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

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. Sec. F discusses potential future research directions for RoboBrain.

A. Details of Models and Training

Model Setting. RoboBrain is built upon the LLaVA [27] framework and consists of three main components: the visual encoder, projector, and large language model (LLM).

For the visual encoder, we utilized the SigLIP [59] model, specifically the siglip-so400m-patch14-384, which is pre-trained on WebLi [8] at a resolution of 384x384. The SigLIP model improves upon traditional CLIP [14, 41] 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 [25] 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 [49] model, which is a state-of-the-art open-source LLM that is part of the latest Qwen series [3]. 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 [15] to train RoboBrain, enabling it to acquire affordance perception and trajectory prediction capabilities. LoRA 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. 1. We primarily referenced the training strategy of LLaVA-Onevision [23], 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.

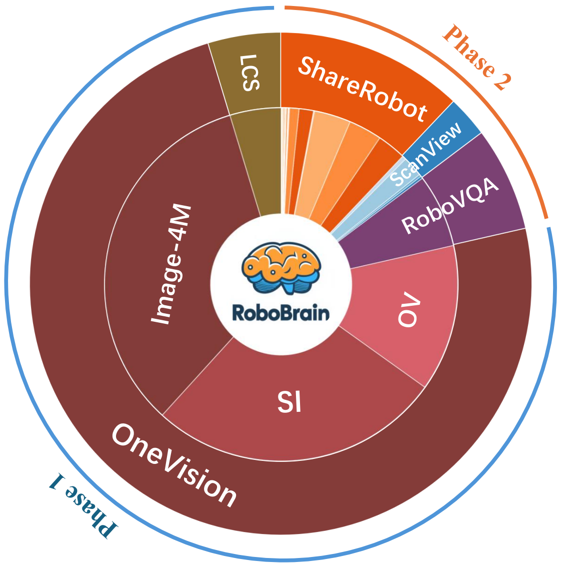


Figure 1. The distribution of the entire training dataset.

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. 1.

- **LCS-558K** is a subset of the LAION/CC/SBU dataset [5, 42], specifically designed as the LLaVA Visual Instruct Pretrain [27] Dataset. This dataset has been filtered to ensure a balanced distribution of concept coverage, providing diverse and representative visual content. The primary purpose of LCS-558K is to facilitate alignment between the visual encoder and the LLM, enabling the LLM to understand visual information.
- **Image-4M** comprises 8 data sources, including 3 from the LLaVA-Recap series [22]: BLIP558K, COCO118K, and CC3M, as well as UReader [57], Instruct Azure DC [22], Evol-Instruct [6], and SynthDog [19] We utilized the download links provided by the LLaVA-OneVision team for the data acquisition.
- **SI-3.2M**[1] [23] consists of 3.2 million samples, carefully

[1]Due to the unavailability of certain datasets, the actual data used

Table 1. 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$\times${2$\times$2}	Max 384$\times${6$\times$6}	Max 384$\times${6$\times$6}	Max 384$\times${6$\times$6}	Max 384$\times${6$\times$6}	Max 384$\times${6$\times$6}
	#Tokens	729	Max 729$\times$5	Max 729$\times$37	Max 729$\times$37	Max 729$\times$37	Max 729$\times$37	Max 729$\times$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: 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
	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

curated to support multimodal learning. It includes subsets from existing datasets such as Cambrian [50], Cauldron [22], and UReader [57], which were subjected to cleaning and re-annotation to ensure data quality. Additionally, it incorporates single-image data from sources like AI2D [18] and OKVQA [37], alongside a newly compiled single-image collection designed to achieve a balanced and diverse dataset.

- **OV-1.6M**[2] [23] comprises 1.6 million samples, which includes approximately 800K high-quality samples re-sampled 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** [43] 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.

amounts to 3.1M.

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

- **ScanView-318K** totals 318K samples, which integrates data from several high-quality sources, including MMScan-224K [34], 3RScan-43K [52], ScanQA-25K [2], and SQA3D-26K [35], 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 MLLMs, 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.

Table 2. Performance comparison on multiple general benchmarks.

Dataset	Split	RoboBrain (Ours)	GPT-4V [1]	LLaVA-OV-7B [25]	InternVL2-8B [9]	Qwen2-VL-7B [54]	GPT-4o [40]
A12D[18]	test	82.03	78.2	81.4	83.8	-	**94.2**
ChartQA[38]	test	80.48	78.5	80	83.3	83	**85.7**
DocVQA[39]	test	88	88.4	87.5	91.6	**94.5**	92.8
TextVQA[44]	val	75.85	-	71.07	77.4	**84.3**	-
MMMU[58]	val	49	56.8	48.8	51.8	54.1	**69.1**
MMStar[7]	test	61.23	57.1	61.7	61.5	60.7	**63.9**
OCRBench[31]	-	677	656	697	794	**845**	805
RealWorldQA[55]	test	68.89	61.4	66.3	64.4	**70.1**	58.6
SeedBench[21]	image	71.03	49.9	75.4	**76.2**	-	**76.2**
MMbench[32]	en-dev	81.52	81.3	83.2	-	-	**83.4**
MMbench[32]	en-test	80.44	75	80.8	81.7	**83**	-
MME[11]	test	2084	1926	1998	2210	**2327**	-

C.1. More Results on General Benchmarks

To evaluate performance on general tasks in real-world scenarios, as is commonly done with MLLMs [1, 9, 25, 40, 54], we conducted experiments using a diverse set of image benchmarks summarized in Table 2. We leveraged the comprehensive evaluation toolkit, LMMs-Eval[61, 63], 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 such as AI2D [18], ChartQA [38], DocVQA [39], and OCRBench [31] were utilized. Open-source models like InternVL2-8B [9] and LLAVA-OV-7B [25] have demonstrated comparable performance to closed-source models such as GPT-4V [1]. For *RoboBrain*, despite being optimized primarily for multidimensional robotic tasks, it surpasses LLAVA-OV-7B [25] and GPT-4V [1] on these benchmarks, achieving a significant improvement in structured OCR tasks, with the only exceptions being DocVQA [39], where it performs slightly lower than GPT-4V [1], and OCRBench [31], where it falls slightly behind LLAVA-OV-7B [25].
- **Visual Perception and Multi-domain Reasoning.** This category focuses on complex visual perception and multidisciplinary reasoning tasks. Benchmarks for visual perception include MMStar [7], MMBench [32], and MME [11], while reasoning benchmarks include MMMU [58] and SeedBench [21]. *RoboBrain* demonstrates comparable performance to GPT-4V [1] and LLAVA-OV-7B [25] across multiple benchmarks.
- **Real-world Understanding and Interaction.** Evaluating MLLMs [1, 9, 25, 40, 54] as general-purpose assistants in real-world settings is crucial, as these scenarios extend beyond controlled environments. For this, the RealworldQA [55] benchmark was utilized. Results indicate that *RoboBrain* not only outperforms open-source models like LLAVA-OV-7B [25] and InternVL2-8B [9], but also exceeds closed-source models such as GPT-4V [1] and GPT-4o [40], 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 [43], OpenEQA [36], and the test set of ShareRobot, extracted from the proposed ShareRobot dataset, as robotic benchmarks for multi-dimensional assessment, as shown in Table 3. The chosen baselines include MLLMs such as GPT-4V [1], LLaVA-OV-7B [25], and Qwen2-VL-7B [54], as well as robotic models like RoboMamba [28]. Detailed descriptions of the three selected robotic benchmarks and the analysis of each results are provided below:

- **RoboVQA** [43] 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 [28], 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** [36] 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

Table 3. Performance comparison on RoboVQA, OpenEQA and ShareRobot Benchmarks.

Dataset	Split / Metric	RoboBrain (Ours)	GPT-4V [1]	LLaVA-OV-7B [25]	RoboMamba [28]	Qwen2-VL-7B [54]
RoboVQA[43]	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[36]	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

metric provided by OpenEQA [36]. Based on the evaluation results, our proposed model, *RoboBrain*, outperforms GPT-4V [1] 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 different robot bodies. Similar to RoboVQA [43], 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 [43], we utilized GPT-4o [40] 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 4. As shown in the table, the inclusion of ShareRobot data enhances the model's performance compared to scenarios without ShareRobot. This highlights ShareRobot's key role in enhancing RoboBrain's planning capabilities.

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

Table 4. EExperimental results demonstrating the effectiveness of different task types. Type-1 refers to Chart, Diagram, and Document Understanding; Type-2 pertains to Visual Perception and Multi-domain Reasoning; Type-3 encompasses Real-world Understanding and Interaction. For detailed task descriptions, please refer to C.1.

Exp. Name	**General Data (%)**	**Robotic Data (%)**		**General Benchmarks**			**Robotic Benchmarks**			**Average**
	OneVision	**ShareRobot**	**Others**	**Type-1**	**Type-2**	**Type-3**	**RoboVQA[43]**	**OpenEQA[36]**	**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 5. **Additional Experimental Results.** "SFT Data (G:R)" indicates the ratio of training data for fine-tuning MLLMs, 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**

Table 6. Additional Evaluation Results. "S1.5" refers to Stage 1.5, and this notation applies similarly to other stages.

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**

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 4. As shown in the table, a 4:6 ratio of robotic data is an effective choice for training, balancing performance on both the robotic and general benchmarks.

C.5. Different Architecture and MLLMs

To validate the effectiveness of different architecture and pre-trained MLLMs and training data in the stage 3 training setup, we selected LLaVA-OV-7B [23], OpenVLA-7B [20], and Qwen2VL-7B [54], each representing a distinct architecture among MLLMs, and conducted supervised fine-tuning (SFT) using the same proportion of training data described in the main text. As shown in Tab. 5 (a), the results demonstrated that incorporating ShareRobot can significant performance improvements. For unaligned MLLMs such as LLaVA 1.5 [26] and OpenVLA, we first aligned the MLP using BLIP-558k [22] before fine-tuning. In contrast, other models that already have aligned vision encoder and LLM were directly fine-tuned.

C.6. Different LLM Backbones

To demonstrate the effectiveness of different LLM backbones when fine-tuned on the ShareRobot dataset, we conducted experiments using four distinct LLMs [3, 10, 17, 51]. These models were fine-tuned using the ShareRobot data, and the experimental results are summarized in Tab. 5 (b). The findings indicate that different LLMs benefit from the ShareRobot data.

C.7. Ablation Studies of Different Stages

We present the evaluation results for each stage in Tab. 6. 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. 2. 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.3. 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.3. 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.4. 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. 4. 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 templates used during data generation. Additionally, we will display some high-level descriptions and low-level instructions examples.

E.1. Prompts

The prompts we used for Gemini [47] in data labeling are shown in Fig.5.

E.2. Templates of Question Types

In the process of planning data generation, the templates used to generate question-answer pairs are shown in Fig.6.

E.3. High-level Descriptions Examples

Our ShareRobot dataset contains 10,290 long-horizon high-level descriptions. Below, we present the 30 most frequently occurring ones.

- 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

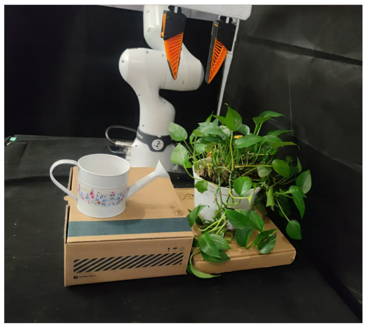
E.4. Low-level Instructions Examples

Our ShareRobot dataset contains 28,181 low-level instructions, with the top 30 occurrences 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

F. Future Work

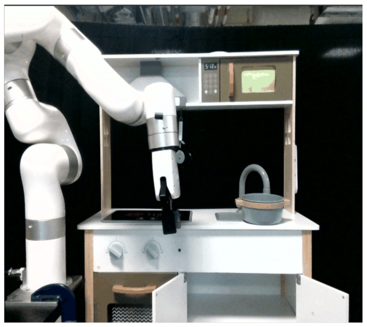
In future research, we aim to enhance various capabilities of RoboBrain, including spatial understanding [24, 56], embodied reasoning [12, 46, 62], tool utilization [29, 53], and long-text comprehension [13, 45, 48]. We will ensure that these capabilities are effectively integrated into downstream action models for application in real-world scenarios. Moreover, we will consider the issues of model efficiency [4, 28] and safety [16, 30, 33, 60], as constructing a RoboBrain that is both efficient in reasoning and secure will be a focal point of our future research.



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.

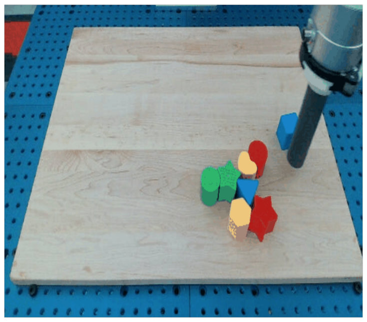
(a) Embodied planning for Task [Water plants].



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].

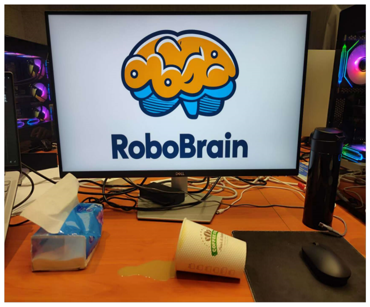


Given the objects in the image, if you are required to complete the task "Cluster blocks of the same color into different corners", what is your detailed plan? Write your plan and explain it in detail, using the following format: Step1: xxx\n Step2: xxx\n ...\n

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 2. **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.

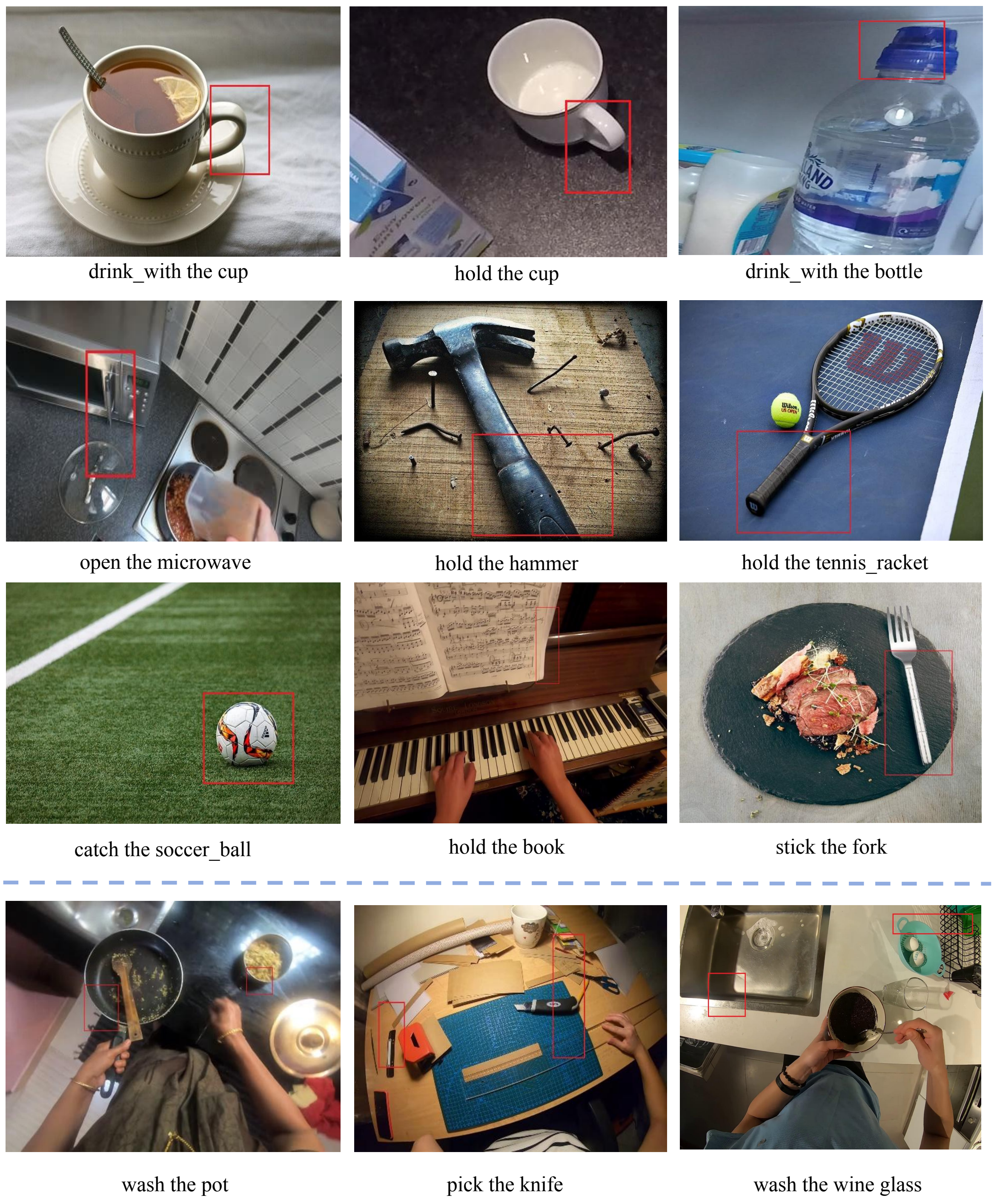
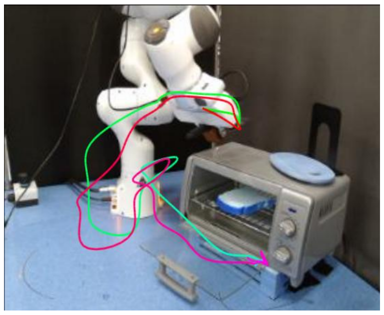
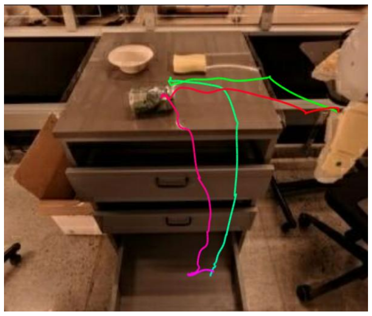
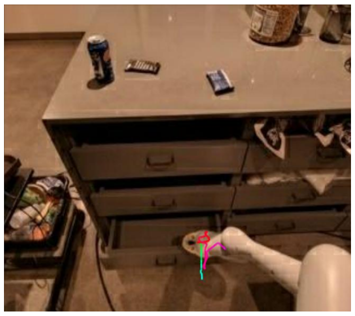
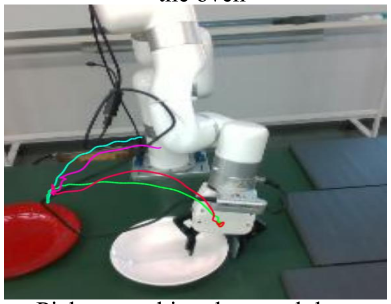


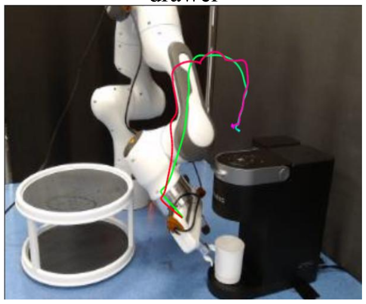
Figure 3. **Additional visualizations of diverse affordance areas.** The text below each subfigure indicates the task instructions, while the red bounding boxes represent the affordance areas predicted by the RoboBrain model. The visualizations in the first three rows demonstrate that our RoboBrain model 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."

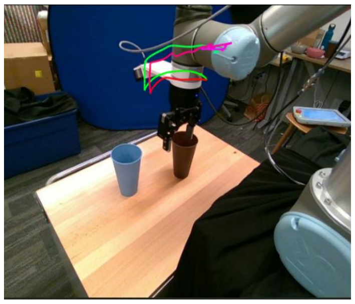

make a piece of toast with the oven

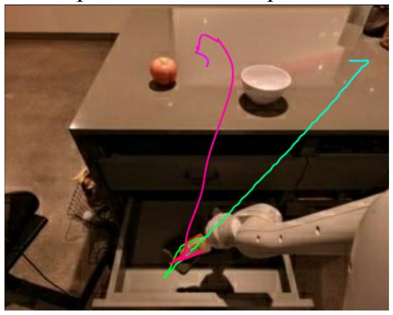

place green rice chip bag into top drawer

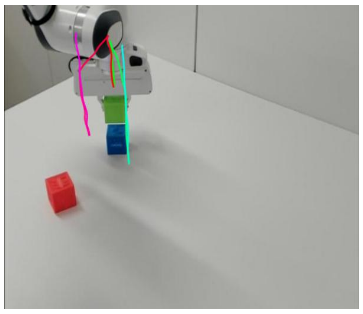

open bottom drawer

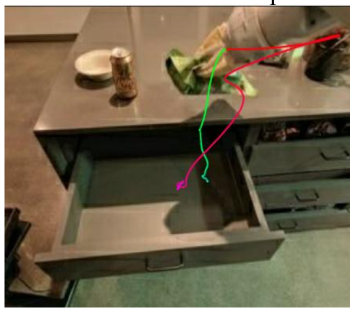

Pick up a white plate, and then place it on the red plate

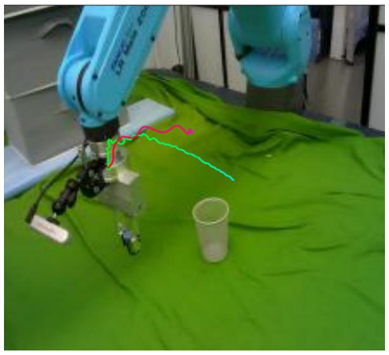

make a cup of coffee with keurig machine

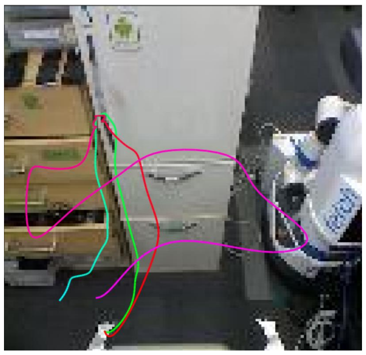

pick up the blue cup and put it into the brown cup


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

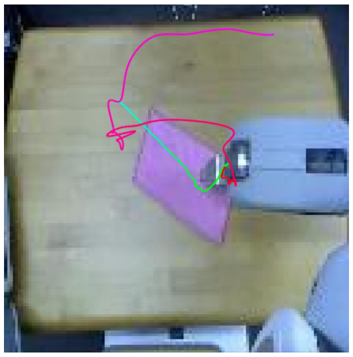

folding a cloth

Figure 4. **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.

Data Labeling Prompt with Gemini

## 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}
    ]
}
```

Now, it's your turn!

{Video} Please output the task summary and steps in the specified JSON format based on your analysis of the video.

Figure 5. **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.

Templates of 10 Question Types

## Planning Task

Template 1: The objective is <long-horizon>, what should be the next step to move forward?
Template 2: In pursuit of achieving <long-horizon>, what's the next action to take?
Template 3: To reach the goal of <long-horizon>, which task should be prioritized next?
Template 4: Given the goal of <long-horizon>, what is the most logical next move?
Template 5: With the aim of <long-horizon>, what should you focus on next?

## Planning with Context Task

Template 1: So far, you've completed these steps: 1-<task 1>, ..., n-1-<task n-1>. What's the next move to achieve the goal of <long-horizon>?
Template 2: With the following steps completed: 1-<task 1>, ..., n-1-<task n-1>. What is the next logical step toward <long-horizon>?
Template 3: Considering the goal of <long-horizon>, and having done 1-<task 1>, ..., n-1-<task n-1>, what should you do next?
Template 4: You are working towards <long-horizon>. After completing steps 1-<task 1>, ..., n-1-<task n-1>, what's the next immediate task?
Template 5: Given your progress so far (1-<task 1>, ... n-1-<task n-1>), what's the next step toward achieving <long-horizon>?

## Planning Remaining Steps Task

Template 1: With <long-horizon> as the goal and the steps 1-<task 1>, ..., n-1-<task n-1> completed, what are the next five things to do?
Template 2: To work toward <long-horizon>, what are the next five steps after completing 1-<task 1>, ..., n-1-<task n-1>?
Template 3: Here's what's been done so far: 1-<task 1>, ..., n-1-<task n-1>. What are the next five tasks to take toward the goal of <long-horizon>?
Template 4: The goal is <long-horizon>. After completing 1-<task 1>, ..., n-1-<task n-1>, what are the next five steps you should take?
Template 5: Given the progress so far: 1-<task 1>, ..., n-1-<task n-1>, what's the next set of five steps to move closer to <long-horizon>?

## Future Prediction Task

Template 1: Based on the current situation, what is expected to happen after <task n-1>?
Template 2: What do you think will happen after <task n-1> is completed?
Template 3: Considering the current sequence of tasks, what's likely to occur after <task n-1>?
Template 4: Given the context, what will most likely happen following <task n-1>?
Template 5: After <task n-1>, what's the most probable next event?

## Success (Positive/Negative) Task

Template 1: Was <task n> completed successfully?
Template 2: Has <task n> been fully carried out?
Template 3: Has <task n> reached completion?
Template 4: Was <task n> finalized?
Template 5: Can we say that <task n> was accomplished?

## Discriminative Affordance (Positive) Task

Template 1: Is <task n> something that can be accomplished right now?
Template 2: Can <task n> be initiated at this moment?
Template 3: Is it feasible to begin <task n> immediately?
Template 4: Is now a suitable time to carry out <task n>?
Template 5: Can you proceed with <task n> given the current conditions?

## Discriminative Affordance (Negative) Task

Template 1: Is <random task> what you're working on at the moment?
Template 2: Are you currently engaged in <random task>?
Template 3: Is this <random task> you're focused on right now?
Template 4: Is this <random task> you're handling at present?
Template 5: Are you doing <random task> at this very moment?

## Generative Affordance Task

Template 1: What can you do at this moment?
Template 2: Which task is possible to start right now?
Template 3: Given the current situation, what action can be taken?
Template 4: What's the next available action?
Template 5: Considering the circumstances, what task can you begin now?

## Past Description Task

Template 1: What was the last task completed?
Template 2: What just occurred?
Template 3: What was the most recent action taken?
Template 4: What task did you just finish?
Template 5: What happened immediately before this?

Figure 6. **Templates of 10 question types.** We have 10 question types for planning, each with 5 different templates to ensure the diversity of our ShareRobot dataset question formulations.

References

[1] 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. 3, 4

[2] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *CVPR*, pages 19129–19139, 2022. 2

[3] 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. 1, 5

[4] Jiajun Cao, Yuan Zhang, Tao Huang, Ming Lu, Qizhe Zhang, Ruichuan An, Ningning Ma, and Shanghang Zhang. Move-kd: Knowledge distillation for vlms with mixture of visual encoders. *arXiv preprint arXiv:2501.01709*, 2025. 7

[5] 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. 1

[6] 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. 1

[7] 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. 3

[8] 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*. 1

[9] 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. 3

[10] 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. 5

[11] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. 3

[12] Google. Gemini robotics brings ai into the physical world. *https://deepmind.google/discover/blog/gemini-robotics-brings-ai-into-the-physical-world/*, 2025. 7

[13] Peng Hao, Chaofan Zhang, Dingzhe Li, Xiaoge Cao, Xiaoshuai Hao, Shaowei Cui, and Shuo Wang. Tla: Tactile-language-action model for contact-rich manipulation. *arXiv preprint arXiv:2503.08548*, 2025. 7

[14] Xiaoshuai Hao, Yi Zhu, Srikar Appalaraju, Aston Zhang, Wanqian Zhang, Bo Li, and Mu Li. Mixgen: A new multi-modal data augmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 379–389, 2023. 1

[15] 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*. 1

[16] Yuheng Ji, Yue Liu, Zhicheng Zhang, Zhao Zhang, Yuting Zhao, Gang Zhou, Xingwei Zhang, Xinwang Liu, and Xiaolong Zheng. Advlora: Adversarial low-rank adaptation of vision-language models. *arXiv preprint arXiv:2404.13425*, 2024. 7

[17] 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. 5

[18] 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. 2, 3

[19] 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. 1

[20] 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. 5

[21] 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. 3

[22] 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. 1, 2, 5

[23] 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. 1, 2, 5

[24] Dingzhe Li, Yixiang Jin, Yuhao Sun, Hongze Yu, Jun Shi, Xiaoshuai Hao, Peng Hao, Huaping Liu, Fuchun Sun, Jianwei Zhang, et al. What foundation models can bring for robot learning in manipulation: A survey. *arXiv preprint arXiv:2404.18201*, 2024. 7

[25] 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. 1, 3, 4

[26] 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. 5

[27] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 36, 2024. 1

[28] 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, 4, 7

[29] Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, et al. Llava-plus: Learning to use tools for creating multimodal agents. In *European Conference on Computer Vision*, pages 126–142. Springer, 2024. 7

[30] Yue Liu, Xiaoxin He, Miao Xiong, Jinlan Fu, Shumin Deng, and Bryan Hooi. Flipattack: Jailbreak llms via flipping. *arXiv preprint arXiv:2410.02832*, 2024. 7

[31] 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. 3

[32] 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. 3

[33] Yue Liu, Hongcheng Gao, Shengfang Zhai, Xia Jun, Tianyi Wu, Zhiwei Xue, Yulin Chen, Kenji Kawaguchi, Jiaheng Zhang, and Bryan Hooi. Guardreasoner: Towards reasoning-based llm safeguards. *arXiv preprint arXiv:2501.18492*, 2025. 7

[34] 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. 2

[35] 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. 2

[36] 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. 3, 4, 5

[37] 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. 2

[38] 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. 3

[39] 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. 3

[40] OpenAI. Hello gpt-4o, 2024. 3, 4

[41] 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. 1

[42] 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. 1

[43] 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, 5

[44] 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. 3

[45] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 7

[46] Yingbo Tang, Shuaike Zhang, Xiaoshuai Hao, Pengwei Wang, Jianlong Wu, Zhongyuan Wang, and Shanghang Zhang. Affordgrasp: In-context affordance reasoning for open-vocabulary task-oriented grasping in clutter. *arXiv preprint arXiv:2503.00778*, 2025. 7

[47] Gemini Team, Rohan Anil, Sebastian Borgeaud, et al. Gemini: A family of highly capable multimodal models, 2024. 7

[48] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025. 7

[49] Qwen Team. Qwen2.5: A party of foundation models, 2024. 1

[50] 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. 2

[51] 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. 5

[52] 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. 2

[53] Junyang Wang, Haiyang Xu, Jiabo Ye, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. Mobile-agent: Autonomous multi-modal mobile device agent with visual perception. *arXiv preprint arXiv:2401.16158*, 2024. 7

[54] 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. 3, 4, 5

[55] x.ai. Grok 1.5v vision preview. `https://x.ai/blog/grok-1.5v`, 2024. Accessed: 2024-11-21. 3

[56] Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. *arXiv preprint arXiv:2412.14171*, 2024. 7

[57] 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. 1, 2

[58] 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. 3

[59] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, pages 11975–11986, 2023. 1

[60] Hangtao Zhang, Chenyu Zhu, Xianlong Wang, Ziqi Zhou, Changgan Yin, Minghui Li, Lulu Xue, Yichen Wang, Shengshan Hu, Aishan Liu, et al. Badrobot: Jailbreaking embodied llms in the physical world. In *The Thirteenth International Conference on Learning Representations*, . 7

[61] 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*, . 3

[62] Lingfeng Zhang, Xiaoshuai Hao, Qinwen Xu, Qiang Zhang, Xinyao Zhang, Pengwei Wang, Jing Zhang, Zhongyuan Wang, Shanghang Zhang, and Renjing Xu. Mapnav: A novel memory representation via annotated semantic maps for vlm-based vision-and-language navigation. *arXiv preprint arXiv:2502.13451*, 2025. 7

[63] 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. 3