

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

Anonymous CVPR submission

Paper ID 13171

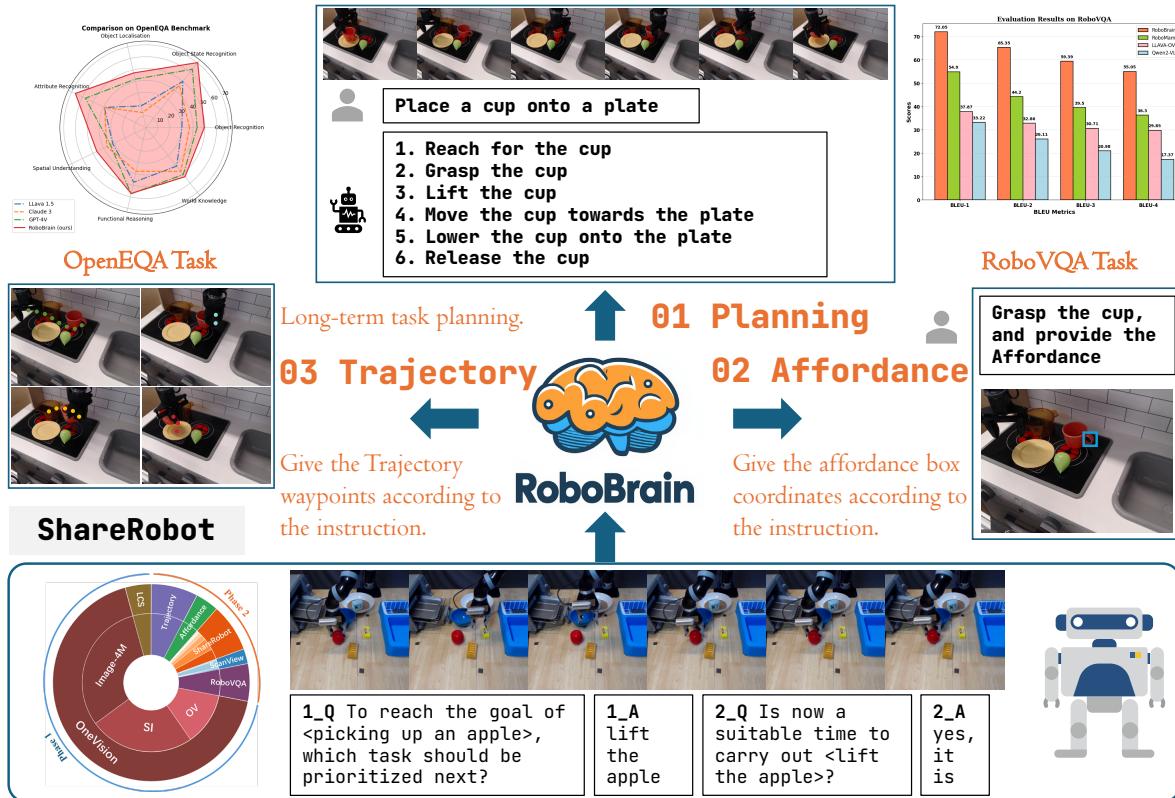


Figure 1. **Overview of RoboBrain.** RoboBrain consists of three key robotic capabilities: planning capability, affordance perception, and trajectory prediction. RoboBrain outperforms previous MLLMs in both robotics and general-purpose tasks. The bottom part shows the data composition of our ShareRobot dataset and provides a specific example of visual question answering. Best viewed in color.

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 in-

structions into manageable sub-tasks; **Affordance Perception**, 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

009
010
011
012
013
014
015
016
017
018

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.

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 [34, 36], and sentiment analysis [16]. 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 [39, 44]. 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 [38] 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 [58] and OpenEQA [47], 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 essential 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 [6–8, 29], with fewer addressing the decomposition of high-level task instructions into actionable steps. PaLM-E [18] generates multimodal inputs by mapping real-world observations into the language embedding space. RT-H [6] and Robomamba [40] 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

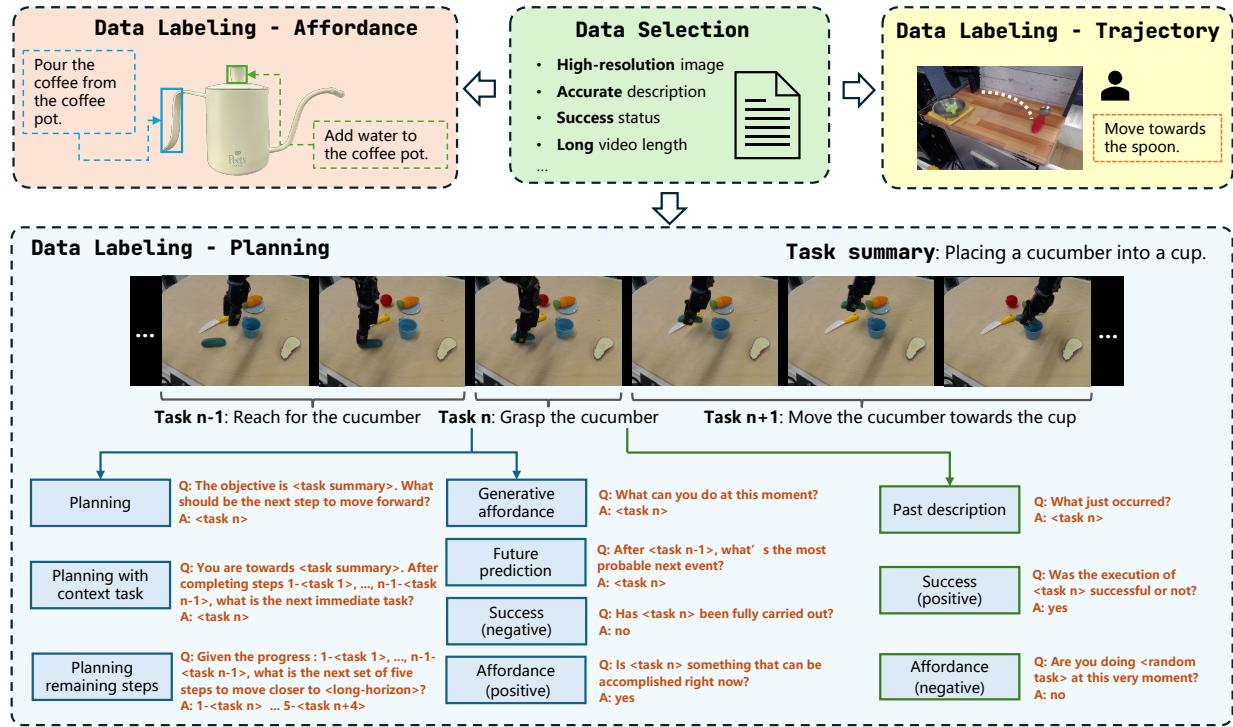


Figure 2. **The generation procession of our ShareRobot dataset.** Our dataset labels multi-dimensional information, including task planning, object affordance, and end-effector trajectories. The task planning is first annotated by atomic tasks and then augmented by constructing question-answer pairs.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

122

affordance perception and trajectory prediction.

123

Datasets for Manipulation Planning Early datasets for Manipulation [11, 22, 30, 42, 60] mainly comprise annotated images and videos that highlight fundamental hand-object interactions, including grasping and pushing. Recent advancements [17, 58] in robotic manipulation emphasize multi-modal and cross-embodiment datasets for enhanced generalization. Datasets such as RH20T [19], Bridge-DataV2 [65], and DROID [27] enhance scene diversity, broadening the range of manipulation scenarios. Notably, RT-X [52] 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.

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

3. ShareRobot Dataset

139

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.

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

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[51], 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 data from the Open-X-Embodiment dataset[51], 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

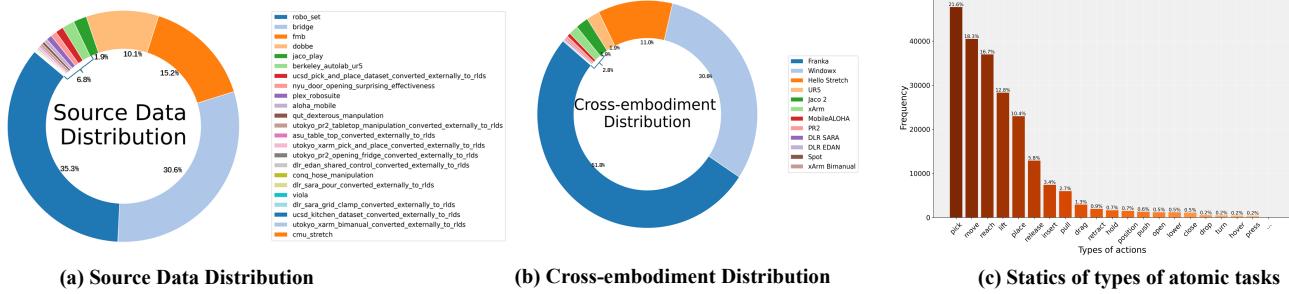


Figure 3. The diversity of our ShareRobot dataset. Our dataset involves (a) 23 original datasets, (b) 12 embodiments and (c) 107 types of atomic tasks. The distribution of the top 20 most frequent atomic actions within our ShareRobot dataset is presented in (c).

planning, affordance prediction, and trajectory prediction, enabling deeper understanding of complex relationships from abstract to concrete.

- **Rich diversity** In contrast to the RoboVQA[58] 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

184 Based on the Open X-embodiment dataset [51], we care-
185 fully selected 51,403 instances, mainly focusing on image
186 quality, description accuracy and success status. Our data
187 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 [61]. 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 [58] 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 co-ordinates and $\{r^{(x)}, r^{(y)}\}$ are the bottom right corner co-ordinates.

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.

3.4. Data Statics

We select 23 original datasets from the Open X-embodiment dataset[51]. 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,

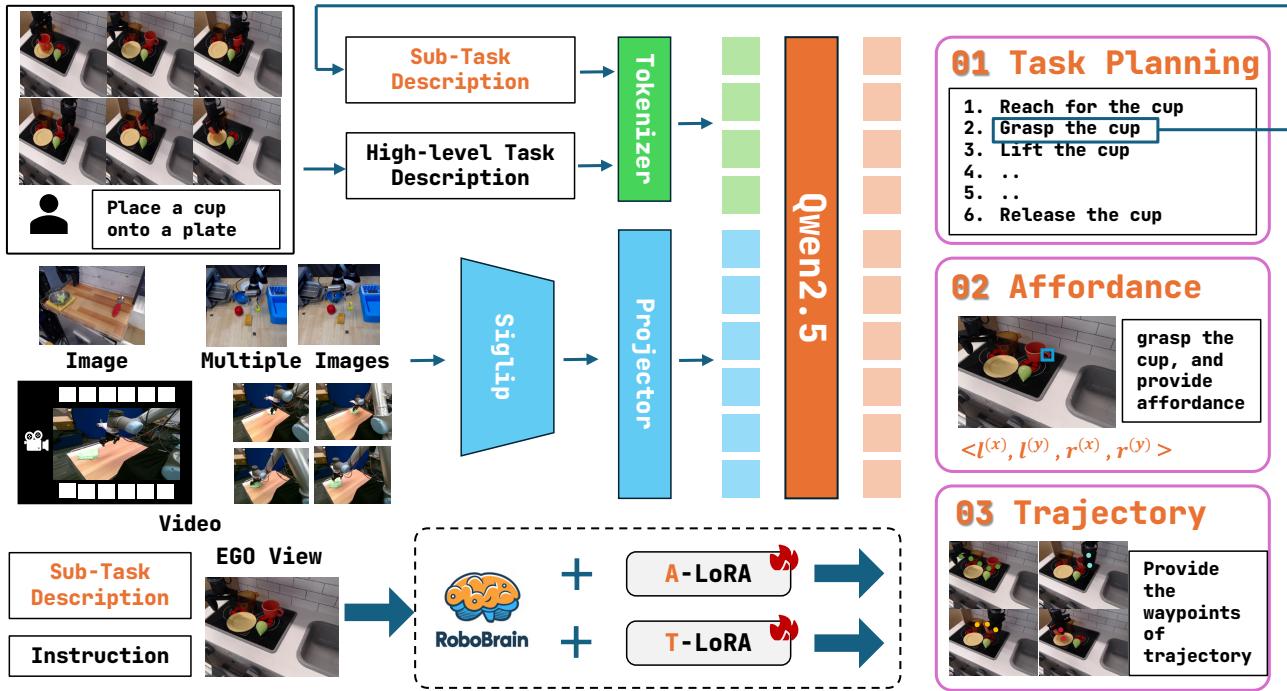


Figure 4. **The pipeline of our RoboBrain.** The images, multiple images, and videos are sent into our model to pre-train a foundation robotic brain. Besides, we fine-tune the RoboBrain via A-LoRA and T-LoRA to develop Affordance and Trajectory skills. In practical applications, the model first generates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

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.

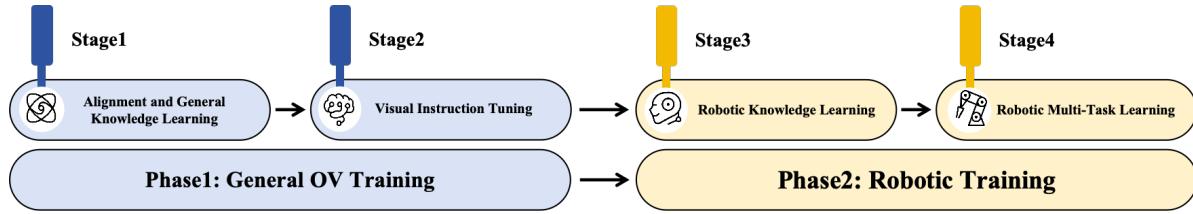
4.1. Model Architecture

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 [71], a 2-layer MLP [37], and Qwen2.5-7B-Instruct [62]. 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 .

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 affor-

244
245
246
247
248
249
250
251
252
253
254
255
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292



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

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

293 dances. Formally, consider an image I consisting of multiple objects with their affordances: $O_i = \{A_i^0, A_i^1, \dots, A_i^N\}$,
294 where the i th object owns N affordances. The format of affordance
295 is defined as $\{(l^{(x)}, l^{(y)}, r^{(x)}, r^{(y)})\}$, and $\{l^{(x)}, l^{(y)}\}$
296 represents the top left corner coordinates of affordance,
297 while $\{r^{(x)}, r^{(y)}\}$ is the bottom right corner coordinates.
298

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

308 4.2. Training

309 **Phase 1: General OV Training** In Phase 1, we drew on the
310 state-of-the-art training data and strategies from LLaVA-
311 OneVision [33] to construct a foundational model with
312 general multi-modal understanding and visual instruction fol-
313 lowing capabilities. This lays the groundwork for enhanc-
314 ing the model’s robotic manipulation planning abilities in
315 Phase 2. Detailed information is provided in Tab. 1.

316 In Stage 1, we utilize the image-text data from the LCS-
317 558K dataset [10, 57] to train Projector, facilitating the
318 alignment of visual features Z_v with the LLM semantic fea-
319 tures H_v . In Stage 1.5, we train the entire model using 4M
320 high-quality image-text data to enhance the model’s multi-
321 modal general knowledge understanding capabilities. In

322 Stage 2, we further train the entire model with 3.2M single-
323 image data and 1.6M image and video data from LLaVA-
324 OneVision-Data [33], aiming to enhance the instruction-
325 following abilities of RoboBrain and improve understand-
326 ing of high-resolution image and video.

327 **Phase 2: Robotic Training** In Phase 2, we build upon
328 the robust multi-modal foundational model developed in
329 Phase 1 to create a more powerful model for robotic manip-
330 ulation planning. Specifically, we aim for RoboBrain to un-
331 derstand complex, abstract instructions, support the percep-
332 tion of historical frame information and high-resolution im-
333 ages, and output object affordance regions while predicting
334 potential manipulation trajectories. This will facilitate the
335 transition from abstract to concrete in manipulation plan-
336 ning tasks. Detailed information is provided in Tab. 1.

337 In stage 3, we collected a dataset of 1.3M robotic data to
338 improve the model’s robotic manipulation planning capa-
339 bilities. Specifically, this data is sourced from RoboVQA-
340 800K [58], ScanView-318K including MMScan-224K [24,
341 45], 3RScan-43K [24, 64], ScanQA-25K [4, 24], SQA3d-
342 26K [24, 46], and the subset of ShareRobot-200K intro-
343 duced in this paper. These datasets contain a substantial
344 amount of scene-scanning image data, long video data, and
345 high-resolution data to support the model’s ability to per-
346 ceive diverse environments. Additionally, the fine-grained,
347 high-quality planning data within the ShareRobot dataset
348 further enhances the robotic manipulation planning capabili-
349 ties of RoboBrain. To mitigate the issue of catastrophic for-
350 getting [72], we selected a subset of high-quality image-text
351 data from Phase 1 about 1.7M to mix with the robotic data

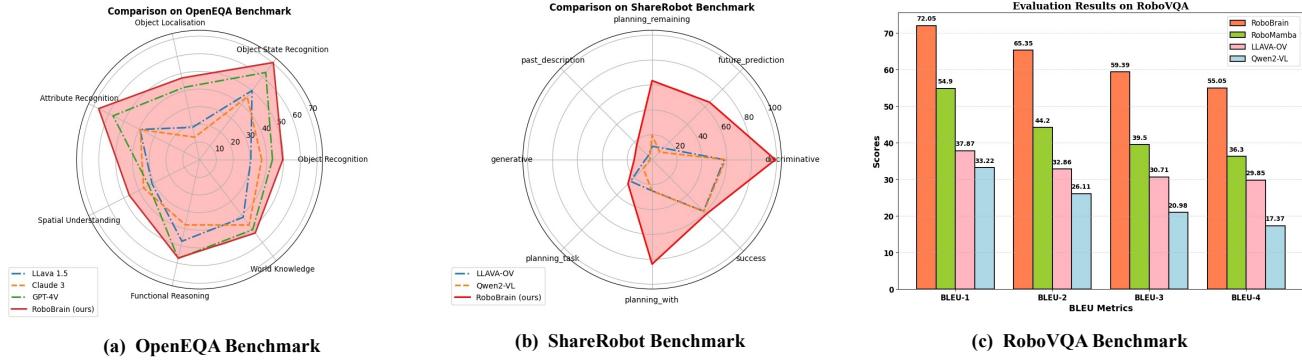


Figure 5. The performance of our model RoboBrain on the OpenEQA, ShareRobot, and RoboVQA benchmarks. RoboBrain surpassed all baseline models, achieving state-of-the-art results.

352 collected in Stage 3 for training, tuning the entire model ac-
 353 cordingly. In Stage 4, we further enhanced the model’s abil-
 354 ity to perceive object affordances based on instructions and
 355 predict manipulation trajectories, utilizing affordance and
 356 trajectory data annotated in the ShareRobot dataset. This
 357 was achieved by introducing LoRA [23] modules for train-
 358 ing to realize fine-grained planning capabilities. For the
 359 specific construction of training data and the training meth-
 360 ods employed, please refer to the supplementary materials.

DFD captures overall shape and temporal alignment, HD
 386 identifies maximum deviation, and RMSE measures aver-
 387 age pointwise error. Together, these metrics provide a com-
 388 prehensive assessment of trajectory accuracy and similarity.
 389

Affordance Prediction Here, we utilize the average pre-
 390 cision (AP) to evaluate the affordance performance of our
 391 model. AP metric summarizes the precision-recall affor-
 392 dance curve, which plots the relationship between preci-
 393 sion and recall at various threshold settings. It is calculated
 394 across multiple IoU (Intersection over Union) thresholds to
 395 obtain a more comprehensive evaluation.
 396

361 5. Experiment

362 5.1. Implementation Details

363 During the entire training phase, we employed the
 364 Zero3 [56] distributed training strategy, conducting all ex-
 365 periments on a cluster of servers, each equipped with
 366 8×A100 GPUs. The training components for each stage,
 367 including image resolution settings, batch size, epochs, and
 368 learning rates, are provided in Tab. 1.

386 DFD captures overall shape and temporal alignment, HD
 387 identifies maximum deviation, and RMSE measures aver-
 388 age pointwise error. Together, these metrics provide a com-
 389 prehensive assessment of trajectory accuracy and similarity.
 390

Evaluation on Planning Task We selected 6 powerful
 391 MLLMs as our baselines for comparison, including both
 392 open-source and closed-source models with different archi-
 393 tectures. Specifically, these models include GPT-4V [2],
 394 Claude3 [1], llava1.5 [38], LLaVA-OneVision-7b [33],
 395 Qwen2-VL-7b [66] and RoboMamba [40]. Our specific ex-
 396 perimental results are shown in Fig. 5. Our RoboBrain out-
 397 performed all baseline models across four robotic bench-
 398 marks. RoboBrain significantly outperformed all baseline
 399 models on OpenEQA and ShareRobot, which can be at-
 400 tributed to its robust capabilities in understanding robotic
 401 tasks and perceiving long videos. Additionally, this pat-
 402 tern was observed in other benchmarks as well, with Robo-
 403 Brain consistently demonstrating superior performance on
 404 RoboVQA, achieving a BLEU-4 score that exceeded that
 405 of the second-place model by 18.75. This result highlights
 406 its capability to decompose complex long-range task plan-
 407 ning. Please refer to the supplementary materials for more
 408 ablation studies due to space limitation.
 409

Evaluation on Trajectory Prediction We compare sev-
 410 eral variants of our model, and the results are in Tab. 2:
 411 (1) **Baseline**, fine-tuned on trajectory-related VQA data;
 412 (2) **Start_Points**, which adds the 2D start coordinates of
 413 the end-effector; (3) **Max_Points**, limiting waypoints to 10
 414

369 5.2. Evaluation Metrics

370 **Planning Task** We selected RoboVQA [58],
 371 OpenEQA [47], and the test set of ShareRobot ex-
 372 tracted from the proposed ShareRobot dataset as robotic
 373 benchmarks for multi-dimensional assessment. For
 374 RoboVQA, we adopt the BLEU1 to BLEU4 metrics [54]
 375 used in RoboMamba [40] for evaluation. Additionally,
 376 for OpenEQA and ShareRobot, we use GPT-4o [53] as
 377 the evaluation tool, scoring based on the alignment or
 378 similarity between model predictions and ground truth,
 379 which serves as the final performance score for the model.
 380

381 **Trajectory Prediction** We evaluate the similarity be-
 382 tween ground truth and predicted trajectories, both rep-
 383 resented as sequences of 2D waypoints normalized to
 384 [0, 1000], following Qwen2-VL [67]. The evaluation uses
 385 three metrics: Discrete Fréchet Distance (DFD) [21], Haus-
 386 dorff Distance (HD), and Root Mean Square Error (RMSE).

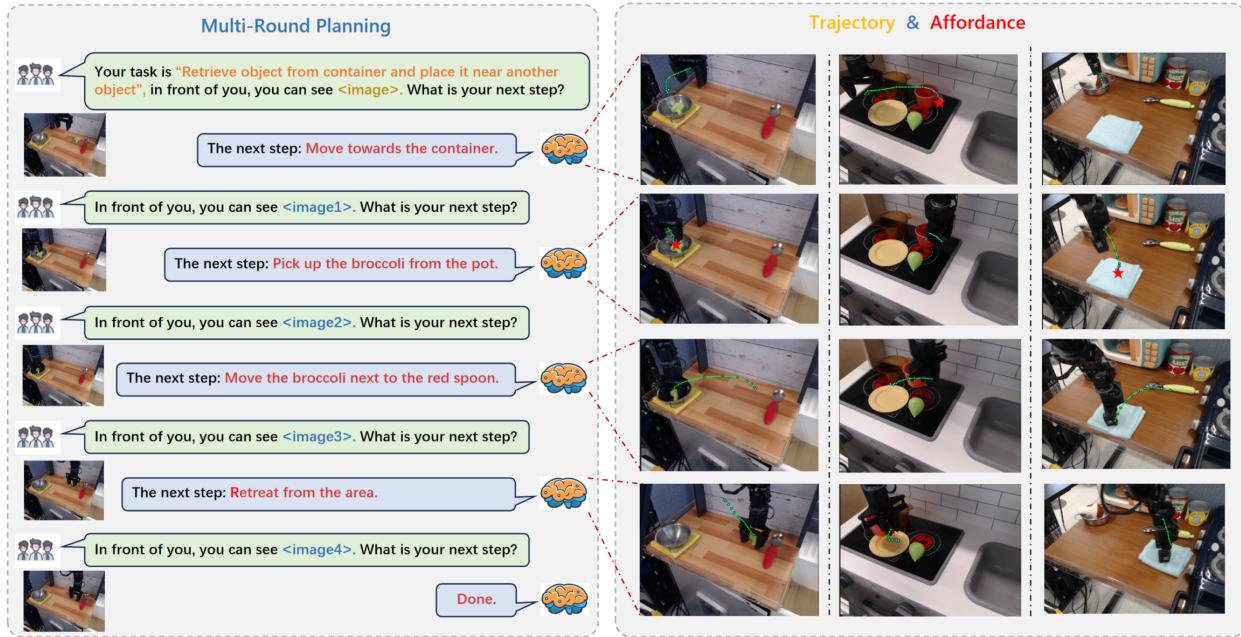


Figure 6. This visualization illustrates that RoboBrain can interpret human instructions and visual images to generate action plans and assessments based on real-time image feedback. Furthermore, it predicts trajectories for each step and identifies corresponding affordances.

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%↓)

422 via uniform sampling; and **(4) Spec-Token & End.Points**,
423 which adds end-effector positions and special tokens to em-
424 phasize waypoints and start/goal points. Each variant builds
425 on the previous one, with the final model integrating all
426 components. Our most effective model integrates all de-
427 sign choices. As shown in the last row of Tab. 2, DFD, HD,
428 and RMSE decreased by 42.9%, 94.2%, and 31.6%, res-
429 pectively, compared to the baseline. We found that adding start
430 points corrected the translational offset between the gener-
431 ated trajectory and the end-effector.

432 **Evaluation on Affordance Prediction** Our results are
433 summarized in Tab. 3. We compare the Qwen2-VL-7B
434 and LLaVA-NeXT-7B models. Qwen2-VL [66] has a su-
435 perior visual grounding ability and LLaVA-NeXT [35] owns
436 a high-resolution and strong vision tower. We test them all
437 on the AGD20K affordance test set. Our RoboBrain outper-
438 forms significantly the other models. It surpasses Qwen2-
439 VL [66] by 14.6 AP, and LLaVA-NeXT by 17.3 AP. It vali-
440 dates our RoboBrain can understand the physical properties
441 of objects and provide the affordance accurately.

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 [38]	9.8 %
Qwen2-VL-7B [5]	12.5 %
RoboBrain (Ours)	27.1 % (14.6↑)

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 such as task planning, object affordance, and end-effector trajectory. We also 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 enhance robotic manipulation. Extensive experiments demonstrate that RoboBrain achieves state-of-the-art performance in various robotic tasks, highlighting its potential to advance robotic capabilities.

442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460

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

This supplementary material provides more details of the proposed method and experiment results that are omitted from the manuscript due to the page limit. Section 7 presents additional details of the models and training strategies. Section 8 presents details of training dataset. Section 9 complements more experiment results and analysis. Section 10 shows more visualization results to prove the effectiveness of RoboBrain. Section 11 introduces more details about the construction of ShareRobot dataset. Section 12 discusses the limitations and societal impact of our work.

7. Details of Models and Training

Model Setting. RoboBrain is built upon the LLaVA [38] framework and consists of three main components: the visual encoder, projector, and large language model (LLM).

For the visual encoder, we utilized the SigLIP [71] 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 [55] 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 [35] 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 [62] 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 [33], a state-of-the-art multimodal large language model, and built upon this foundation to expand the robotic training phase. During the entire

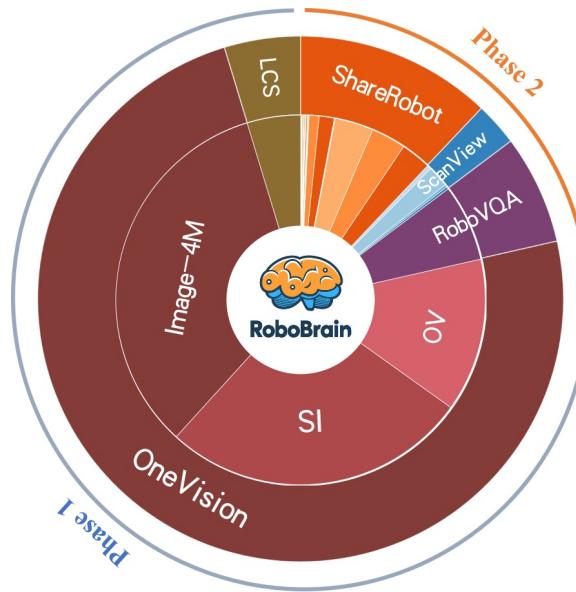


Figure 7. The distribution of the entire training dataset.

training phase, we conducted all experiments on a cluster of servers, each equipped with 8×A100 GPUs.

8. 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, 57], specifically designed as the LLaVA Visual Instruct Pretrain [38] 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 [32]: BLIP558K, COCO118K, and CC3M, as well as UReader [69], Instruct Azure DC [32], Evol-Instruct [12], and SynthDog [28]. We utilized the download links provided by the LLaVA-Onevision team for the data acquisition.
- **SI-3.2M¹** [33] consists of 3.2 million samples, carefully

¹Due 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.

Vision	Stage-1	Stage-1.5	Stage-2		Stage-3	Stage-4	
			Single-Image	OneVision		A-LoRA	T-LoRA
	Resolution	384	Max 384×{2×2}	Max 384×{6×6}	Max 384×{6×6}	Max 384×{6×6}	Max 384×{6×6}
#Tokens	729	Max 729×5	Max 729×37				
Model	Trainable	Projector	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
Per-device Batch Size	8	2	1	1	1	4	4
Gradient Accumulation	1	2	2	2	2	2	2
LR: vit	-	2×10^{-6}					
LR: {Proj.:LLM :LoRA }	1×10^{-3}	1×10^{-5}					
Epoch	1	1	1	1	1	1	1
Training	Optimizer	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						
Projector Type	mlp2x_gelu						
Vision Select Layer	-2	-2	-2	-2	-2	-2	-2
Patch Merge Type	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 [63], Cauldron [32], and UReader [69], which were subjected to cleaning and re-annotation to ensure data quality. Additionally, it incorporates single-image data from sources like AI2D [26] and OKVQA [48], alongside a newly compiled single-image collection designed to achieve a balanced and diverse dataset.

- **OV-1.6M²** [33] 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** [58] 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 inte-

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

grates data from several high-quality sources, including MMScan-224K [45], 3RScan-43K [64], ScanQA-25K [4], and SQA3D-26K [46], 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.

9. 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.

9.1. More Results on General Benchmarks

To evaluate performance on general tasks in real-world scenarios, as is commonly done with MLLMs [2, 15, 35, 53, 66], we conducted experiments using a diverse set of image benchmarks summarized in Table 5. We leveraged the comprehensive evaluation toolkit, LMMs-Eval[73], to evaluate RoboBrain’s performance on general benchmarks. These benchmarks are categorized into three classes:

Table 5. Performance comparison on multiple general benchmarks.

Dataset	Split	RoboBrain (Ours)	GPT-4V [2]	LLaVA-OV-7B [35]	InternVL2-8B [15]	Qwen2-VL-7B [66]	GPT-4o [53]
A12D[26]	test	82.03	78.2	81.4	83.8	-	94.2
ChartQA[49]	test	80.48	78.5	80	83.3	83	85.7
DocVQA[50]	test	88	88.4	87.5	91.6	94.5	92.8
TextVQA[59]	val	75.85	-	71.07	77.4	84.3	-
MMMU[70]	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
OCRBBench[41]	-	677	656	697	794	845	805
RealWorldQA[68]	test	68.89	61.4	66.3	64.4	70.1	58.6
SeedBench[31]	image	71.03	49.9	75.4	76.2	-	76.2
MMBench[43]	en-dev	81.52	81.3	83.2	-	-	83.4
MMBench[43]	en-test	80.44	75	80.8	81.7	83	-
MME[20]	test	2084	1926	1998	2210	2327	-

- **Chart, Diagram, and Document Understanding.** As key visual formats for structured OCR data, benchmarks such as AI2D [26], ChartQA [49], DocVQA [50], and OCRBBench [41] were utilized. Open-source models like InternVL2-8B [15] and LLaVA-OV-7B [35] 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 [35] and GPT-4V [2] on these benchmarks, achieving a significant improvement in structured OCR tasks, with the only exceptions being DocVQA [50], where it performs slightly lower than GPT-4V [2], and OCRBBench [41], where it falls slightly behind LLaVA-OV-7B [35].
- **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 [43], and MME [20], while reasoning benchmarks include MMMU [70] and SeedBench [31]. *RoboBrain* demonstrates comparable performance to GPT-4V [2] and LLaVA-OV-7B [35] across multiple benchmarks.
- **Real-world Understanding and Interaction.** Evaluating MLLMs [2, 15, 35, 53, 66] as general-purpose assistants in real-world settings is crucial, as these scenarios extend beyond controlled environments. For this, the RealworldQA [68] benchmark was utilized. Results indicate that *RoboBrain* not only outperforms open-source models like LLaVA-OV-7B [35] and InternVL2-8B [15], but also exceeds closed-source models such as GPT-4V [2] and GPT-4o [53], showcasing its extensive knowledge base and strong generalization capabilities.

9.2. More Results on Robotic Benchmarks.

To evaluate *RoboBrain*'s performance on robotic capabilities in real-world scenarios, we selected RoboVQA [58],

OpenEQA [47], 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 [35], and Qwen2-VL-7B [66], as well as robotic models like RoboMamba [40]. Detailed descriptions of the three selected robotic benchmarks and the analysis of each results are provided below:

- **RoboVQA** [58] 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 [40], we utilized BLEU-1~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** [47] 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 [47]. 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

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

Dataset	Split / Metric	RoboBrain (Ours)	GPT-4V [2]	LLaVA-OV-7B [35]	RoboMamba [40]	Qwen2-VL-7B [66]
RoboVQA[58]	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[47]	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

VQA pairs, drawn from 102 diverse scenes (e.g., bedroom, laboratory, kitchen, office) and covering 12 different robot bodies. Similar to RoboVQA [58], 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 [58], we utilized GPT-4o [53] 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.

9.3. 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.

9.4. 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 4:6, comprising 40% robotic data and 60% general data, **same to Exp A**.
- **Exp D** utilizes a ratio of 5:5, with 50% robotic data and 50% general data.
- **Exp E** utilizes a ratio of 6:4, featuring 60% robotic data and 40% general data.
- **Exp F** 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 benchmark 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[58]	OpenEQA[47]	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

716

10. More Qualitative Results

717
718
719
720

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.

721

10.1. Visualization on Planning

722
723
724
725
726
727
728
729
730
731

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.

732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747

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.

748
749
750
751
752

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.

10.2. 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

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791 noisy environments.

792 10.3. Visualization on Trajectory

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

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

812 The visualizations in the third row further suggest that
 813 RoboBrain avoids overfitting; it generalizes well across dif-
 814 ferent scenarios, producing trajectories that are both exe-
 815 cutable and reasonable.

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

825 11. Details of ShareRobot Dataset

826 In the previous section, we introduced the process of col-
 827 lecting and annotating our ShareRobot dataset. Here, we
 828 will provide detailed prompts for data labeling and display
 829 examples.

830 11.1. Prompts

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

833 11.2. High-level Descriptions Examples

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

- 837 • Closing a drawer
- 838 • Opening a drawer

• Opening a cabinet door	839
• Dragging a strainer across a table	840
• Picking up a bowl	841
• Inserting a three-pronged object into its matching slot	842
• Inserting a double-square object into its matching slot	843
• Opening a door	844
• Closing a cabinet door	845
• Inserting a star-shaped object into its corresponding slot	846
• Opening a laptop	847
• Inserting an oval object into its corresponding slot	848
• Picking up a ketchup bottle from a table	849
• Moving a banana from a plate to a table	850
• Closing a door	851
• Switching a light switch	852
• Inserting an arch-shaped object into its corresponding slot	853
• Inserting a square-circle object into its matching slot	854
• Dragging a strainer backwards	855
• Dragging a mug from left to right	856
• Dragging a mug forward	857
• Picking up a red object from a table	858
• Placing a ketchup bottle onto a plate	859
• Placing a bowl inside an oven	860
• Inserting a hexagonal object into its corresponding slot	861
• Closing a microwave door	862
• Moving a banana from a table to a plate	863
• Turning on a toaster	864
• Opening a microwave	865
• Closing an oven door	866
• Making tea	867
• Dragging a strainer forward	868
• Placing a bowl into an oven	869
• Picking up a banana and placing it in a mug	870
• Inserting an arch-shaped object into its matching slot	871
• Closing a tea container	872
• Inserting a green object into a designated slot	873
• Picking up a banana and placing it in a strainer	874
• Moving a cloth to the left side of a table	875
• Dragging a mug backwards	876
• Placing a bottle into a pot	877
• Dragging a strainer forwards	878
• Inserting a rectangular prism into its matching slot	879
• Opening a refrigerator	880
• Opening a tea container	881
• Opening a double door	882
• Inserting a cylinder into a matching hole	883
• Picking up a piece of toast	884
• Dragging a mug from left to right on a table	885
• Closing a refrigerator door	886

887 11.3. Low-level Instructions Examples

888 Our ShareRobot dataset contains 28,181 low-level instruc-
 889 tions. The top 50 frequency occurrences are displayed be-
 890 low.

- 891 • Grasp the ketchup bottle
892 • Reach for the ketchup bottle
893 • Grasp the banana
894 • Lift the ketchup bottle
895 • Lift the banana
896 • Reach for the strainer
897 • Reach for the banana
898 • Reach for the mug
899 • Grasp the mug
900 • Lift the pot
901 • Lift the bowl
902 • Pull the drawer open
903 • Reach for the bowl
904 • Reach for the pot
905 • Grasp the strainer
906 • Reach for the drawer handle
907 • Grasp the handle
908 • Lift the spoon
909 • Grasp the bowl
910 • Reach for the spoon
911 • Place the ketchup bottle on the table
912 • Release the banana
913 • Reach the drawer
914 • Place the banana on the table
915 • Lift the mug
916 • Reach the cabinet door
917 • Grasp the pot
918 • Grasp the strainer
919 • Grasp the drawer handle
920 • Release the mug
921 • Grasp the pot
922 • Grasp the spoon
923 • Place the mug down
924 • Move the banana towards the table
925 • Grasp the bowl
926 • Pull the drawer closed
927 • Move towards the bowl
928 • Reach for the cloth
929 • Release the pot
930 • Grasp the bottle
931 • Lift the cloth
932 • Lift the red object
933 • Grasp the banana
934 • Lift the butter
935 • Reach for the banana on the table
936 • Place the ketchup bottle on the plate
937 • Grasp the drawer handle
938 • Move the ketchup bottle towards the plate
939 • Move towards the red object
940 • Place the banana on the plate

12. Potential Limitation

Due to the high cost of manual annotation, our proposed ShareRobot includes only 1,028,060 question-answer pairs. Fine-grained, high-quality data still requires support from the community. Furthermore, due to the cost limitations of distributed training, we have not explored additional possibilities for the model itself.

941

942

943

944

945

946

947



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

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

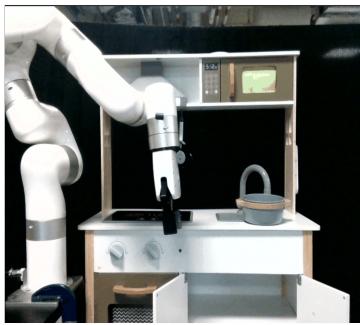
Step1: Move to the potted plant. Move towards the potted plant on the right side of the cardboard box.

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.



(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 "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.

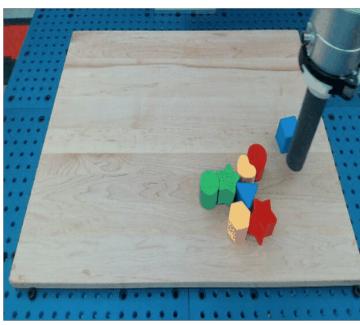
Step2: Pick up the pot. Grasp the pot with the gripper of the robotic arm and lift it from the counter. Make sure to hold the pot securely to avoid dropping it.

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.



(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 "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.



(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.10.1.

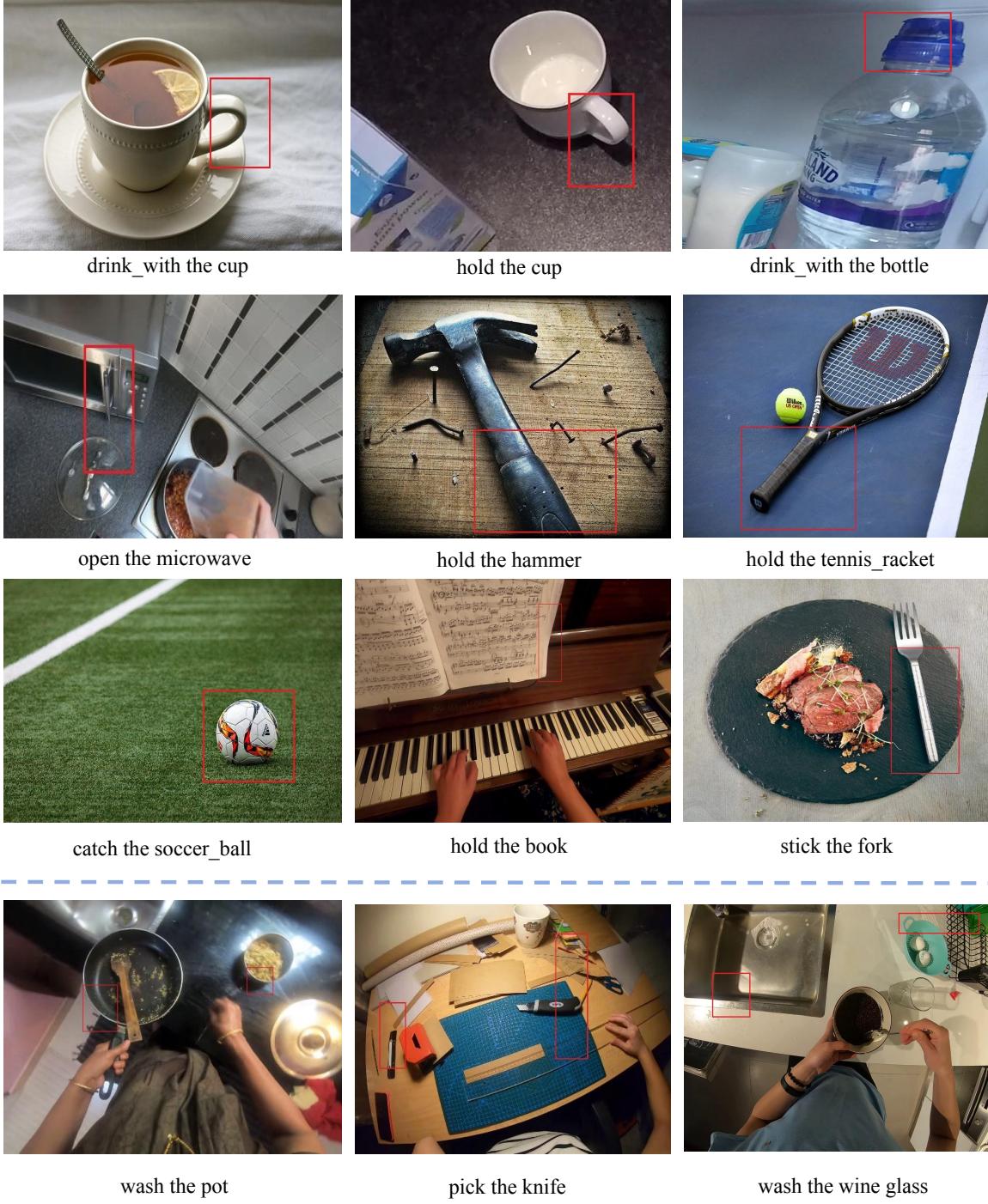


Figure 9. **Additional visualizations of diverse affordance areas.** The text below each subfigure indicates the task instructions, while the red bounding boxes represent the affordance areas predicted by the RoboBrain model. The visualizations in the first three rows demonstrate that our RoboBrain model 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."

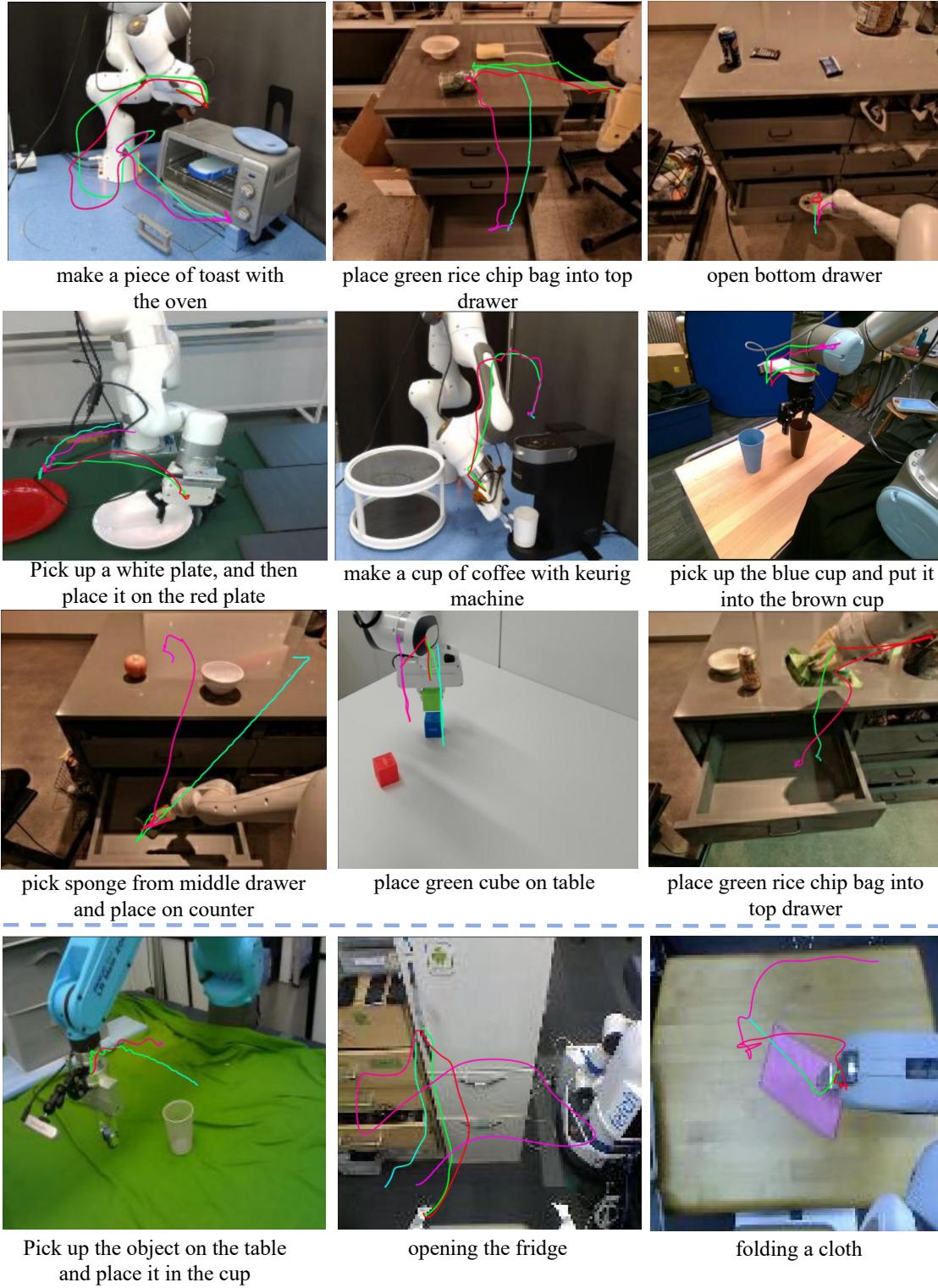


Figure 10. **Additional visualizations of diverse 2D trajectories.** The red-to-purple gradient curves represent the ground truth, while the green-to-blue gradient curves indicate the predicted trajectories. The visualizations in the first two rows demonstrate that our RoboBrain model effectively generates 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 description, 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.

948

References

949

- [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*, 2023. 7, 2, 3, 4
- [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, 2
- [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*, 2023. 8, 1
- [6] Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debiddatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. *arXiv*, 2024. 2
- [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*, 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*, 2023. 2
- [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, 1
- [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*, 2024. 1
- [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*, 2024. 3
- [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, 1

15

- Zhe Chen, Jiannan Wu, Wenhui 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, 3
- [16] 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
- [17] 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
- [18] 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*, 2023. 2
- [19] 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
- [20] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models, 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*, 2023. 6, 7
- [22] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnote: 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, 1
- [24] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puha 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] 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
- [27] 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*, 2024. 3

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

- 1061 [28] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon
1062 Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sang-
1063 doo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free
1064 document understanding transformer. In *ECCV*, 2022. 1
- 1065 [29] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao,
1066 Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Fos-
1067 ter, Grace Lam, Pannag Sanketi, et al. Openvla: An open-
1068 source vision-language-action model. *arXiv*, 2024. 2
- 1069 [30] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and
1070 Marc Pollefeys. H2O: two hands manipulating objects for
1071 first person interaction recognition. In *ICCV*, pages 10118–
1072 10128, 2021. 3
- 1073 [31] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao
1074 Ge, and Ying Shan. Seed-bench: Benchmarking multimodal
1075 llms with generative comprehension. *arXiv*, 2023. 3
- 1076 [32] Bo Li, Hao Zhang, Kaichen Zhang, Dong Guo, Yuanhan
1077 Zhang, Renrui Zhang, Feng Li, Ziwei Liu, and Chunyuan
1078 Li. Llava-next: What else influences visual instruction tun-
1079 ing beyond data, 2024. 1, 2
- 1080 [33] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng
1081 Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and
1082 Chunyuan Li. Llava-onevision: Easy visual task transfer.
1083 *arXiv*, 2024. 6, 7, 1, 2
- 1084 [34] Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming
1085 Yan, Bin Bi, Jiabo Ye, He Chen, Guohai Xu, Zheng Cao,
1086 et al. mplug: Effective and efficient vision-language learning
1087 by cross-modal skip-connections. In *EMNLP*, pages 7241–
1088 7259, 2022. 2
- 1089 [35] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li,
1090 Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave:
1091 Tackling multi-image, video, and 3d in large multimodal
1092 models. *arXiv*, 2024. 8, 1, 2, 3, 4
- 1093 [36] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi.
1094 Blip-2: Bootstrapping language-image pre-training with
1095 frozen image encoders and large language models. In *ICML*,
1096 pages 19730–19742. PMLR, 2023. 2
- 1097 [37] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee.
1098 Improved baselines with visual instruction tuning. In *CVPR*,
1099 pages 26296–26306, 2024. 5
- 1100 [38] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee.
1101 Visual instruction tuning. *NeurIPS*, 36, 2024. 2, 7, 8, 1
- 1102 [39] Jiaming Liu, Chenxuan Li, Guanqun Wang, Lily Lee,
1103 Kaichen Zhou, Sixiang Chen, Chuyan Xiong, Jiaxin Ge,
1104 Renrui Zhang, and Shanghang Zhang. Self-corrected mul-
1105 timodal large language model for end-to-end robot manipu-
1106 lation. *arXiv*, 2024. 2
- 1107 [40] Jiaming Liu, Mengzhen Liu, Zhenyu Wang, Lily Lee,
1108 Kaichen Zhou, Pengju An, Senqiao Yang, Renrui Zhang,
1109 Yandong Guo, and Shanghang Zhang. Robomamba: Mul-
1110 timodal state space model for efficient robot reasoning and
1111 manipulation. *arXiv*, 2024. 2, 7, 3, 4
- 1112 [41] Yuliang Liu, Zhang Li, Biao Yang, Chunyuan Li, Xucheng
1113 Yin, Cheng-lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench:
1114 On the hidden mystery of ocr in large multimodal models.
1115 *arXiv*, 2023. 3
- 1116 [42] Yun Liu, Haolin Yang, Xu Si, Ling Liu, Zipeng Li, Yuxiang
1117 Zhang, Yebin Liu, and Li Yi. TACO: benchmarking general-
izable bimanual tool-action-object understanding. In *CVPR*,
1118 pages 21740–21751, 2024. 3
- [43] 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
- [44] Zeyi Liu, Arpit Bahety, and Shuran Song. Reflect: Summa-
rizing robot experiences for failure explanation and correc-
tion. In *CoRL*, pages 3468–3484. PMLR, 2023. 2
- [45] 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 multi-
modal 3d scene dataset with hierarchical grounded language
annotations. *arXiv*, 2024. 6, 2
- [46] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao
Liang, Song-Chun Zhu, and Siyuan Huang. Sq3d: Situated
question answering in 3d scenes. In *ICLR*, 2023. 6, 2
- [47] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav
Putta, Sriram Yenamandra, Mikael Henaff, Sneha Sil-
wal, Paul McVay, Oleksandr Maksymets, Sergio Ar-
naud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit
Sharma, Vincent-Pierre Berges, Shiqi Zhang, Pulkit
Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan,
Franziska Meier, Chris Paxton, Alexander Sax, and Aravind
Rajeswaran. Openeqa: Embodied question answering in the
era of foundation models. In *CVPR*, pages 16488–16498,
2024. 2, 7, 3, 4, 5
- [48] 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
- [49] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and
Enamul Hoque. Chartqa: A benchmark for question answer-
ing about charts with visual and logical reasoning. *arXiv*,
2022. 3
- [50] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar.
Docvqa: A dataset for vqa on document images. In *Proceed-
ings of the IEEE/CVF winter conference on applications of
computer vision*, pages 2200–2209, 2021. 3
- [51] 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*, 2023. 3, 4
- [52] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Ab-
hishek 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
- [53] OpenAI. Hello gpt-4o, 2024. 7, 2, 3, 4
- [54] 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
- [55] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya
Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry,
Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learn-

- 1175 ing transferable visual models from natural language super-
1176 vision. In *ICML*, pages 8748–8763. PMLR, 2021. 1
- 1177 [56] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and
1178 Yuxiong He. Deepspeed: System optimizations enable train-
1179 ing deep learning models with over 100 billion parameters.
1180 In *SIGKDD*, pages 3505–3506, 2020. 7
- 1181 [57] Christoph Schuhmann, Romain Beaumont, Richard Vencu,
1182 Cade Gordon, Ross Wightman, Mehdi Cherti, Theo
1183 Coombes, Aarush Katta, Clayton Mullis, Mitchell Worts-
1184 man, et al. Laion-5b: An open large-scale dataset for train-
1185 ing next generation image-text models. *NeurIPS*, 35:25278–
1186 25294, 2022. 6, 1
- 1187 [58] Pierre Sermanet, Tianli Ding, Jeffrey Zhao, Fei Xia, De-
1188 bidatta Dwibedi, Keerthana Gopalakrishnan, Christine Chan,
1189 Gabriel Dulac-Arnold, Sharath Maddineni, Nikhil J. Joshi,
1190 Pete Florence, Wei Han, Robert Baruch, Yao Lu, Suvir Mir-
1191 chandani, Peng Xu, Pannag Sanketi, Karol Hausman, Izhak
1192 Shafran, Brian Ichter, and Yuan Cao. Robovqa: Multimodal
1193 long-horizon reasoning for robotics. In *ICRA*, pages 645–
1194 652, 2024. 2, 3, 4, 6, 7, 5
- 1195 [59] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang,
1196 Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus
1197 Rohrbach. Towards vqa models that can read. In *CVPR*,
1198 pages 8317–8326, 2019. 3
- 1199 [60] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dim-
1200 itrios Tzionas. GRAB: A dataset of whole-body human
1201 grasping of objects. In *ECCV*, pages 581–600, 2020. 3
- 1202 [61] Gemini Team, Rohan Anil, Sebastian Borgeaud, et al. Gem-
1203 ini: A family of highly capable multimodal models, 2024.
1204 4
- 1205 [62] Qwen Team. Qwen2.5: A party of foundation models, 2024.
1206 5, 1
- 1207 [63] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun
1208 Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang,
1209 Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-
1210 1: A fully open, vision-centric exploration of multimodal
1211 llms. *arXiv*, 2024. 2
- 1212 [64] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico
1213 Tombari, and Matthias Nießner. Rio: 3d object instance
1214 re-localization in changing indoor environments. In *ICCV*,
1215 pages 7658–7667, 2019. 6, 2
- 1216 [65] Homer Rich Walke, Kevin Black, Tony Z. Zhao, Quan
1217 Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre
1218 Wang He, Vivek Myers, Moo Jin Kim, Max Du, Abraham
1219 Lee, Kuan Fang, Chelsea Finn, and Sergey Levine.
1220 Bridgedata V2: A dataset for robot learning at scale. In
1221 *CoRL*, pages 1723–1736, 2023. 3
- 1222 [66] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan,
1223 Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin
1224 Ge, et al. Qwen2-vl: Enhancing vision-language model’s
1225 perception of the world at any resolution. *arXiv*, 2024. 7, 8,
1226 2, 3, 4
- 1227 [67] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan,
1228 Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin
1229 Ge, et al. Qwen2-vl: Enhancing vision-language model’s
1230 perception of the world at any resolution. *arXiv*, 2024. 7
- 1231 [68] x.ai. Grok 1.5v vision preview. <https://x.ai/blog/grok-1.5v>, 2024. Accessed: 2024-11-21. 3
- [69] Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan,
1233 Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang,
1234 et al. Ureader: Universal ocr-free visually-situated language
1235 understanding with multimodal large language model. *arXiv*,
1236 2023. 1, 2
- [70] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruqi
1238 Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming
1239 Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline
1240 multimodal understanding and reasoning benchmark for ex-
1241 pert agi. In *CVPR*, pages 9556–9567, 2024. 3
- [71] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and
1243 Lucas Beyer. Sigmoid loss for language image pre-training.
1244 In *ICCV*, pages 11975–11986, 2023. 5, 1
- [72] Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu,
1246 Yong Jae Lee, and Yi Ma. Investigating the catastrophic for-
1247 getting in multimodal large language models. In *NeurIPS*
1248 2023 Workshop on Instruction Tuning and Instruction Fol-
1249 lowing. 6
- [73] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu,
1251 Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan
1252 Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Re-
1253 ality check on the evaluation of large multimodal models,
1254 2024. *arxiv*. 2