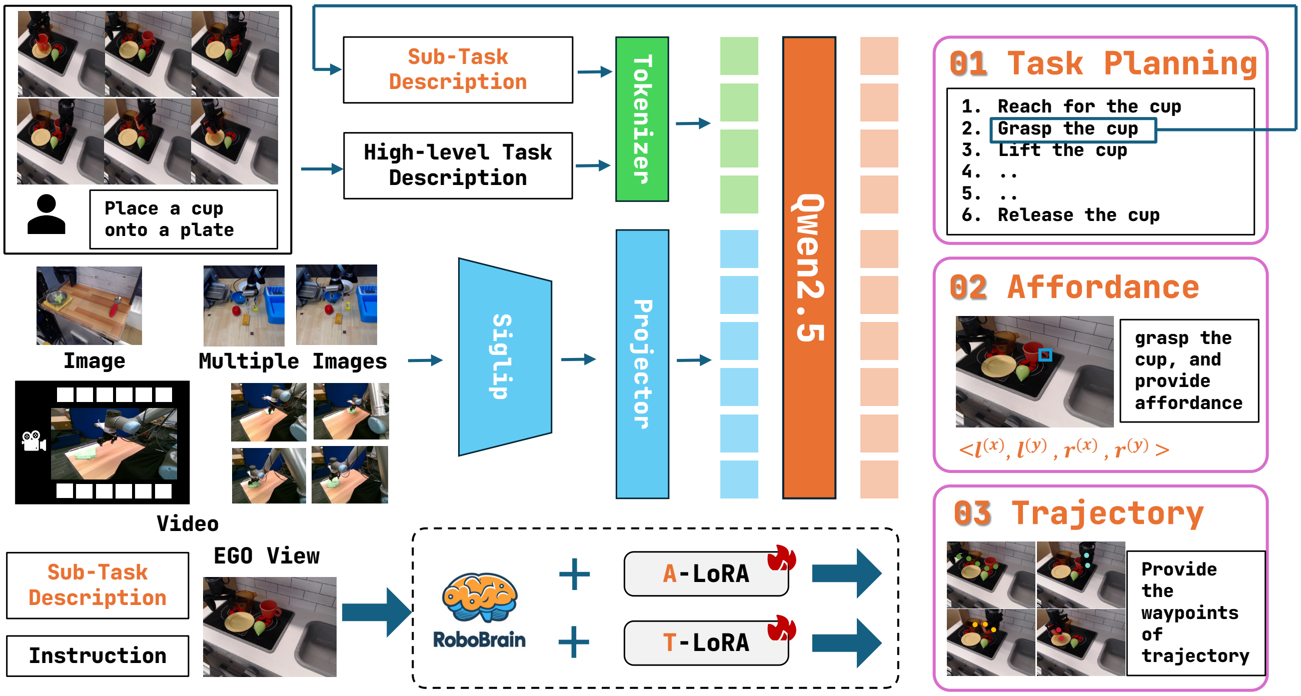


Figure 4. **The pipeline of our RoboBrain.** The images, multiple images, and videos are sent into our model to pre-train a foundation robotic brain. Besides, we fine-tune the RoboBrain via A-LoRA and T-LoRA to develop Affordance and Trajectory skills. In practical applications, the model first generates detailed plans, and then splits it into sub-task descriptions to execute specific robotic tasks.

3.4. Data Statistics

We select 23 original datasets from the Open X-embodiment dataset[53]. The distribution of the source data is shown in the Fig. 3. The data involves 102 various scenes (e.g. bedroom, laboratory, kitchen, office), and covers 12 different robot bodies. According to statistics, there are 132 types of atomic actions in this dataset, tasks with higher word frequency are shown in Fig. 3 (c). The 5 most frequent atomic tasks are “pick”, “move”, “reach”, “lift”, and “place”, which are frequent task types in real robotic operation scenarios. This suggests that the distribution of our dataset is reasonable. Finally, we get 1,028,060 question-answer pairs for planning. For the planning QA pairs dataset, we split 1 million QA pairs as the training set and 2,050 QA pairs as the test set. For the affordance dataset, we split 8000 images as the training set and 511 images as the test set. For the trajectory dataset, we allocate 8000 images for training and 511 images for testing.

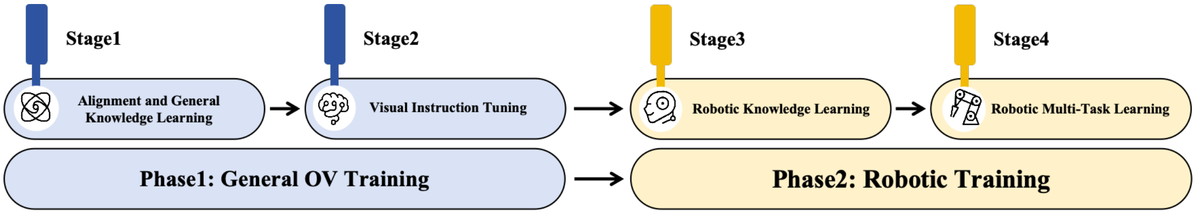
4. RoboBrain Model

In this section, we provide an overview of ***RoboBrain***. Our goal is to enable the Multi-modal Large Language Model (MLLM) to understand abstract instructions and explicitly output object affordance regions and potential operational trajectories, facilitating a transition from abstract to concrete. We employ a multi-stage training strategy: Phase 1 focuses on general OneVision (OV) training to develop a foundational MLLM with strong understanding and instruction-following abilities. Phase 2, the robotic training phase, aims to empower the core capabilities of RoboBrain from abstract to concrete.

4.1. Model Architecture

RoboBrain consists of three modules: the foundational model for planning, the A-LoRA model for affordance perception, and the T-LoRA model for trajectory prediction. In practical applications, the model first generates detailed plans, and then splits it into sub-task descriptions to execute affordance perception and trajectory prediction. The pipeline of our RoboBrain is shown to Fig. 4.

Foundational Model for Planning We utilize LLaVA as the foundational Model for RoboBrain, which consists of three main modules: the Vision Encoder (ViT) $g(\cdot)$, the Projector $h(\cdot)$, and the Large Language Model (LLM) $f(\cdot)$. Specifically, we employ SigLIP [74], a 2-layer MLP [39], and Qwen2.5-7B-Instruct [64]. Given an image or video X_v as visual input, ViT encodes it into visual features $Z_v = g(X_v)$, which are then mapped to the semantic space of the LLM through Projector, resulting in a sequence of visual tokens $H_v = h(Z_v)$. Finally, the LLM generates a textual response in an autoregressive manner based on the human language instruction X_t and H_v.


Stage1
Alignment and General Knowledge Learning
Stage2
Visual Instruction Tuning
Stage3
Robotic Knowledge Learning
Stage4
Robotic Multi-Task Learning
Phase1: General OV Training
Phase2: Robotic Training

		Stage-1	Stage-1.5	Stage-2		Stage-3	Stage-4	
				Single-Image	OneVision		A-LoRA	T-LoRA
Vision	**Resolution**	384	Max 384×{2×2}	Max 384×{6×6}	Max 384×{6×6}	Max 384×{6×6}	Max 384×{6×6}	Max 384×{6×6}
	#Tokens	729	Max 729×5	Max 729×37	Max 729×37	Max 729×37	Max 729×37	Max 729×37
Data	**Dataset**	LCS	Image	Image	Image & Video	Robotic Data	Afford. Data	Traj. Data
	#Samples	558K	4M	3.2M	1.6M	3M	10K	400K
Model	**Trainable**	Projector	Full Model	Full Model	Full Model	Full Model	A-LoRA	T-LoRA
	#Tunable Parameters	17.0M	8.0B	8.0B	8.0B	8.0B	28.0M	28.0M
Training	**Batch Size**	8	2	1	1	1	4	4
	LR: ψ_{ViT}	-	2×10^{-6}	2×10^{-6}	2×10^{-6}	2×10^{-6}	2×10^{-6}	2×10^{-6}
	LR: $\{\theta_{\text{Proj.}}, \phi_{\text{LLM}}, \phi_{\text{LoRA}}\}$	1×10^{-3}	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-5}
	Epoch	1	1	1	1	1	1	1

Table 1. Detailed configuration for each training stage of the RoboBrain.

A-LoRA Module for Affordance Perception The term affordance in our work refers to the area where the human hand makes contact with objects. During interactions, humans instinctively engage with various objects within specific regions. We utilize *bounding boxes* to represent affordances. Formally, consider an image I consisting of multiple objects with their affordances: $O_i = \{A_i^0, A_i^1, ..., A_i^N\}$, where the ith object owns N affordances. The format of affordance is defined as $\{l^{(x)}, l^{(y)}, r^{(x)}, r^{(y)}\}$, and $\{l^{(x)}, l^{(y)}\}$ represents the top left corner coordinates of affordance, while $\{r^{(x)}, r^{(y)}\}$ is the bottom right corner coordinates.

T-LoRA Module for Trajectory Prediction The term trajectory in our work refers to the concept of *2D visual traces*, as presented in [21]. We define trajectory waypoints as a series of 2D coordinates representing the movement of the end-effector or hand throughout the process. Formally, at time step t, the trajectory waypoints can be represented as $P_{t:N} = \{(x_i, y_i) \mid i = t, t+1, \ldots, N\}$, where (x_i, y_i) denotes the i-th coordinate in the visual trace, and N represents the total number of time steps in the episode.

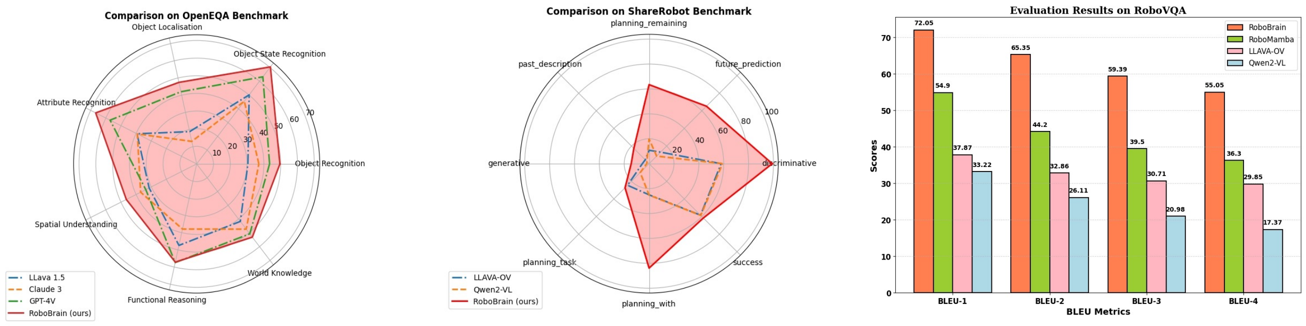
4.2. Training

Phase 1: General OV Training In Phase 1, we drew on the state-of-the-art training data and strategies from LLaVA-OneVision [34] to construct a foundational model with general multi-modal understanding and visual instruction following capabilities. This lays the groundwork for enhancing the model's robotic manipulation planning abilities in Phase 2. Detailed information is provided in Tab. 1.

In Stage 1, we utilize the image-text data from the LCS-558K dataset [10, 59] to train Projector, facilitating the alignment of visual features Z_v with the LLM semantic features H_v. In Stage 1.5, we train the entire model using 4M high-quality image-text data to enhance the model's multi-modal general knowledge understanding capabilities. In Stage 2, we further train the entire model with 3.2M single-image data and 1.6M image and video data from LLaVA-OneVision-Data [34], aiming to enhance the instruction-following abilities of RoboBrain and improve understanding of high-resolution image and video.

Phase 2: Robotic Training In Phase 2, we build upon the robust multi-modal foundational model developed in Phase 1 to create a more powerful model for robotic manipulation planning. Specifically, we aim for RoboBrain to understand complex, abstract instructions, support the perception of historical frame information and high-resolution images, and output object affordance regions while predicting potential manipulation trajectories. This will facilitate the transition from abstract to concrete in manipulation planning tasks. Detailed information is provided in Tab. 1.

In stage 3, we collected a dataset of 1.3M robotic data to improve the model's robotic manipulation planning capabilities. Specifically, this data is sourced from RoboVQA-800K [60], *ScanView-318K* including MMScan-224K [24, 47], 3RScan-43K[24, 67], ScanQA-25K [4, 24], SQA3d-26K [24, 48], and the subset of ShareRobot-200K introduced in this paper. These datasets contain a substantial amount of scene-scanning image data, long video data, and high-resolution data to support the model's ability to perceive diverse environments. Additionally, the fine-grained, high-quality planning data within the ShareRobot dataset further enhances the robotic manipulation planning capabil-



(a) OpenEQA Benchmark

(b) ShareRobot Benchmark

(c) RoboVQA Benchmark

Figure 5. The performance of our model RoboBrain on the OpenEQA, ShareRobot, and RoboVQA benchmarks. RoboBrain surpassed all baseline models, achieving state-of-the-art results.

ities of RoboBrain. To mitigate the issue of catastrophic forgetting [75], we selected a subset of high-quality image-text data from Phase 1 about 1.7M to mix with the robotic data collected in Stage 3 for training, tuning the entire model accordingly. In Stage 4, we further enhanced the model's ability to perceive object affordances based on instructions and predict manipulation trajectories, utilizing affordance and trajectory data annotated in the ShareRobot dataset. This was achieved by introducing LoRA [23] modules for training to realize fine-grained planning capabilities. For the specific construction of training data and the training methods employed, please refer to the supplementary materials.

5. Experiment

5.1. Implementation Details

During the entire training phase, we employed the Zero3 [58] distributed training strategy, conducting all experiments on a cluster of servers, each equipped with 8×A800 GPUs. The training components for each stage, including image resolution settings, batch size, epochs, and learning rates, are provided in Tab. 1.

5.2. Evaluation Metrics

Planning Task We selected RoboVQA [60], OpenEQA [49], and the test set of ShareRobot extracted from the proposed ShareRobot dataset as robotic benchmarks for multi-dimensional assessment. For RoboVQA, we adopt the BLEU1 to BLEU4 metrics [56] used in RoboMamba [42] for evaluation. Additionally, for OpenEQA and ShareRobot, we use GPT-4o [55] as the evaluation tool, scoring based on the alignment or similarity between model predictions and ground truth, which serves as the final performance score for the model.

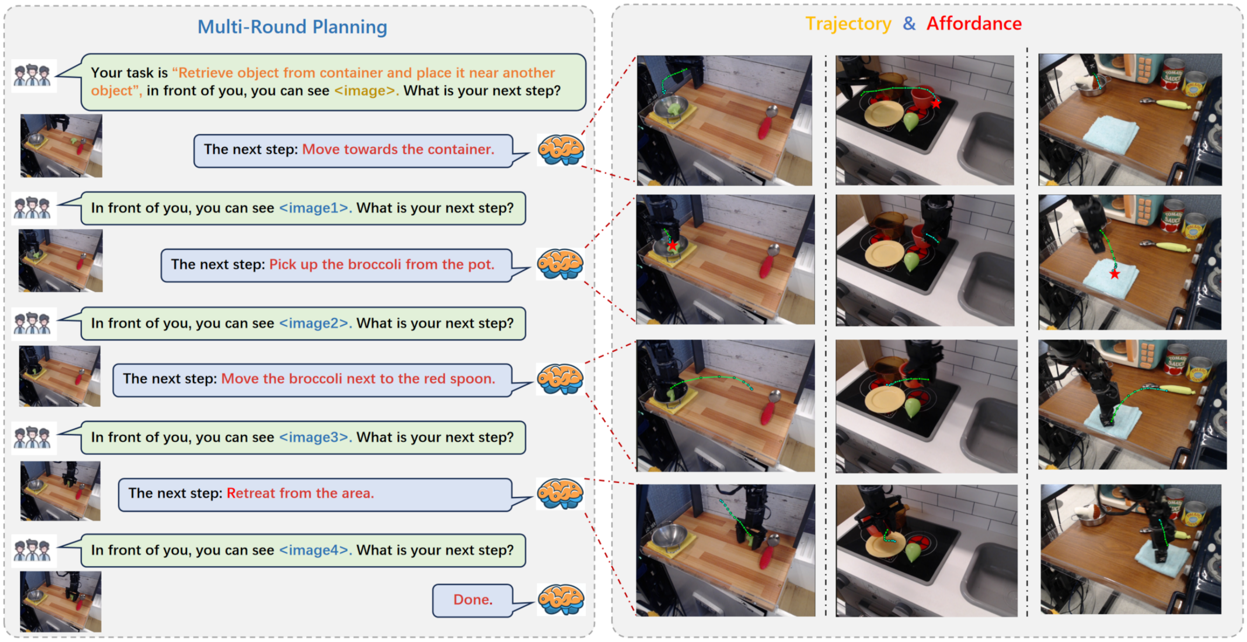
Trajectory Prediction We evaluate the similarity between ground truth and predicted trajectories, both represented as sequences of 2D waypoints normalized to $[0, 1000)$, following Qwen2-VL [70]. The evaluation uses three metrics: Discrete Fréchet Distance (DFD) [21], Hausdorff Distance (HD), and Root Mean Square Error (RMSE). DFD captures overall shape and temporal alignment, HD identifies maximum deviation, and RMSE measures average pointwise error. Together, these metrics provide a comprehensive assessment of trajectory accuracy and similarity.

Affordance Prediction Here, we utilize the average precision (AP) to evaluate the affordance performance of our model. AP metric summarizes the precision-recall affordance curve, which plots the relationship between precision and recall at various threshold settings. It is calculated across multiple IoU (Intersection over Union) thresholds to obtain a more comprehensive evaluation.

5.3. Evaluation on Robot Brain Task

Evaluation on Planning Task We selected 6 powerful MLLMs as our baselines for comparison, including both open-source and closed-source models with different architectures. Specifically, these models include GPT-4V [2], Claude3 [1], llava1.5 [40], LLaVA-OneVision-7b [34], Qwen2-VL-7b [69] and RoboMamba [42]. Our specific experimental results are shown in Fig. 5. Our RoboBrain outperformed all baseline models across three robotic benchmarks. RoboBrain significantly outperformed all baseline models on OpenEQA and ShareRobot, which can be attributed to its robust capabilities in understanding robotic tasks and perceiving long videos. Additionally, this pattern was observed in other benchmarks as well, with RoboBrain consistently demonstrating superior performance on RoboVQA, achieving a BLEU-4 score that exceeded that of the second-place model by 18.75. This result highlights its capability to decompose complex long-range task planning. Please refer to the supplementary materials for more ablation studies due to space limitation.

Evaluation on Trajectory Prediction We compare sev-



Your task is "Retrieve object from container and place it near another object", in front of you, you can see <image>. What is your next step?

The next step: Move towards the container.

In front of you, you can see <image1>. What is your next step?

The next step: Pick up the broccoli from the pot.

In front of you, you can see <image2>. What is your next step?

The next step: Move the broccoli next to the red spoon.

In front of you, you can see <image3>. What is your next step?

The next step: Retreat from the area.

In front of you, you can see <image4>. What is your next step?

Done.

Figure 6. This visualization illustrates that RoboBrain can interpret human instructions and visual images to generate action plans and assessments based on real-time image feedback. Furthermore, it predicts trajectories for each step and identifies corresponding affordances.

Table 2. **Trajectory Prediction Results Comparison**. Discrete Fréchet Distance (DFD), Hausdorff Distance (HD), and Root Mean Square Error (RMSE).

Method	DFD ↓	HD ↓	RMSE ↓
RoboBrain (Base)	0.191	0.171	0.133
+ Start_Points	0.176	0.157	0.117
+ Max_Points	0.185	0.163	0.125
+ Spec_Token	**0.109** (42.9%↓)	**0.010** (94.2%↓)	**0.091** (31.6%↓)

eral variants of our model, and the results are in Tab. 2: **(1) Baseline**, fine-tuned on trajectory-related VQA data; **(2) Start_Points**, which adds the 2D start coordinates of the end-effector; **(3) Max_Points**, limiting waypoints to 10 via uniform sampling; and **(4) Spec_Token & End_Points**, which adds end-effector positions and special tokens to emphasize waypoints and start/goal points. Each variant builds on the previous one, with the final model integrating all components. Our most effective model integrates all design choices. As shown in the last row of Tab. 2, DFD, HD, and RMSE decreased by 42.9%, 94.2%, and 31.6%, respectively, compared to the baseline. We found that adding start points corrected the translational offset between the generated trajectory and the end-effector.

Evaluation on Affordance Prediction Our results are summarized in Tab. 3. We compare the Qwen2-VL-7B and LLaVA-NeXT-7B models. Qwen2-VL [69] has a superior visual grounding ability and LLaVA-NeXT [36] owns a high-resolution and strong vision tower. We test them all

Table 3. **The comparison of affordance prediction.** We utilize AP as the metric, and test them on affordance test set.

Model	**AP** ↑
LLaVA-NeXT-7B [40]	9.8 %
Qwen2-VL-7B [5]	12.5 %
RoboBrain (Ours)	**27.1** % (14.6↑)

on the AGD20K affordance test set. Our RoboBrain outperforms significantly the other models. It surpasses Qwen2-VL [69] by 14.6 AP, and LLaVA-NeXT by 17.3 AP. It validates our RoboBrain can understand the physical properties of objects and provide the affordance accurately.

5.4. Visualization

In this section, we present visual examples of RoboBrain, as shown in Fig 6. Given human language instructions and visual images, RoboBrain can engage in multi-turn interactions with humans, understanding and predicting future steps. Additionally, it outputs more concrete trajectories and affordances.

6. Conclusion

In this paper, we introduce ***ShareRobot***, a high-quality dataset that labels multi-dimensional information, including task planning, object affordance, and end-effector trajectory. We also present ***RoboBrain***, an MLLM-based model that integrates robotic and general multi-modal data, employs a multi-stage training strategy, and leverages long videos and high-resolution images to enhance robotic manipulation. Extensive experiments demonstrate that RoboBrain achieves state-of-the-art performance across various robotic tasks, underscoring its potential to significantly advance robotic capabilities.

Acknowledgments This work was supported by the National Natural Science Foundation of China (62476011, 72225011 and 72434005).

References

[1] The claude 3 model family: Opus, sonnet, haiku. 7

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 7, 13, 14, 15

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, pages 2425–2433, 2015. 2

[4] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *CVPR*, pages 19129–19139, 2022. 6, 13

[5] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 8, 12, 16

[6] Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*, 2024. 2, 3

[7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[8] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 2, 3

[9] Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *CoRL*, pages 287–318, 2023. 2

[10] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, pages 3558–3568, 2021. 6, 12

[11] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S. Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, Jan Kautz, and Dieter Fox. Dexycb: A benchmark for capturing hand grasping of objects. In *CVPR*, pages 9044–9053, 2021. 3

[12] Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang, Zhihong Chen, Jianquan Li, Xiang Wan, and Benyou Wang. Allava: Harnessing gpt4v-synthesized data for a lite vision-language model. *arXiv preprint arXiv:2402.11684*, 2024. 12

[13] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024. 14

[14] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. In *ICLR*. 2, 12

[15] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, pages 24185–24198, 2024. 2, 13, 14

[16] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 16

[17] Ringki Das and Thoudam Doren Singh. Multimodal sentiment analysis: a survey of methods, trends, and challenges. *ACM Computing Surveys*, 55(13s):1–38, 2023. 2

[18] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. In *CoRL*, pages 885–897, 2019. 3

[19] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. 3

[20] Haoshu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. RH20T: A comprehensive robotic dataset for learning diverse skills in one-shot. In *ICRA*, pages 653–660, 2024. 3

[21] Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, et al. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. *arXiv preprint arXiv:2311.01977*, 2023. 6, 7

[22] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, pages 3193–3203, 2020. 3

[23] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *ICLR*. 7, 12

[24] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. In *ICLR 2024 Workshop: How Far Are We From AGI*. 6

[25] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor

Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In *CoRL*, pages 1769–1782. PMLR, 2023. 2

[26] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. 16

[27] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *ECCV*, pages 235–251. Springer, 2016. 12, 14

[28] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. 3

[29] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *ECCV*, 2022. 12

[30] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 3, 16

[31] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2O: two hands manipulating objects for first person interaction recognition. In *ICCV*, pages 10118–10128, 2021. 3

[32] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023. 14

[33] Bo Li, Hao Zhang, Kaichen Zhang, Dong Guo, Yuanhan Zhang, Renrui Zhang, Feng Li, Ziwei Liu, and Chunyuan Li. Llava-next: What else influences visual instruction tuning beyond data, 2024. 12, 16

[34] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 6, 7, 12, 13, 16

[35] Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, He Chen, Guohai Xu, Zheng Cao, et al. mplug: Effective and efficient vision-language learning by cross-modal skip-connections. In *EMNLP*, pages 7241–7259, 2022. 2

[36] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024. 8, 12, 13, 14, 15

[37] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, pages 19730–19742. PMLR, 2023. 2

[38] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. 16

[39] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, pages 26296–26306, 2024. 5

[40] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 36, 2024. 2, 7, 8, 12

[41] Jiaming Liu, Chenxuan Li, Guanqun Wang, Lily Lee, Kaichen Zhou, Sixiang Chen, Chuyan Xiong, Jiaxin Ge, Renrui Zhang, and Shanghang Zhang. Self-corrected multimodal large language model for end-to-end robot manipulation. *arXiv preprint arXiv:2405.17418*, 2024. 2

[42] Jiaming Liu, Mengzhen Liu, Zhenyu Wang, Lily Lee, Kaichen Zhou, Pengju An, Senqiao Yang, Renrui Zhang, Yandong Guo, and Shanghang Zhang. Robomamba: Multimodal state space model for efficient robot reasoning and manipulation. *arXiv preprint arXiv:2406.04339*, 2024. 3, 7, 14, 15

[43] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024. 14

[44] Yun Liu, Haolin Yang, Xu Si, Ling Liu, Zipeng Li, Yuxiang Zhang, Yebin Liu, and Li Yi. TACO: benchmarking generalizable bimanual tool-action-object understanding. In *CVPR*, pages 21740–21751, 2024. 3

[45] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *ECCV*, pages 216–233. Springer, 2025. 14

[46] Zeyi Liu, Arpit Bahety, and Shuran Song. Reflect: Summarizing robot experiences for failure explanation and correction. In *CoRL*, pages 3468–3484. PMLR, 2023. 2

[47] Ruiyuan Lyu, Tai Wang, Jingli Lin, Shuai Yang, Xiaohan Mao, Yilun Chen, Runsen Xu, Haifeng Huang, Chenming Zhu, Dahua Lin, and Jiangmiao Pang. Mmscan: A multimodal 3d scene dataset with hierarchical grounded language annotations. *arXiv preprint arXiv:2406.09401*, 2024. 6, 13

[48] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. In *ICLR*, 2023. 6, 13

[49] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, et al. Openeqa: Embodied question answering in the era of foundation models. In *CVPR*, pages 16488–16498, 2024. 2, 7, 14, 15, 16

[50] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *CVPR*, pages 3195–3204, 2019. 12

[51] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022. 14

[52] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021. 14

[53] 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. 3, 4, 5

[54] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, et al. Open x-embodiment: Robotic learning datasets and RT-X models : Open x-embodiment collaboration. In *ICRA*, pages 6892–6903, 2024. 3

[55] OpenAI. Hello gpt-4o, 2024. 7, 13, 14, 15

[56] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318, 2002. 7

[57] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 12

[58] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *SIGKDD*, pages 3505–3506, 2020. 7

[59] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *NeuIPS*, 35:25278–25294, 2022. 6, 12

[60] Pierre Sermanet, Tianli Ding, Jeffrey Zhao, Fei Xia, Debidatta Dwibedi, Keerthana Gopalakrishnan, Christine Chan, Gabriel Dulac-Arnold, Sharath Maddineni, Nikhil J Joshi, et al. Robovqa: Multimodal long-horizon reasoning for robotics. In *ICRA*, pages 645–652, 2024. 2, 3, 4, 6, 7, 13, 14, 15, 16

[61] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *CVPR*, pages 8317–8326, 2019. 14

[62] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *ECCV*, pages 581–600, 2020. 3

[63] Gemini Team, Rohan Anil, Sebastian Borgeaud, et al. Gemini: A family of highly capable multimodal models, 2024. 4

[64] Qwen Team. Qwen2.5: A party of foundation models, 2024. 5, 12

[65] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024. 12

[66] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 16

[67] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. Rio: 3d object instance re-localization in changing indoor environments. In *ICCV*, pages 7658–7667, 2019. 6, 13

[68] Homer Rich Walke, Kevin Black, Tony Z. Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, Abraham Lee, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata V2: A dataset for robot learning at scale. In *CoRL*, pages 1723–1736, 2023. 3

[69] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 7, 8, 13, 14, 15, 16

[70] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 7

[71] x.ai. Grok 1.5v vision preview. `https://x.ai/blog/grok-1.5v`, 2024. Accessed: 2024-11-21. 14

[72] Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, et al. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. *arXiv preprint arXiv:2310.05126*, 2023. 12

[73] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *CVPR*, pages 9556–9567, 2024. 14

[74] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, pages 11975–11986, 2023. 5, 12

[75] Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. Investigating the catastrophic forgetting in multimodal large language models. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*. 7

[76] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models, 2024. *arXiv preprint arXiv:2407.12772*. 13

[77] Yuan Zhang, Fei Xiao, Tao Huang, Chun-Kai Fan, Hongyuan Dong, Jiawen Li, Jiacong Wang, Kuan Cheng, Shanghang Zhang, and Haoyuan Guo. Unveiling the tapestry of consistency in large vision-language models. *arXiv preprint arXiv:2405.14156*, 2024. 13

Appendix

This supplementary material provides more details of the proposed method and experiment results that are omitted from the manuscript due to the page limit. Sec. A presents additional details of the models and training strategies. Sec. B presents details of training dataset. Sec. C complements more experiment results and analysis. Sec. D shows more visualization results to prove the effectiveness of RoboBrain. Sec. E introduces more details about the construction of ShareRobot dataset.

A. Details of Models and Training

Model Setting. RoboBrain is built upon the LLaVA [40] framework and consists of three main components: the visual encoder, projector, and large language model (LLM).

For the visual encoder, we utilized the SigLIP [74] model, specifically the siglip-so400m-patch14-384, which is pre-trained on WebLi [14] at a resolution of 384x384. The SigLIP model improves upon traditional CLIP [57] architectures by employing a sigmoid loss function that operates solely on image-text pairs, eliminating the need for global normalization of pairwise similarities. This enhancement allows for more efficient scaling of batch sizes while maintaining performance, even at smaller scales. SigLIP has 27 hidden layers and processes input images using patches of size 14x14, resulting in 729 visual tokens per image. The projector consists of a two-layer MLP [36] that projects the visual tokens obtained from the visual encoder to the dimensions of the text embeddings. For the LLM, we adopted the Qwen2.5-7B-Instruct [64] model, which is a state-of-the-art open-source LLM that is part of the latest Qwen series [5]. It features 28 hidden layers and supports long-context inputs of up to 128K tokens, providing multilingual capabilities across more than 29 languages.

In Stage 4, we introduced LoRA [23] to train RoboBrain, enabling it to acquire affordance perception and trajectory prediction capabilities. LoRA is a technique that allows for parameter-efficient fine-tuning of large models by adding low-rank parameter matrices to existing layers. We incorporated LoRA modules with a rank of 64 into the feed-forward network layers of both the Projector and the LLM, freezing all parameters except those of the LoRA modules during training.

Training Setting. In the main text of the paper, we employed a staged training strategy, with complete settings presented in Tab. 4. We primarily referenced the training strategy of LLaVA-Onevision [34], a state-of-the-art multimodal large language model, and built upon this foundation to expand the robotic training phase. During the entire training phase, we conducted all experiments on a cluster of servers, each equipped with 8×A800 GPUs.

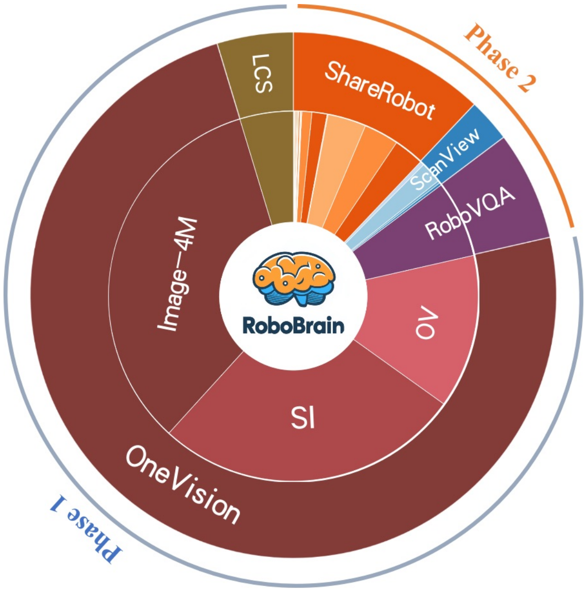


Figure 7. The distribution of the entire training dataset.

B. Details of Training Dataset

In the main body of the paper, we emphasize the importance of the training data and the proportion of robotic data. In this section, we will provide a detailed overview of the training data and its sources. The distribution of the entire training dataset is illustrated in Fig. 7.

- **LCS-558K** is a subset of the LAION/CC/SBU dataset [10, 59], specifically designed as the LLaVA Visual Instruct Pretrain [40] Dataset. This dataset has been carefully filtered to achieve a more balanced distribution of concept coverage, ensuring diverse and representative visual content. The primary purpose of LCS-558K is to facilitate the alignment between the visual encoder and the LLM, enabling the LLM to comprehend visual information.
- **Image-4M** comprises 8 data sources, including 3 from the LLaVA-Recap series [33]: BLIP558K, COCO118K, and CC3M, as well as UReader [72], Instruct Azure DC [33], Evol-Instruct [12], and SynthDog [29] We utilized the download links provided by the LLaVA-OneVision team for the data acquisition.
- **SI-3.2M** [34] consists of 3.2 million samples, carefully curated to support multimodal learning. It includes subsets from existing datasets such as Cambrian [65], Cauldron [33], and UReader [72], which were subjected to cleaning and re-annotation to ensure data quality. Additionally, it incorporates single-image data from sources like AI2D [27] and OKVQA [50], alongside a newly compiled single-image collection designed to achieve a

Due to the unavailability of certain datasets, the actual data used amounts to 3.1M.

Table 4. Detailed configuration for each training stage of the RoboBrain.

		Stage-1	Stage-1.5	Stage-2		Stage-3	Stage-4	
				Single-Image	**OneVision**		**A-LoRA**	**T-LoRA**
Vision	**Resolution**	384	Max 384×{2×2}	Max 384×{6×6}	Max 384×{6×6}	Max 384×{6×6}	Max 384×{6×6}	Max 384×{6×6}
	#Tokens	729	Max 729×5	Max 729×37	Max 729×37	Max 729×37	Max 729×37	Max 729×37
Model	**Trainable**	Projector	Full Model	Full Model	Full Model	Full Model	A-LoRA	T-LoRA
	#Tunable Parameters	17.0M	8.0B	8.0B	8.0B	8.0B	28.0M	28.0M
Training	**Per-device Batch Size**	8	2	1	1	1	4	4
	Gradient Accumulation	1	2	2	2	2	2	2
	LR: $\psi_{\textbf{ViT}}$	-	2×10^{-6}	2×10^{-6}	2×10^{-6}	2×10^{-6}	2×10^{-6}	2×10^{-6}
	LR: $\{\theta_{\textbf{Proj.}}, \phi_{\textbf{LLM}}, \phi_{\textbf{LoRA}}\}$	1×10^{-3}	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-5}
	Epoch	1	1	1	1	1	1	1
	Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
	Deepspeed	Zero3	Zero3	Zero3	Zero3	Zero3	Zero2	Zero2
	Weight Decay	0	0	0	0	0	0	0
	Warmup Ratio	0.03	0.03	0.03	0.03	0.03	0.03	0.03
	LR Schedule	cosine	cosine	cosine	cosine	cosine	cosine	cosine
	Projector Type	mlp2x_gelu	mlp2x_gelu	mlp2x_gelu	mlp2x_gelu	mlp2x_gelu	mlp2x_gelu	mlp2x_gelu
	Vision Select Layer	-2	-2	-2	-2	-2	-2	-2
	Patch Merge Type	spatial_unpad	spatial_unpad	spatial_unpad	spatial_unpad	spatial_unpad	spatial_unpad	spatial_unpad
	Frames Upbound	-	-	-	32	32	32	32
	Max Seq Length	8192	32768	32768	32768	32768	4096	4096
	GPU Nums	16*8	16*8	20*8	20*8	22*8	4*8	4*8

balanced and diverse dataset.

- **OV-1.6M** [34] comprises 1.6 million samples, which includes approximately 800K high-quality samples resampled from earlier SI-3.2M datasets with a data replay strategy, ensuring improved data reliability and relevance. Additionally, the dataset incorporates M4-Instruct data to enrich instructional learning tasks. A significant component of OV-1.6M is its video data, which has been released alongside LLaVA-video data. The video subset used in the dataset is specifically aligned with the previous annotation format, providing a diverse multimodal resource for advancing vision-language learning.
- **RoboVQA-800K** [60] consists of realistic data gathered from various user requests, utilizing different embodiments including robots, humans, and humans equipped with grasping tools. The dataset features 5,246 long-horizon episodes and 92,948 medium-horizon episodes of robotic tasks, with each episode accompanied by corresponding image and text prompt inputs. The primary purpose of RoboVQA-800K is to enhance RoboBrain's reasoning capabilities in robotic-related scenarios.
- **ScanView-318K** totals 318K samples, which integrates data from several high-quality sources, including MMScan-224K [47], 3RScan-43K [67], ScanQA-25K [4], and SQA3D-26K [48], each contributing unique strengths. MMScan-224K provides multimodal scene data with detailed annotations, such as object segmentation and textual descriptions. 3RScan-43K offers 3D reconstructions and semantic annotation. ScanQA-25K includes question-answer pairs based on 3D scanned environments. SQA3D-26K focuses on spatial question answering. Together, these datasets provide diverse scene-scanning image data, long video sequences, and high-resolution samples, equipping models with fine-grained environmental perception and reasoning abilities.

C. Complementary Experiments

In this section, we present the complete experiments and results that are omitted from the manuscript due to page limitations. This includes an exploration of the impact of incorporating ShareRobot on training, the effects of varying proportions of robotic data in the training dataset, and more comprehensive results comparing RoboBrain with the baselines on both general and robotic benchmarks.

Additionally, we explore the impact of different architectures and pre-trained VLMs, as well as different LLM backbones on our experimental results. We also conduct ablation studies at various stages to meticulously analyze the contributions of each stage to overall performance.

C.1. More Results on General Benchmarks

To evaluate performance on general tasks in real-world scenarios, as is commonly done with MLLMs [2, 15, 36, 55, 69], we conducted experiments using a diverse set of image benchmarks summarized in Table 5. We leveraged the comprehensive evaluation toolkit, LMMs-Eval[76, 77], to evaluate RoboBrain's performance on general benchmarks. These benchmarks are categorized into three classes:

- **Chart, Diagram, and Document Understanding.** As key visual formats for structured OCR data, benchmarks

Due to the vague descriptions and missing key information regarding dataset filtering in the original paper, we ended up using 2.4M data.

Table 5. Performance comparison on multiple general benchmarks.

Dataset	Split	RoboBrain (Ours)	GPT-4V [2]	LLaVA-OV-7B [36]	InternVL2-8B [15]	Qwen2-VL-7B [69]	GPT-4o [55]
A12D[27]	test	82.03	78.2	81.4	83.8	-	**94.2**
ChartQA[51]	test	80.48	78.5	80	83.3	83	**85.7**
DocVQA[52]	test	88	88.4	87.5	91.6	**94.5**	92.8
TextVQA[61]	val	75.85	-	71.07	77.4	**84.3**	-
MMMU[73]	val	49	56.8	48.8	51.8	54.1	**69.1**
MMStar[13]	test	61.23	57.1	61.7	61.5	60.7	**63.9**
OCRBench[43]	-	677	656	697	794	**845**	805
RealWorldQA[71]	test	68.89	61.4	66.3	64.4	**70.1**	58.6
SeedBench[32]	image	71.03	49.9	75.4	**76.2**	-	**76.2**
MMbench[45]	en-dev	81.52	81.3	83.2	-	-	**83.4**
MMbench[45]	en-test	80.44	75	80.8	81.7	**83**	-
MME[**?**]	test	2084	1926	1998	2210	**2327**	-

such as AI2D [27], ChartQA [51], DocVQA [52], and OCRBench [43] were utilized. Open-source models like InternVL2-8B [15] and LLAVA-OV-7B [36] have demonstrated comparable performance to closed-source models such as GPT-4V [2]. For *RoboBrain*, despite being optimized primarily for multidimensional robotic tasks, it surpasses LLAVA-OV-7B [36] and GPT-4V [2] on these benchmarks, achieving a significant improvement in structured OCR tasks, with the only exceptions being DocVQA [52], where it performs slightly lower than GPT-4V [2], and OCRBench [43], where it falls slightly behind LLAVA-OV-7B [36].

- **Visual Perception and Multi-domain Reasoning.** This category focuses on complex visual perception and multidisciplinary reasoning tasks. Benchmarks for visual perception include MMStar [13], MMBench [45], and MME [**?**], while reasoning benchmarks include MMMU [73] and SeedBench [32]. *RoboBrain* demonstrates comparable performance to GPT-4V [2] and LLAVA-OV-7B [36] across multiple benchmarks.
- **Real-world Understanding and Interaction.** Evaluating MLLMs [2, 15, 36, 55, 69] as general-purpose assistants in real-world settings is crucial, as these scenarios extend beyond controlled environments. For this, the RealworldQA [71] benchmark was utilized. Results indicate that *RoboBrain* not only outperforms open-source models like LLAVA-OV-7B [36] and InternVL2-8B [15], but also exceeds closed-source models such as GPT-4V [2] and GPT-4o [55], showcasing its extensive knowledge base and strong generalization capabilities.

C.2. More Results on Robotic Benchmarks.

To evaluate *RoboBrain*'s performance on robotic capabilities in real-world scenarios, we selected RoboVQA [60], OpenEQA [49], and the test set of ShareRobot, extracted from the proposed ShareRobot dataset, as robotic benchmarks for multi-dimensional assessment, as shown in Table 6. The chosen baselines include MLLMs such as GPT-4V [2], LLaVA-OV-7B [36], and Qwen2-VL-7B [69], as well as robotic models like RoboMamba [42]. Detailed descriptions of the three selected robotic benchmarks and the analysis of each results are provided below:

- **RoboVQA** [60] provides a robotics VQA benchmark and a long-horizon planning benchmark with an intervention mechanism on real robots. Specifically, this benchmark includes 18,248 video-text pairs designed from 100 long-horizon episodes for various robotic VQA tasks, including planning, planning with context, planning remaining steps, future prediction, generative affordance, past description, success (positive/negative), and discriminative affordance (positive/negative). Similar to RoboMamba [42], we utilized BLEU-1$\sim$BLEU-4 to evaluate the average performance across all tasks. According to the evaluation results, our proposed model, *RoboBrain*, outperforms all baselines, achieving approximately 30% higher performance than the second-best model.
- **OpenEQA** [49] provides a robotics VQA benchmark with over 1,600 high-quality human-generated questions drawn from more than 180 real-world scenes, targeting the task of Embodied Question Answering (EQA) for environment understanding. For fairness, we evaluated all models using the prompt templates and the LLM-Score metric provided by OpenEQA [49]. Based on the evaluation results, our proposed model, *RoboBrain*, outperforms GPT-4V [2] overall and achieves comparable performance to other baselines. In the future, we plan to further enhance *RoboBrain*'s spatial intelligence to improve its generalization across scenes.
- **ShareRobot (Eval)** provides a cross-scene and cross-embodiment robotics benchmark consisting of 2,050 VQA pairs, drawn from 102 diverse scenes (e.g., bedroom, laboratory, kitchen, office) and covering 12 differ-

Table 6. Performance comparison on RoboVQA, OpenEQA and ShareRobot Benchmarks.

Dataset	**Split / Metric**	**RoboBrain (Ours)**	**GPT-4V** [2]	**LLaVA-OV-7B** [36]	**RoboMamba** [42]	**Qwen2-VL-7B** [69]
RoboVQA[60]	BLEU1	**72.05**	32.23	38.12	54.9	33.22
	BLEU2	**65.35**	26.51	33.56	44.2	26.11
	BLEU3	**59.39**	24.65	31.76	39.5	20.98
	BLEU4	**55.05**	23.94	30.97	36.3	17.37
OpenEQA[49]	OBJECT-STATE-RECOGNITION	70.4	63.2	72.02	-	**72.06**
	OBJECT-RECOGNITION	49.54	43.4	51.73	-	**61.91**
	FUNCTIONAL-REASONING	57.14	**57.4**	55.53	-	54.23
	SPATIAL-UNDERSTANDING	46.46	33.6	48.98	-	**50.39**
	ATTRIBUTE-RECOGNITION	66.7	57.2	**75.52**	-	73.88
	WORLD-KNOWLEDGE	53.12	50.7	56.46	-	**57.3**
	OBJECT-LOCALIZATION	**47.45**	42	45.25	-	47.29
ShareRobot (Eval)	DISCRIMINATIVE	**99.02**	-	57.9	-	76.47
	FUTURE-PREDICTION	**72.92**	-	13.1	-	8.04
	GENERATIVE	**32.43**	-	5.44	-	4.63
	PAST-DESCRIPTION	**37.07**	-	4.4	-	13.65
	PLANNING-REMAINING	**71.29**	-	24.5	-	7.56
	PLANNING-TASK	**52.43**	-	25	-	36.34
	PLANNING-WITH	**91.95**	-	44.25	-	45.12
	SUCCESS	**61.7**	-	58.5	-	54.63

ent robot bodies. Similar to RoboVQA [60], we categorized various robotic VQA tasks into planning, planning with context, planning remaining steps, future prediction, generative affordance, past description, success (positive/negative), and discriminative affordance (positive/negative). Unlike RoboVQA benchmark [60], we utilized GPT-4o [55] to score the evaluation results instead of BLEU metrics for each task, aiming for more accurate performance assessment. Based on the results, our proposed model, *RoboBrain*, outperforms all baselines, demonstrating its exceptional planning capabilities across diverse scenes and embodiments.

C.3. Effectiveness of ShareRobot

In this subsection, we investigate the effectiveness of the proposed ShareRobot dataset for training RoboBrain. We maintain the ratio of robotic data to general data used in the main body of the paper, approximately 4:6. Based on the original data source proportions, we randomly sampled 200K samples, which include:

- **Exp A** consists of 40% robotic data, with 20% sourced from ShareRobot and 20% from other robotic sources, along with 60% general data.
- **Exp B** consists of 40% robotic data, excluding ShareRobot, with the same other robotic data resampled as in Experiment A, resulting in a total of 40%. It also includes 60% general data, which is identical to that of Exp A.

We conducted a complete epoch for all the experiments mentioned above. The results are presented in Tab 7. As shown in the table, the inclusion of ShareRobot data enhances the model's performance compared to scenarios without ShareRobot.

C.4. Effectiveness of Robot Data Proportion

In this subsection, we investigate the effectiveness of the ratio of robotic data (including ShareRobot) to general data used in training RoboBrain. We maintain a constant total training dataset size of 200K while varying the sampling proportions of robotic and general data. The configurations are as follows:

- **Exp C** utilizes a ratio of 3:7, comprising 30% robotic data and 70% general data.
- **Exp D** utilizes a ratio of 4:6, comprising 40% robotic data and 60% general data, **same to Exp A**.
- **Exp E** utilizes a ratio of 5:5, with 50% robotic data and 50% general data.
- **Exp F** utilizes a ratio of 6:4, featuring 60% robotic data and 40% general data.
- **Exp G** utilizes a ratio of 7:3, containing 70% robotic data and 30% general data.

We conducted a complete epoch for all the experiments mentioned above. The results are presented in Tab 7. As shown in the table, a 4:6 ratio of robotic data represents a good choice for training data, effectively balancing performance on both the robotic and the general benchmark.

Table 7. Experimental results for effectiveness of ShareRobot and different robot data proportion.

Exp. Name	General Data (%)	Robotic Data (%)		General Benchmarks			Robotic Benchmarks			Average
	OneVision	ShareRobot	Others	Type-1	Type-2	Type-3	RoboVQA[60]	OpenEQA[49]	ShareRobot	
EXP A	60%	20%	20%	62.44	71.98	70.33	48.29	58.74	63.11	**62.48**
EXP B	60%	0%	40%	62.36	71.38	66.01	49.20	57.96	27.03	**55.66**
EXP C	70%	15%	15%	62.73	72.19	68.10	45.96	56.59	61.73	**61.22**
EXP D	60%	20%	20%	62.44	71.98	70.33	48.29	58.74	63.11	**62.48**
EXP E	50%	25%	25%	62.28	71.25	66.54	49.34	58.76	63.35	**61.92**
EXP F	40%	30%	30%	62.39	71.61	68.37	49.22	56.24	64.57	**62.07**
EXP G	30%	35%	35%	62.69	71.92	69.54	47.74	55.72	65.22	**62.14**

Table 8. **Additional Experimental Results.** "SFT Data (G:R)" indicates the ratio of training data for fine-tuning VLMs, where "G" represents general VQA data and "R" denotes robot data (with half being ShareRobot). The total dataset size is 1.47M.

	Model	SFT Data(G:R)	RoboVQA	ShareRobot	MME	MMMU
(a)	LLaVA-OV-7b	6:0	36.29	27.04	**2001**	**49.65**
		6:4	**43.63**	**54.66**	1945	48.83
	Qwen2VL-7B	6:0	24.05	28.17	**2313**	52.10
		6:4	**58.94**	**58.86**	2295	**52.33**
	OpenVLA-7B	6:0	4.11	21.44	1681	35.07
		6:4	**54.79**	**60.56**	**1722**	**37.25**
(b)	LLaVA1.5-Qwen	6:0	24.17	26.73	1720	44.28
		6:4	**49.01**	**43.41**	**1732**	**48.33**
	LLaVA1.5-LLaMA	6:0	21.40	25.06	1529	46.40
		6:4	**49.67**	**54.87**	**1722**	**43.41**
	LLaVA1.5-Vicuna	6:0	26.19	22.18	**1668**	30.09
		6:4	**50.40**	**51.42**	1650	**31.51**
	LLaVA1.5-Mistral	6:0	14.30	21.88	**1602**	23.91
		6:4	**36.29**	**57.47**	1548	**24.32**

C.5. Different Architecture and Pre-trained VLMs

To validate the effectiveness of different architecture and pre-trained VLMs and training data in the stage 3 training setup, we selected LLaVA-OneVision-7B [34], OpenVLA-7B [30], and Qwen2VL-7B [69], each representing a distinct architecture among VLMs, and conducted SFT using the same proportion of training data described in the main text. As shown in Tab. 8 (a), the results demonstrated that incorporating ShareRobot can significant performance improvements. For unaligned VLMs like LLaVA 1.5 [38] and OpenVLA, we first aligned the MLP using BLIP-558k [33]; other models were directly finetuned.

C.6. Different LLM Backbones

To demonstrate the effectiveness of different LLM backbones when finetuned on the ShareRobot dataset, we conducted experiments using four distinct LLMs [5, 16, 26, 66]. These models were finetuned using the ShareRobot data, and the experimental results are summarized in Tab. 8 (b). The findings indicate that different LLMs benefit from the ShareRobot data.

Table 9. Additional Evaluation Results.

Stage	RoboVQA	ShareRobot	MME	MMMU	Affordance↑	Trajectory↓
S1.5	2.60	9.81	1406	46.00	0.00	1.00
S2-si	28.90	13.31	**2110**	**50.76**	3.11	1.00
S2-ov	31.81	34.84	2083	49.95	8.50	1.00
S3	**62.96**	**65.05**	2084	49.00	7.14	1.00
S4-A	**62.96**	**65.05**	2084	49.00	**27.1**	-
S4-T	**62.96**	**65.05**	2084	49.00	-	**0.09**

C.7. Ablation Studies of Different Stages

We present the evaluation results for each stage in Tab. 9. The results demonstrate that staged training from stage 1 to stage 3 consistently and effectively improves the model's planning performance, while stage 4 enhances the model's affordance and trajectory capabilities.

D. More Qualitative Results

In this section, we provide additional visual results for planning, affordance perception, and trajectory prediction. This includes the presentation of both positive and negative samples, as well as further analysis.

D.1. Visualization on Planning

Here, we present additional embodied planning for robotic tasks generated by RoboBrain, as shown in Fig. 8. In this figure, we demonstrate the planning results of RoboBrain for four distinct robotic manipulation tasks: ”Water plants”, ”Put the pot in the drawer”, ”Cluster blocks of the same color into different corners”, and ”Clean the desk”, where the first three are categorized as good cases, and the last one as a bad case. Additionally, the model provides a rationale and detailed explanation for each step of the planning process across all four cases.

From the first three planning cases, it is evident that RoboBrain effectively utilizes environmental information and the states of interactive objects—captured from first- or third-person perspective images—to generate task plans for various types of robotic manipulation tasks. Notably, in the ”Cluster blocks of the same color into different corners” task, RoboBrain not only analyzes the number of blocks of each color on the table in Steps 1 and 2 but also provides detailed sub-steps in Step 3, i.e., *”Move the objects to form clusters”*. Specifically, it plans the movement of blocks of four different colors to their designated locations: *”top left corner”*, *”top right corner”*, *”bottom left corner”*, and *”bottom right corner”*. The exceptional task generalization capability of RoboBrain in planning further validates the effectiveness of our training dataset—including the proposed ShareRobot dataset—and the Multi-Phase training strategy.

We also present a bad case for RoboBrain, namely the ”Clean the desk” task. In this case, the first-person perspective image depicts a work desk spilled with coffee, where the main objects of focus include a *”tissue box”*, a *”tipped-over coffee cup”*, and the *”spilled coffee liquid”*. The errors in the planning results inferred by RoboBrain are summarized as follows: **(1) Object recognition error.** The only available object for wiping the desk in the image is a *”tissue”*, rather than a *”disinfectant wipe”*. **(2) Omission of critical steps.** Before wiping the desk, it is necessary to extract a tissue from the tissue box. However, this step is missing in RoboBrain’s planning. **(3) Action decision deviation.** In Step 2, i.e., *”Wipe down the desk with a disinfectant wipe”*, the detailed description states, *”Start from one end of the desk and move to the other”*. This implies that RoboBrain fails to prioritize wiping the *”spilled coffee liquid”* specifically, focusing instead on cleaning *”the entire desk”*. The primary cause might be the similarity in color between the desk and the spilled coffee, making it difficult for the model to distinguish.

In our extensive testing, although a small number of unreasonable bad cases like the one described above were observed, RoboBrain demonstrated robust planning capabilities in the vast majority of cases. This provides a solid foundation for executing long-horizon manipulation tasks.

D.2. Visualization on Affordance

Here, we present the visualizations of RoboBrain’s perception of affordance areas, as shown in Fig.9. The text below each subfigure indicates the task instructions, while the red bounding boxes represent the affordance areas predicted by the RoboBrain model. The visualizations in the first three rows demonstrate that our RoboBrain model can effectively provide reasonable affordance areas based on human instructions and visual information. For example, given the instruction “drink_with the bottle”, RoboBrain can determine that the bottle cap is in a closed state, thus providing affordance information for the cap area. This highlights RoboBrain’s strong understanding of abstract instructions.

We also present several failure cases, as illustrated in the fourth row of Fig.9. These include misidentified objects, interference from other objects in the scene, and instances where no objects were recognized. These issues may stem from the model’s limited ability to perceive and localize in noisy environments.

D.3. Visualization on Trajectory

Here, we present additional visualizations generated by RoboBrain using start points, as shown in Fig.10. In this figure, the red-to-purple gradient curves represent the ground truth, while the green-to-blue gradient curves indicate the predicted trajectories. For clarity, waypoints are omitted. The first three rows demonstrate that, regardless of the complexity of the end-effector trajectory, RoboBrain accurately predicts 2D trajectories based on visual observations and task instructions. These predictions closely align with the structure of the ground truth and remain executable.

Additionally, RoboBrain’s predictions often capture the essential features of the trajectories, leading to smoother and potentially more efficient paths compared to the ground truth. This improvement may stem from the inherent variability in the robot’s actual trajectories, which can include redundant waypoints under similar manipulation scenarios. By learning from a large, embodied dataset and utilizing the reasoning capabilities of large language models, RoboBrain is able to infer effective and optimized execution paths.

The visualizations in the third row further suggest that RoboBrain avoids overfitting; it generalizes well across different scenarios, producing trajectories that are both executable and reasonable.

We also present several failure cases, as shown in the fourth row of Fig. 10. These include the robot’s end-effector failing to accurately locate the cup, neglecting the articulated nature of the fridge door while opening it, and not accounting for the deformable properties of clothing during folding. These examples highlight the need for improved spatial perception, as well as the incorporation of object-specific physical constraints and world knowledge to generate more feasible and realistic trajectories.

E. Details of ShareRobot Dataset

In the previous section, we introduced the process of collecting and annotating our ShareRobot dataset. Here, we will provide detailed prompts for data labeling and display examples.

E.1. Prompts

The prompts used for data planning labeling are shown in Fig.11.

E.2. High-level Descriptions Examples

In our ShareRobot dataset, there are 10,290 long-horizon high-level descriptions. We provide the 50 most frequently occurring ones below.

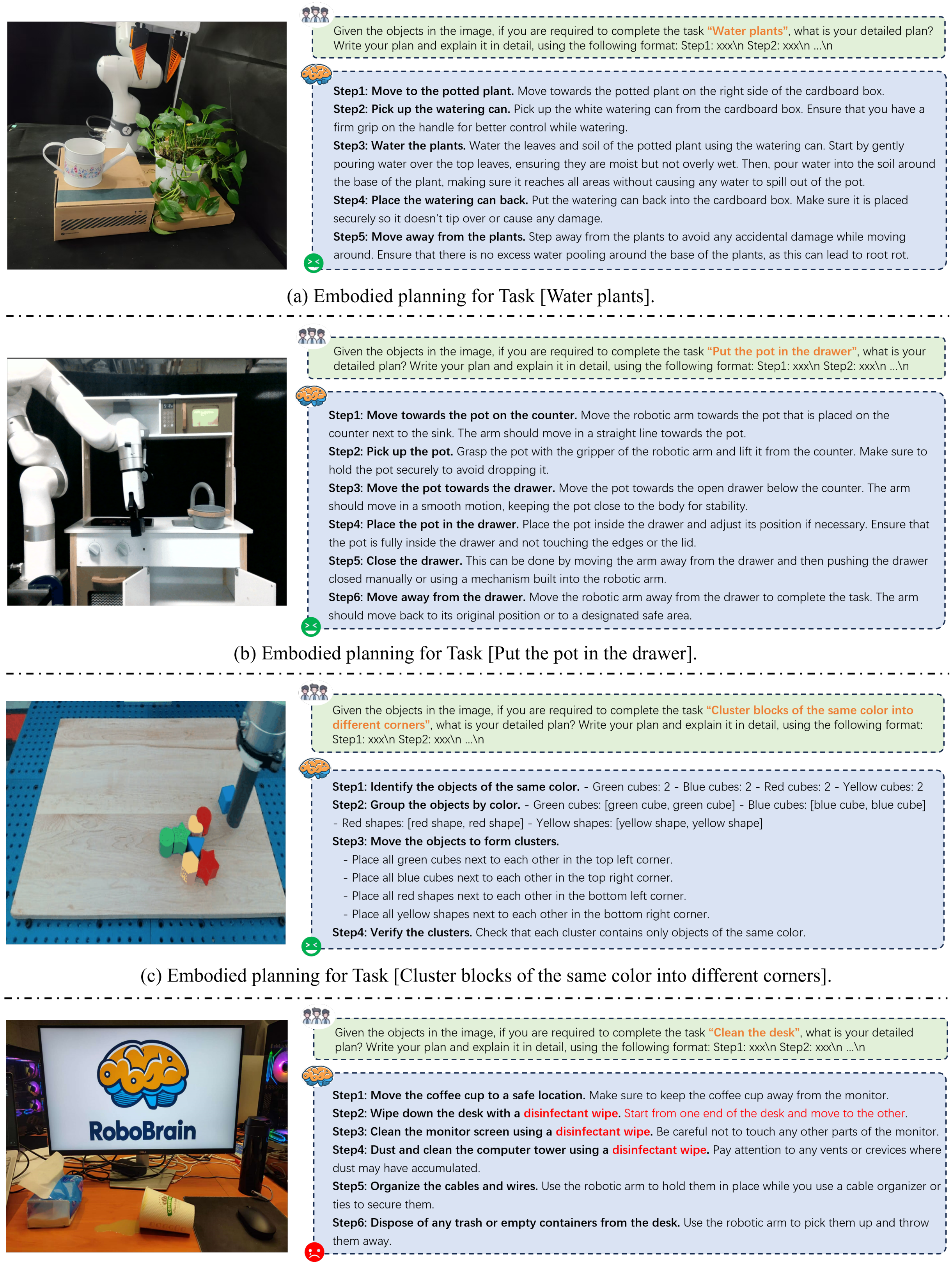
- Closing a drawer
- Opening a drawer
- Opening a cabinet door
- Dragging a strainer across a table

- Picking up a bowl
- Inserting a three-pronged object into its matching slot
- Inserting a double-square object into its matching slot
- Opening a door
- Closing a cabinet door
- Inserting a star-shaped object into its corresponding slot
- Opening a laptop
- Inserting an oval object into its corresponding slot
- Picking up a ketchup bottle from a table
- Moving a banana from a plate to a table
- Closing a door
- Switching a light switch
- Inserting an arch-shaped object into its corresponding slot
- Inserting a square-circle object into its matching slot
- Dragging a strainer backwards
- Dragging a mug from left to right
- Dragging a mug forward
- Picking up a red object from a table
- Placing a ketchup bottle onto a plate
- Placing a bowl inside an oven
- Inserting a hexagonal object into its corresponding slot
- Closing a microwave door
- Moving a banana from a table to a plate
- Turning on a toaster
- Opening a microwave
- Closing an oven door
- Making tea
- Dragging a strainer forward
- Placing a bowl into an oven
- Picking up a banana and placing it in a mug
- Inserting an arch-shaped object into its matching slot
- Closing a tea container
- Inserting a green object into a designated slot
- Picking up a banana and placing it in a strainer
- Moving a cloth to the left side of a table
- Dragging a mug backwards
- Placing a bottle into a pot
- Dragging a strainer forwards
- Inserting a rectangular prism into its matching slot
- Opening a refrigerator
- Opening a tea container
- Opening a double door
- Inserting a cylinder into a matching hole
- Picking up a piece of toast
- Dragging a mug from left to right on a table
- Closing a refrigerator door

E.3. Low-level Instructions Examples

Our ShareRobot dataset contains 28,181 low-level instructions. The top 50 frequency occurrences are displayed below.

- Grasp the ketchup bottle
- Reach for the ketchup bottle
- Grasp the banana
- Lift the ketchup bottle
- Lift the banana
- Reach for the strainer
- Reach for the banana
- Reach for the mug
- Grasp the mug
- Lift the pot
- Lift the bowl
- Pull the drawer open
- Reach for the bowl
- Reach for the pot
- Grasp the strainer
- Reach for the drawer handle
- Grasp the handle
- Lift the spoon
- Grasp the bowl
- Reach for the spoon
- Place the ketchup bottle on the table
- Release the banana
- Reach the drawer
- Place the banana on the table
- Lift the mug
- Reach the cabinet door
- Grasp the pot
- Grasp the strainer
- Grasp the drawer handle
- Release the mug
- Grasp the pot
- Grasp the spoon
- Place the mug down
- Move the banana towards the table
- Grasp the bowl
- Pull the drawer closed
- Move towards the bowl
- Reach for the cloth
- Release the pot
- Grasp the bottle
- Lift the cloth
- Lift the red object
- Grasp the banana
- Lift the butter
- Reach for the banana on the table
- Place the ketchup bottle on the plate
- Grasp the drawer handle
- Move the ketchup bottle towards the plate
- Move towards the red object
- Place the banana on the plate



Given the objects in the image, if you are required to complete the task "Water plants", what is your detailed plan? Write your plan and explain it in detail, using the following format: Step1: xxx\n Step2: xxx\n ...\n

Step1: Move to the potted plant. Move towards the potted plant on the right side of the cardboard box.
Step2: Pick up the watering can. Pick up the white watering can from the cardboard box. Ensure that you have a firm grip on the handle for better control while watering.
Step3: Water the plants. Water the leaves and soil of the potted plant using the watering can. Start by gently pouring water over the top leaves, ensuring they are moist but not overly wet. Then, pour water into the soil around the base of the plant, making sure it reaches all areas without causing any water to spill out of the pot.
Step4: Place the watering can back. Put the watering can back into the cardboard box. Make sure it is placed securely so it doesn't tip over or cause any damage.
Step5: Move away from the plants. Step away from the plants to avoid any accidental damage while moving around. Ensure that there is no excess water pooling around the base of the plants, as this can lead to root rot.

(a) Embodied planning for Task [Water plants].

Given the objects in the image, if you are required to complete the task "Put the pot in the drawer", what is your detailed plan? Write your plan and explain it in detail, using the following format: Step1: xxx\n Step2: xxx\n ...\n

Step1: Move towards the pot on the counter. Move the robotic arm towards the pot that is placed on the counter next to the sink. The arm should move in a straight line towards the pot.
Step2: Pick up the pot. Grasp the pot with the gripper of the robotic arm and lift it from the counter. Make sure to hold the pot securely to avoid dropping it.
Step3: Move the pot towards the drawer. Move the pot towards the open drawer below the counter. The arm should move in a smooth motion, keeping the pot close to the body for stability.
Step4: Place the pot in the drawer. Place the pot inside the drawer and adjust its position if necessary. Ensure that the pot is fully inside the drawer and not touching the edges or the lid.
Step5: Close the drawer. This can be done by moving the arm away from the drawer and then pushing the drawer closed manually or using a mechanism built into the robotic arm.
Step6: Move away from the drawer. Move the robotic arm away from the drawer to complete the task. The arm should move back to its original position or to a designated safe area.

(b) Embodied planning for Task [Put the pot in the drawer].

Given the objects in the image, if you are required to complete the task "Cluster blocks of the same color into different corners", what is your detailed plan? Write your plan and explain it in detail, using the following format: Step1: xxx\n Step2: xxx\n ...\n

Step1: Identify the objects of the same color. - Green cubes: 2 - Blue cubes: 2 - Red cubes: 2 - Yellow cubes: 2
Step2: Group the objects by color. - Green cubes: [green cube, green cube] - Blue cubes: [blue cube, blue cube] - Red shapes: [red shape, red shape] - Yellow shapes: [yellow shape, yellow shape]
Step3: Move the objects to form clusters.
- Place all green cubes next to each other in the top left corner.
- Place all blue cubes next to each other in the top right corner.
- Place all red shapes next to each other in the bottom left corner.
- Place all yellow shapes next to each other in the bottom right corner.

Step4: Verify the clusters. Check that each cluster contains only objects of the same color.

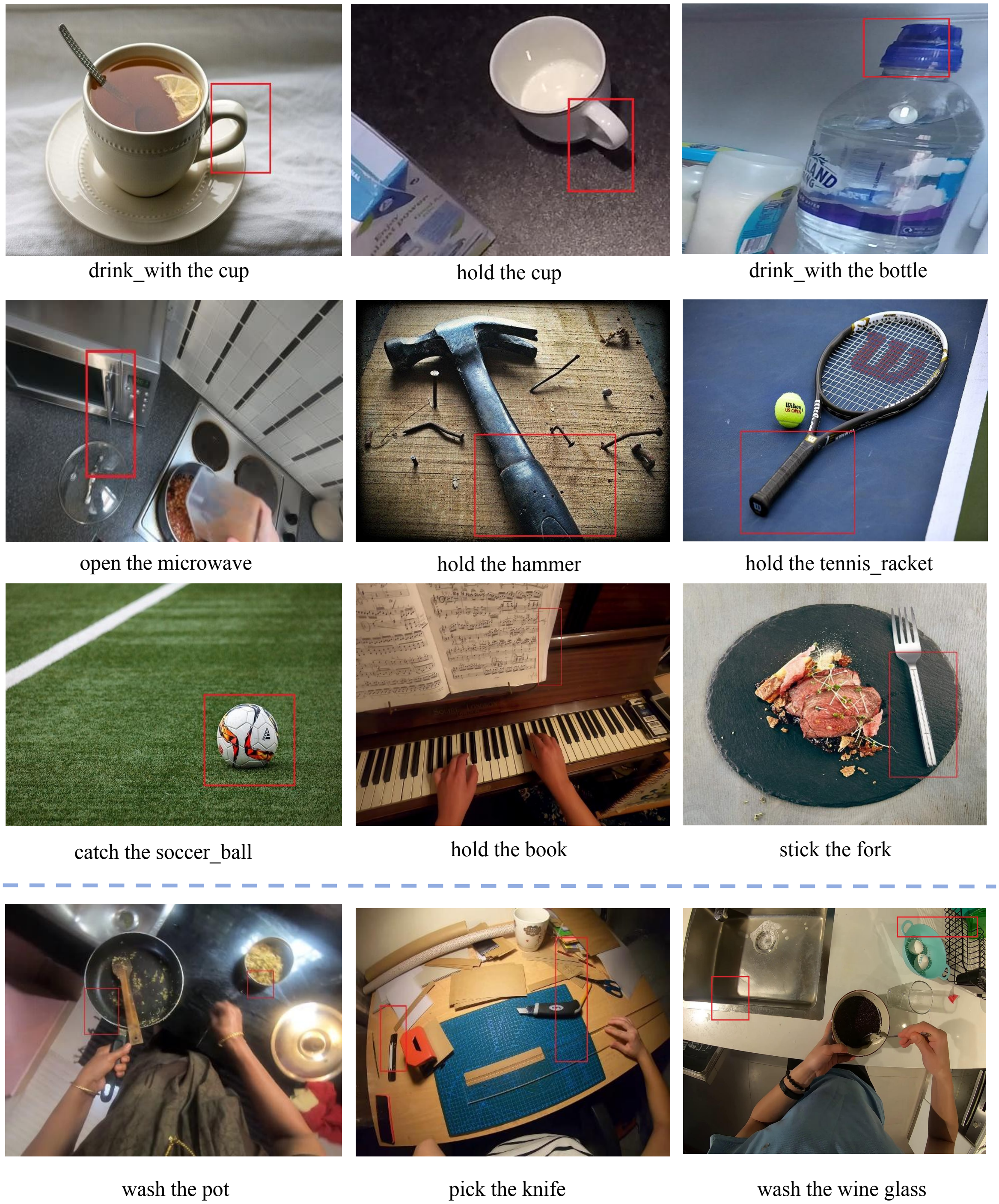
(c) Embodied planning for Task [Cluster blocks of the same color into different corners].

Given the objects in the image, if you are required to complete the task "Clean the desk", what is your detailed plan? Write your plan and explain it in detail, using the following format: Step1: xxx\n Step2: xxx\n ...\n

Step1: Move the coffee cup to a safe location. Make sure to keep the coffee cup away from the monitor.
Step2: Wipe down the desk with a disinfectant wipe. Start from one end of the desk and move to the other.
Step3: Clean the monitor screen using a disinfectant wipe. Be careful not to touch any other parts of the monitor.
Step4: Dust and clean the computer tower using a disinfectant wipe. Pay attention to any vents or crevices where dust may have accumulated.
Step5: Organize the cables and wires. Use the robotic arm to hold them in place while you use a cable organizer or ties to secure them.
Step6: Dispose of any trash or empty containers from the desk. Use the robotic arm to pick them up and throw them away.

(d) Embodied planning for Task [Clean the desk].

Figure 8. **Additional embodied planning of RoboBrain.** (a)∼(c) show some good cases of RoboBrain's embodied planning, while (d) shows its bad case. More detailed analysis can be found in Sec.D.1.



drink_with the cup

hold the cup

drink_with the bottle

open the microwave

hold the hammer

hold the tennis_racket

catch the soccer_ball

hold the book

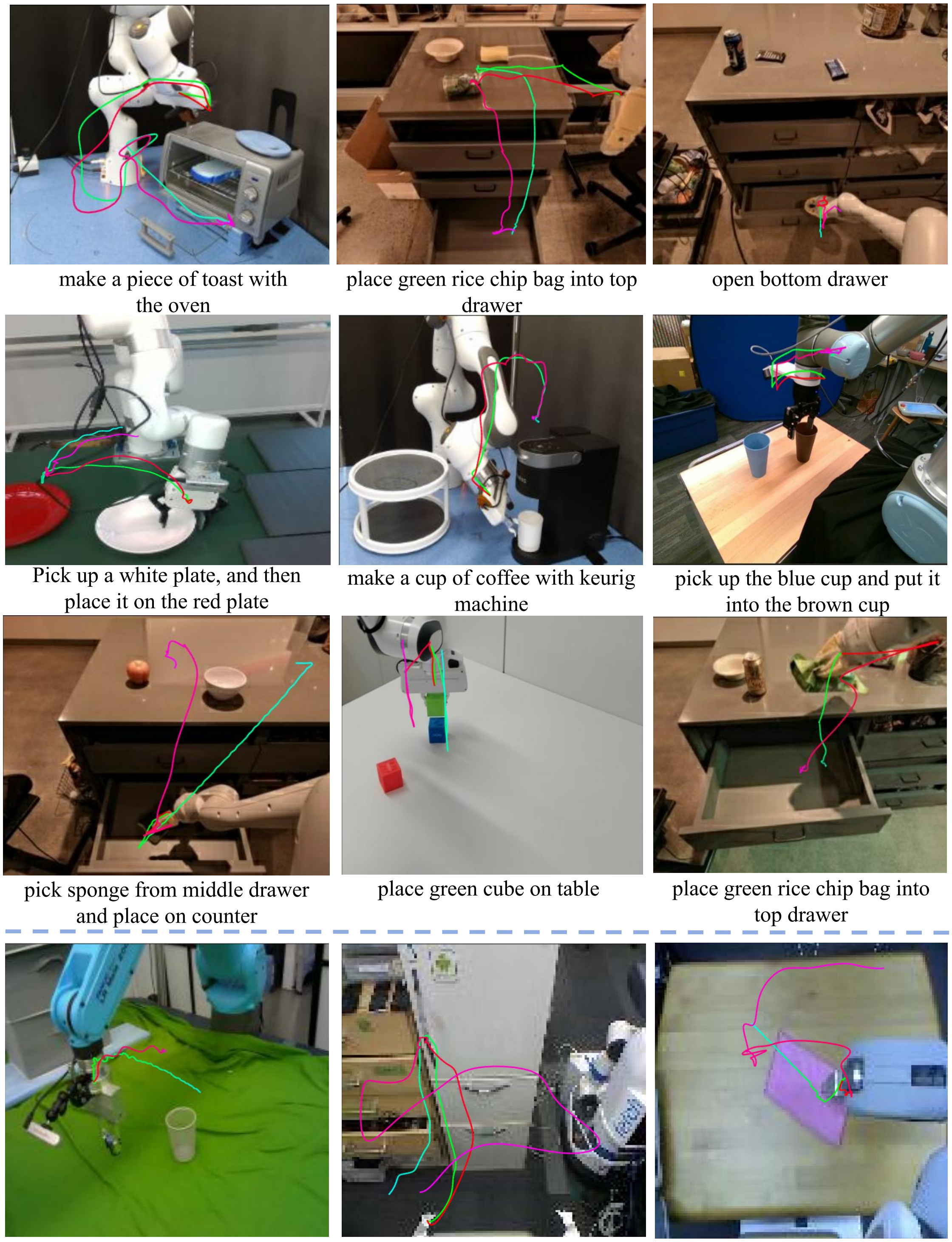
stick the fork

wash the pot

pick the knife

wash the wine glass

Figure 9. **Additional visualizations of diverse affordance areas.** The text below each subfigure indicates the task instructions, while the red bounding boxes represent the affordance areas predicted by the RoboBrain model. The visualizations in the first three rows demonstrate that our RoboBrain model effectively identifies reasonable affordance areas based on human instructions and visual information. The fourth row presents several failure cases, which may stem from the model's lack of ability to perceive and localize in noisy environments. This limitation could be attributed to the absence of such scenarios in the training data used during Stage 4. The complete prompt provided to RoboBrain is: ”You are a Franka robot using joint control. The task is $TASK. Please predict all possible affordance areas of the end effector.” Here, $TASK represents specific task instructions, such as ”drink with the cup.”



make a piece of toast with the oven

place green rice chip bag into top drawer

open bottom drawer

Pick up a white plate, and then place it on the red plate

make a cup of coffee with keurig machine

pick up the blue cup and put it into the brown cup

pick sponge from middle drawer and place on counter

place green cube on table

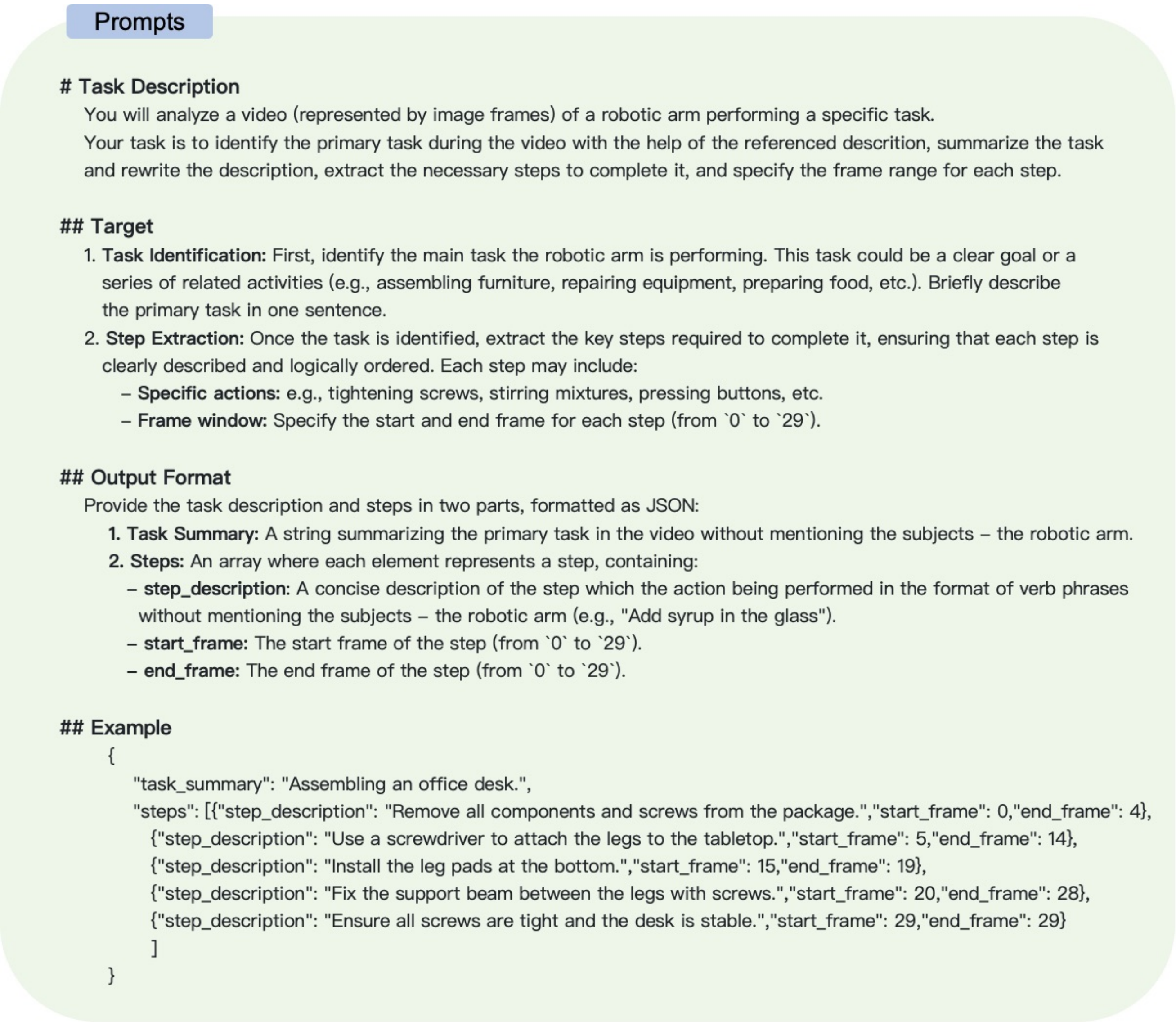
place green rice chip bag into top drawer

Pick up the object on the table and place it in the cup

opening the fridge

folding a cloth

Figure 10. **Additional visualizations of diverse 2D trajectories.** The red-to-purple gradient curves represent the ground truth, while the green-to-blue gradient curves indicate the predicted trajectories. The visualizations in the first two rows demonstrate that our RoboBrain model effectively generates end-effector manipulation curves based on the robot's observations and task instructions. The third row shows that RoboBrain is not merely fitting trajectories but also exhibits the ability to generate more reasonable and feasible curves. The fourth row presents some failure cases, which stem from a lack of spatial awareness and world knowledge. These limitations result in an inability to accurately localize the objects involved in interactions, account for physical constraints, and adapt to the variability of deformable objects.



Prompts

```
# Task Description
  You will analyze a video (represented by image frames) of a robotic arm performing a specific task.
  Your task is to identify the primary task during the video with the help of the referenced descrition, summarize the task
  and rewrite the description, extract the necessary steps to complete it, and specify the frame range for each step.

## Target
  1. Task Identification: First, identify the main task the robotic arm is performing. This task could be a clear goal or a
     series of related activities (e.g., assembling furniture, repairing equipment, preparing food, etc.). Briefly describe
     the primary task in one sentence.
  2. Step Extraction: Once the task is identified, extract the key steps required to complete it, ensuring that each step is
     clearly described and logically ordered. Each step may include:
       – Specific actions: e.g., tightening screws, stirring mixtures, pressing buttons, etc.
       – Frame window: Specify the start and end frame for each step (from `0` to `29`).

## Output Format
  Provide the task description and steps in two parts, formatted as JSON:
    1. Task Summary: A string summarizing the primary task in the video without mentioning the subjects – the robotic arm.
    2. Steps: An array where each element represents a step, containing:
      – step_description: A concise description of the step which the action being performed in the format of verb phrases
        without mentioning the subjects – the robotic arm (e.g., "Add syrup in the glass").
      – start_frame: The start frame of the step (from `0` to `29`).
      – end_frame: The end frame of the step (from `0` to `29`).

## Example
    {
      "task_summary": "Assembling an office desk.",
      "steps": [{"step_description": "Remove all components and screws from the package.","start_frame": 0,"end_frame": 4},
        {"step_description": "Use a screwdriver to attach the legs to the tabletop.","start_frame": 5,"end_frame": 14},
        {"step_description": "Install the leg pads at the bottom.","start_frame": 15,"end_frame": 19},
        {"step_description": "Fix the support beam between the legs with screws.","start_frame": 20,"end_frame": 28},
        {"step_description": "Ensure all screws are tight and the desk is stable.","start_frame": 29,"end_frame": 29}
        ]
    }
```

Figure 11. **Additonal visualizations of prompts for Gemini.** The prompts encapsulate the task description for robotic arm action recognition, the components of the target, and the desired response format. Additionally, an example is included to assist Gemini in understanding the specific task.