Embodied-R1.5: Evolving Physical Intelligence via Embodied Foundation Models

Yifu Yuan[1,2*†] **Yaoting Huang**[1] **Xianze Yao**[1] **Shuoheng Zhang**[1]
Linqi Han[1] **Yutong Li**[1] **Pengyi Li**[1] **Jiangeng Sun**[1] **Wenting Jia**[1]
Yucheng Hu[1] **Yuhao Liu**[1] **Ruihao Liao**[1] **Qiyu Wu**[1] **Yuxiao Li**[1]
Zhao Zhang[1] **Zibin Dong**[1] **Fei Ni**[1] **Yan Zheng**[1] **Shuyang Gu**[2†]
Yi Ma[4,1†] **Hongyao Tang**[1†] **Han Hu**[2†] **Jianye Hao**[1,3†]

[1]Tianjin University [2]Tencent Hunyuan [3]MemoraX AI [4]Shanxi University

Project website: https://embodied-r.github.io/

Abstract

We introduce Embodied-R1.5, a unified Embodied Foundation Model (EFM) that integrates comprehensive embodied reasoning capabilities, spanning embodied cognition, task planning, correction, and pointing, within a single architecture toward general physical intelligence. Leveraging three automated data construction pipelines to significantly expand the data coverage of critical capabilities, we build a large-scale data system of over 15B tokens, and design a multi-task balanced RL recipe to alleviate heterogeneous task conflicts. We further introduce a Planner-Grounder-Corrector (PGC) closed-loop framework that enables a single model to autonomously execute and self-correct over long-horizon tasks. With only 8B parameters, Embodied-R1.5 achieves the best results among compared baselines on 16 out of 24 embodied VLM benchmarks, surpassing leading models like Gemini-Robotics-ER-1.5 and GPT-5.4. Embodied-R1.5 can be adapted into a VLA through benchmark-specific action fine-tuning without large-scale action pretraining, achieving strong results on SimplerEnv, LIBERO, and LIBERO-Plus. Separate ManiSkill experiments evaluate affordance grounding. We further conduct zero-shot real-robot experiments, validating performance in instruction following, affordance grounding, articulated object manipulation, and long-horizon complex tasks, demonstrating strong generalization to the physical world.

1 Introduction

Large language models have achieved remarkable success in the digital world, yet grounding intelligence in the physical world to realize general-purpose physical intelligence remains a central challenge [1–3]. Embodied reasoning is emerging as a critical pathway to bridge the seeing-to-doing gap [4, 5]: it requires models not only to perceive the physical world but also to reason about spatial geometry and task arrangement within it. To this end, we argue that achieving general physical intelligence calls for an Embodied Foundation Model (EFM) [1]. An EFM requires all capabilities to be grounded in physics, unifying perception, reasoning and execution within a single architecture. We organize the capabilities required by an EFM into three core dimensions: *spatial cognition and reasoning*, which endows the model with the ability to comprehend the semantic and spatial structure of the physical world; *task planning and correction*, which organizes execution logic while monitoring progress and correcting errors; and *embodied pointing and location*, which grounds high-level reasoning in coordinates and trajectories. Together, they enable a single model to perceive, act, and self-correct in a unified loop.

*Project leader.
†Corresponding authors. Contact: yuanyf@tju.edu.cn, jianye.hao@tju.edu.cn.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

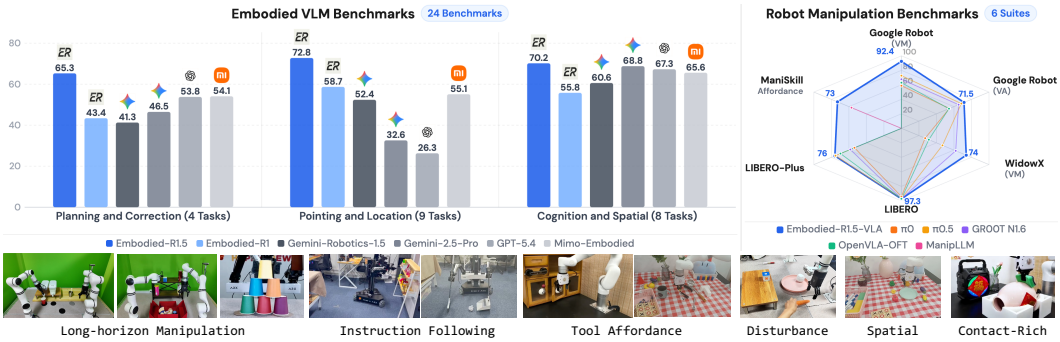


Figure 1: **Performance overview of Embodied-R1.5. Top:** Performance across 24 embodied VLM benchmarks (21 main benchmarks + 3 visual trace benchmarks) and 4 robotic manipulation benchmark suites, compared with leading general and embodied models. **Bottom:** Zero-shot real-robot experiments validating diverse embodied capabilities.

However, existing work faces three fundamental bottlenecks in realizing such a unified EFM. 1) *Fragmented capabilities.* Current embodied models each cover only part of the spectrum: some focus on cognition [6], some on planning and correction [7–9], and others on grounding [10, 4, 2]; efforts like RynnBrain [11] attempt to span multiple dimensions but rely on separate models of different scales for different tasks rather than truly unifying them within a single model. 2) *Multi-task conflict* [12]. Long text reasoning, trajectory prediction, and diverse grounding tasks differ drastically in output format; multi-task joint learning suffers from severe convergence difficulties, where different capabilities erode one another. 3) *Lack of closed-loop autonomy validation on long-horizon tasks.* Most existing EFMs remain at the level of Embodied QA, without verifying whether reasoning capabilities are truly grounded in physics under long-horizon complex decision-making.

Embodied-R1.5 serves as a comprehensive EFM that unifies all three capability dimensions, addressing the three bottlenecks above with a systematic solution. Specifically, Embodied-R1.5 jointly internalizes all three capability dimensions within a single 8B model, breaking free from the fragmented landscape. To support this capability unification, we combine existing open-source data with new data from three automated pipelines into a corpus of over 15B tokens, and design a two-stage training paradigm together with a multi-task balanced RL recipe to mitigate interference in heterogeneous joint training. Building on this, we further propose the Planner-Grounder-Corrector (PGC) closed-loop framework, where a single model simultaneously drives planning, grounding, and correction interfaces, enabling autonomous execution and recovery on long-horizon tasks such as making milk tea, sweeping garbage, and stacking cups.

With 8B parameters, Embodied-R1.5 achieves the best results among compared baselines on 16 out of 24 embodied VLM benchmarks. Its average score across the 21 main benchmarks is 70.4%, exceeding Gemini-Robotics-ER-1.5 [13] and GPT-5.4 by 17.0 and 23.2 percentage points, respectively. Through benchmark-specific action fine-tuning without large-scale action pretraining, Embodied-R1.5-VLA achieves strong results on SimplerEnv, LIBERO, and LIBERO-Plus, including 92.4% on SimplerEnv Google VM, 19.7 percentage points above $\pi_{0.5}$ [14]. Separately, affordance grounding reaches 0.66 on unseen PartNet-Mobility categories, compared with 0.54 for ManipLLM [15]. These results support transferring embodied representations to action learning. Zero-shot real-robot experiments further evaluate instruction following, affordance grounding, articulated object manipulation, and long-horizon execution.

Our contributions are as follows: (1) **Unified embodied capability system with closed-loop autonomy.** We unify all capability dimensions within a single model, and propose the PGC closed-loop execution framework that enables autonomous planning, execution, and self-correction on long-horizon complex tasks without human intervention. (2) **A complete EFM recipe.** We provide an end-to-end recipe covering data mixture, training, and EmbodiedEvalKit, a comprehensive unified EFM evaluation framework supporting 20+ tasks. (3) **Broad empirical validation.** We achieve the best results among compared baselines on 16 out of 24 embodied VLM benchmarks, strong VLA performance on three simulation suites, and effective affordance grounding on ManiSkill. Zero-shot real-robot experiments quantify task success and recovery with PGC. (4) **Open-source ecosystem.** We release training code, model weights and datasets, and EmbodiedEvalKit to support reproducible EFM research.

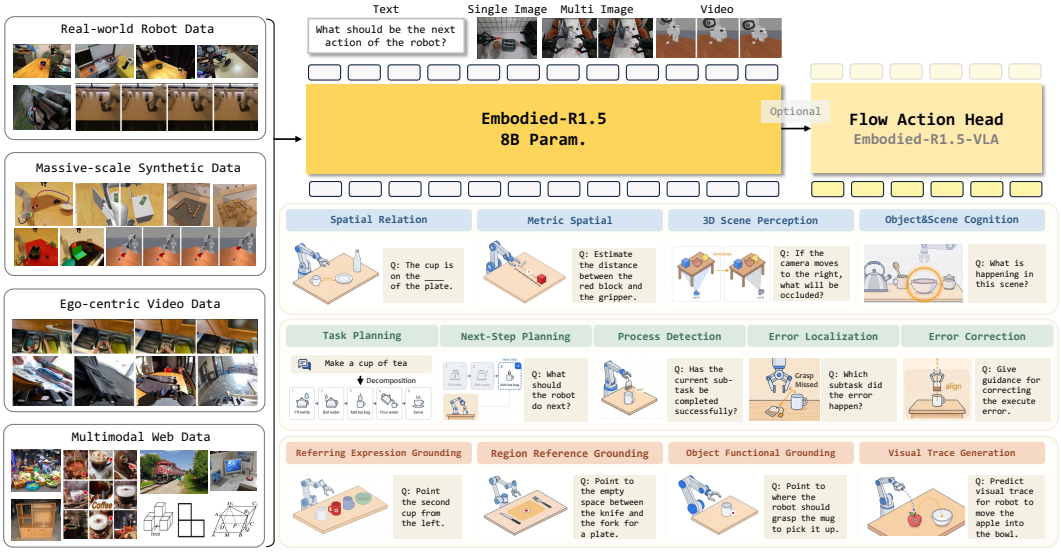


Figure 2: **Capability taxonomy and architecture of Embodied-R1.5. Top:** Embodied-R1.5 is a VLM-backbone EFM that can be further extended into a VLA by attaching a lightweight flow-matching action expert. **Bottom:** The three unified embodied capability dimensions, color-coded as Cognition & Spatial Reasoning, Planning & Correction, and Pointing & Location.

2 Unified Embodied Capabilities & Architecture

Unified Embodied Capabilities We organize the capabilities required by an EFM into three complementary dimensions (detailed definitions in Appendix B). These three dimensions form a progressive reasoning chain from perception to decision to execution, and unifying them within a single model allows information to flow freely across dimensions without external communication. Specifically: *(1) Embodied Cognition & Spatial Reasoning* endows the model with the ability to perceive and understand the physical world, covering spatial relations, metric reasoning, 3D perception, and scene cognition from robot-centric viewpoints; *(2) Embodied Planning & Correction* handles the full task execution lifecycle, from long-horizon decomposition and next-step planning to process detection, error localization, and error correction, forming a closed-loop autonomy pipeline; *(3) Embodied Pointing & Location* grounds reasoning into actionable coordinates, spanning Referring Expression Grounding (REG), Region Referring Grounding (RRG), Object Functional Grounding (OFG) for affordance, and Visual Trace Generation (VTG) that supports both object flow and end-effector flow.

Architecture As shown in Fig. 2, Embodied-R1.5 is an 8B-parameter VLM whose scale balances strong reasoning capability with practical deployment cost. All outputs are expressed as plain-text token sequences, with coordinates normalized to $[0, 1000]$, trajectories as ordered coordinate sequences, and reasoning as free-form text, enabling the model to freely interleave reasoning chains and planning steps. Furthermore, Embodied-R1.5 treats coordinates as symbols that can persist across reasoning steps, maintaining visual anchoring throughout multi-step inference. Embodied-R1.5 can be further extended into a VLA that directly outputs continuous actions, termed **Embodied-R1.5-VLA**. Since the VLM has already internalized rich embodied reasoning capabilities, the mapping from understood intent to actions is comparatively simple; we therefore attach a lightweight flow-matching action expert based on DiT [16, 17] to the VLM backbone, forming a dual-system architecture (System 2 for reasoning, System 1 for action generation) [18]. The action expert extracts vision-language features from intermediate VLM layers and generates continuous action sequences via action chunking [19]. Our hypothesis is that when the VLM has sufficiently internalized embodied reasoning capabilities, large-scale action pretraining becomes less critical; a stronger embodied backbone should substantially reduce the data requirement for downstream action learning [20, 21].

3 Training Data Construction

Building a unified Embodied Foundation Model (EFM) demands a training corpus that is both large in scale and balanced across capability dimensions. We perform large-scale integration and restructuring

of existing open-source resources, and design three automated data construction pipelines that generate proprietary data targeting critical capability gaps not covered by existing datasets. Together, existing open-source and newly constructed data span 34 datasets totaling over 15B tokens. Full data composition details and pipeline technical specifications are provided in Sec. E and Sec. F.

Pipeline 1: 3D Scene Annotation for Spatial Reasoning. We first integrate multi-view spatiotemporal reasoning data (VLM-3R [22], Cambrian-S [23], SAT [24]), depth estimation data (metric depth data curated following DepthLM [25]), and robot view cognition data (Robo2VLM [26]), covering spatial relations, distance metrics, and scene understanding. We complement these resources with additional fine-grained tabletop spatial reasoning data. Specifically, we construct the *ER1.5-Spatial dataset* ($\sim$20K samples): a fully automated pipeline reconstructs 3D semantic scene graphs from RGB images of real robot scenes (Fractal [27], BridgeData V2 [28], DROID [29]), chaining dual-backend semantic understanding, metric-scale monocular geometry estimation from MoGe-2 [30], open-vocabulary instance segmentation (Grounded-SAM [31]), and RANSAC-based horizontal plane alignment, with multi-layer quality control embedded at each stage. Spatial reasoning QA pairs are then programmatically generated from the scene graphs to form ER1.5-Spatial.

Pipeline 2: Failure-Aware Annotation for Planning and Correction. Planning data integrates RoboVQA [6], EgoPlan-IT [32], etc., covering long-horizon task decomposition and next-step planning. However, existing datasets only provide successful demonstrations and lack structured failure annotations [33]. We therefore construct the *ER1.5-Correction* dataset ($\sim$800K samples), organized along two orthogonal dimensions: by stage into planning failures and execution failures, and by cognitive level into failure detection, localization, and correction, yielding six QA types. Detection identifies whether a failure has occurred, localization identifies the erroneous step or action, and correction proposes a repair plan. These complementary annotations supervise the successive decisions needed for closed-loop recovery. For planning failures, five structured perturbation operators (step omission, redundancy, swap, object error, action replacement) are applied to correct sub-task plans, with each perturbation simultaneously generating three levels of QA. For execution failures, we combine video truncation to simulate interruption, description replacement to simulate object/action errors, and physics-engine perturbation injection in simulation to simulate manipulation failures. Data sources span real-world (BridgeData V2 [28], RoboFail [34], RoboFAC [35]) and simulated (ManiSkill [36], GEMBench [37]) scenarios. All types undergo perturbation validity verification, positive/negative balancing, and human sampling review.

Pipeline 3: Affordance and Trajectory Data for Pointing. Pointing is the capability dimension where Embodied-R1.5 exhibits the largest advantage over existing models. We construct the *ER1.5-Pointing* dataset, substantially expanding data coverage in both functional affordance and trace generation through automated pipelines. For affordance, we synthesize large-scale object functional part grounding data from ManiSkill-PartNet [38] and PRISM [39], and introduce RegularRearrangement to supplement regular arrangement capabilities. These examples contain geometric patterns with one or two missing elements, requiring the model to predict placement locations that complete the pattern and thereby connect spatial reasoning with pointing. For trajectories, to simultaneously support end-effector trace and object trace, we design an automated extraction pipeline. EEF trace follows two paths depending on whether the dataset provides pose metadata: for datasets with 3D end-effector poses (e.g., DROID), poses are directly projected onto the 2D image plane; for datasets without metadata, a Detectron2 [40]-based end-effector detector fine-tuned on OXE data tracks gripper motion. Object trace leverages models such as Co-Tracker3 [41] to track manipulated object motion across frames. Additionally, we utilize the rich assets provided by RoboCasa [42] to generate 3D trace data for articulated object manipulation. Data sources extend beyond real-robot datasets to egocentric data and large-scale simulation data. Notably, the interaction motion semantics required by trace data do not demand high-fidelity simulation; low-fidelity simulated data still achieves strong generalization (we term this *semantic generalization*). After multiple rounds of quality filtering and coordinate normalization, the final ER1.5-Pointing dataset comprises $\sim$400K samples covering diverse affordance and trajectory annotations.

Finally, we balance the overall data mixture across all capability categories and incorporate substantial general visual cognition, logical reasoning, and instruction-following data [43–46] as regularization to prevent catastrophic forgetting of general visual understanding.

4 Training Strategy

Embodied-R1.5 adopts a two-stage paradigm: Supervised Fine-Tuning (SFT) builds foundational capability, and Reinforced Fine-Tuning (RFT) refines it via RL with verifiable rewards, particularly boosting pointing. We fine-tune on the full data system for one epoch, reserving low-resource but high-value tasks for the RL stage where verifiable rewards provide more efficient signals.

Starting from GRPO [47], we address two challenges in heterogeneous multi-task RFT [12]: (1) *Intra-task*—uninformative rollout groups and group-level standard-deviation normalization can distort the relative contribution of training examples; (2) *Inter-task*—reward distributions and sample counts differ across capabilities. Our recipe combines data filtering, shared normalization, and task-appropriate rewards:

(1) Difficulty-aware data filtering. We filter the SFT corpus via rollout pass rates from the SFT checkpoint, retaining ∼200K medium-difficulty samples with informative reward variation.

(2) Dynamic filtering. When all rollouts in a group receive identical rewards, the group-centered reward advantage is zero. We automatically detect and mask such degenerate groups during training, retaining groups with discriminative reward variation for reward-driven updates.

(3) Global batch reward normalization. Standard GRPO divides group-centered rewards by each query group's standard deviation, making update weights sensitive to within-group variance. EMA-GRPO [12] introduces per-task moving averages, but low-resource task statistics become unstable when data volumes differ substantially. We instead compute advantages as $\hat{A}_i = (R_i - \mu_{\text{group}})/(\sigma_{\text{batch}} + \epsilon)$, where μ_{group} is the within-group mean and σ_{batch} is the std over the entire mixed batch. Group centering preserves within-query comparisons, while the shared denominator avoids amplifying low-variance groups independently. This requires neither task labels nor historical state, but does not itself equalize task frequencies or reward scales.

(4) Multi-task reward design. We design five reward families matching output structure: exact matching, region IoU, point localization, trajectory error, and semantic scoring. Distance-based localization and trajectory rewards use piecewise-linear decay for partial credit; all tasks additionally receive a format reward.

(5) Adaptive thinking. We only constrain the output format (requiring answers within `<answer></answer>` tags) without forcing explicit reasoning chains. We observe task-dependent output patterns: pointing often produces coordinates directly, whereas complex planning produces structured intermediate reasoning. This permits different reasoning lengths without prescribing a fixed chain for every task; we do not evaluate inference latency here.

Full reward definitions and training configurations are provided in Secs. C and D.

5 Closed-Loop Autonomy Framework

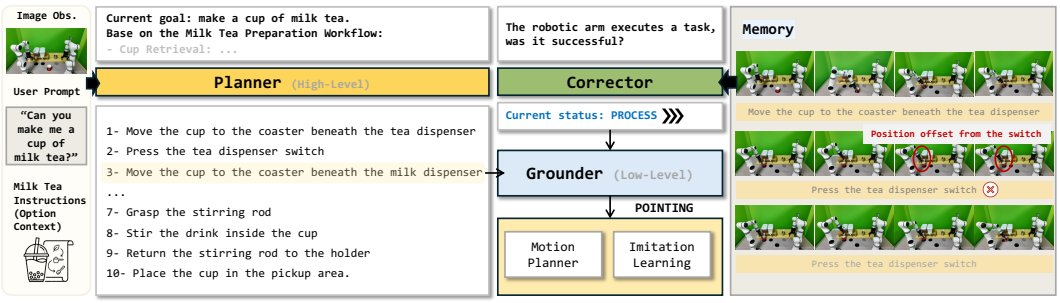


Figure 3: **Planner-Grounder-Corrector (PGC) closed-loop framework.** Embodied-R1.5 model asynchronously serves all three roles, maintaining a minimal context buffer for adaptive replanning.

To validate whether the three capability dimensions can truly ground in physics under long-horizon execution, we propose the minimalist **Planner-Grounder-Corrector (PGC)** closed-loop framework (Fig. 3). The core design principle is to deploy a single Embodied-R1.5 model as a unified inference service and asynchronously invoke its different capability interfaces to drive the full autonomy stack,

requiring no multi-model cascading or multi-agent orchestration. Given a user instruction and the current observation, the *Planner* invokes the planning capability to decompose the long-horizon task into a structured sub-task sequence; upon completion of each sub-task, it performs next-step planning conditioned on the updated observation and history buffer to advance execution. The *Grounder* invokes the pointing capability for the current sub-task, translating high-level reasoning into executable spatial commands that are then passed to a low-level executor, which can be a motion planning pipeline or a pointing-guided imitation learning policy [48]. The *Corrector* asynchronously polls the execution state: if the current sub-task is judged complete, it signals the Planner to advance; if execution has failed, it localizes the failure cause and generates corrective suggestions. The entire interaction maintains a minimal context memory that records the sub-task sequence, execution status, and historical feedback; when the Corrector detects an error, the error information is appended to the buffer so that the Planner incorporates it in the next planning round, enabling adaptive replanning from failure experience. The shared context buffer makes planning history and execution feedback available across role invocations. This framework supports autonomous execution of long-horizon tasks, with success and recovery rates quantified in Tab. 9.

6 Experiments

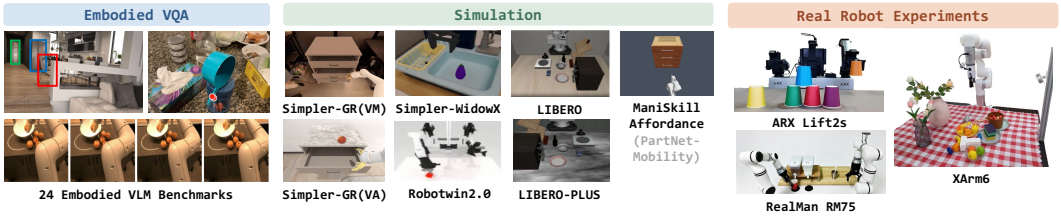


Figure 4: **Overview of the experimental evaluation of Embodied-R1.5.** Evaluation covers embodied VLM benchmarks, manipulation benchmarks, zero-shot real-world manipulation, long-horizon closed-loop execution, and ablation analysis. RoboTwin results are provided in Sec. K.

Fig. 4 summarizes the evaluation settings. Our experiments are organized into three parts: (1) comprehensive EFM capability evaluation (§6.1); (2) four robot manipulation benchmark suites, zero-shot real-robot experiments, and long-horizon closed-loop demonstrations (§6.2); (3) ablation and analysis for core research questions (§6.3).

6.1 EFM Capability Evaluation

Setup. All evaluations are conducted using EmbodiedEvalKit (detailed in Sec. H). We evaluate on 21 embodied VLM benchmarks spanning three capability dimensions (Tab. 1) and 3 extra visual trace benchmarks (Tab. 3). We additionally verify general vision capability retention on 7 standard benchmarks (Tab. 2). To prevent data contamination, we perform rigorous deduplication between our training corpus and all evaluation benchmarks at both the image and text levels; all reported results use official held-out test splits. We compare against: (a) leading general-purpose VLMs (GPT-5.4, Gemini-2.5-Pro, and Qwen3-VL-8B); (b) embodied VLMs (Gemini-Robotics-ER-1.5, RoboBrain2-7B [8], Mimo-Embodied [49], Pelican-VL [50], and Embodied-R1 [2]). Extended comparisons covering additional models and the SFT-only ablation are provided in Sec. I.1.

Results. Tab. 1 and Tab. 3 present the results. Embodied-R1.5 achieves the best results among the compared baselines on 16 out of 24 benchmarks. *Cognition & Spatial Reasoning*: Embodied-R1.5 ranks first among embodied models in overall performance and achieves the best results on RoboSpatial (69.7%) and VSIBench (56.1%), validating its spatial reasoning capability. *Pointing & Location*: This dimension exhibits the largest advantage, with Embodied-R1.5 leading on 9 out of 9 benchmarks. Part-Affordance (82.9%) validates functional part grounding, VAbench-P (73.7%) demonstrates strong affordance prediction, and RoboRefit (88.4%) confirms precise spatial referring. On visual trace (Tab. 3), Embodied-R1.5 also achieves best results on VABench-V and PIO-S3-Verified, demonstrating strong trajectory prediction. *Planning & Correction*: RoboFAC (77.2%) directly validates error correction, while Cosmos (69.3%) confirms physical reasoning. The category averages exceed Gemini-Robotics-ER-1.5 by 20.4 percentage points (pp) on Pointing and 24.0pp on Planning.

Table 1: **Embodied VLM benchmark results.** We compare against leading general-purpose VLMs and embodied VLMs. **Bold**: best result; underline: second best. †Rel. Depth Subset; ‡S1&S2 Subset; § All Split. Extended comparisons with more models are in Sec. I.1.

Benchmark	*General-Purpose VLMs*			*Embodied VLMs*					*Ours*
	GPT 5.4	Gemini 2.5-Pro	Qwen3-VL 8B	Gemini-Rob. ER-1.5	RoboBrain 2.0-7B	Mimo Embodied	Pelican-VL 1.0	Embodied R1	**Embodied R1.5**
Embodied Cognition & Spatial Reasoning									
ERQA [1]	50.5	**55.7**	41.8	48.5	38.5	46.8	39.8	35.2	46.0
OpenEQA [51]	**69.3**	65.8	65.1	50.5	24.2	53.6	54.7	34.5	62.6
CV-Bench [52]	76.7	84.6	86.3	83.6	85.8	**88.8**	78.9	82.7	86.9
EmbSpatial [53]	73.2	**78.7**	78.5	73.4	76.3	76.2	73.2	67.4	78.1
SAT [54]	74.7	**76.7**	65.3	62.7	75.3	54.6	54.6	76.3	74.7
RoboSpatial [55]	53.1	59.9	65.4	40.1	54.2	61.8	57.5	47.4	**69.7**
BLINK† [56]	87.9	85.5	78.2	85.9	79.0	**94.4**	78.2	76.6	87.1
VSIBench [57]	52.6	43.4	55.1	39.9	36.1	48.5	52.8	26.6	**56.1**
Embodied Pointing & Location									
VAbench-P [4]	25.7	21.7	42.3	41.6	41.0	46.9	14.5	66.0	**73.7**
Where2Place [10]	34.7	49.6	64.0	48.3	63.6	63.6	57.8	69.5	**74.0**
RefSpatial§ [58]	15.7	36.5	42.2	39.7	32.5	48.0	37.5	39.7	**54.2**
Part-Afford [2]	17.2	25.5	48.7	27.0	45.6	65.5	28.8	56.6	**82.9**
RoboRefit [59]	29.2	38.4	83.6	80.0	70.4	82.3	74.9	85.6	**88.4**
RoboAfford [60]	29.4	23.4	69.9	57.6	51.5	69.8	63.4	67.2	**80.0**
PointBench [61]	37.9	62.8	65.6	70.7	60.4	50.2	36.4	49.7	**74.6**
PIO‡ [62]	19.3	23.2	56.8	54.0	54.0	27.5	32.9	44.4	**62.6**
Pixmo-Point [63]	27.5	12.7	59.9	52.5	54.7	42.4	26.1	49.4	**64.8**
Embodied Planning & Correction									
RoboVQA [6]	31.8	33.9	55.0	32.6	57.5	**62.0**	58.5	51.8	61.0
EgoPlan-2 [64]	44.1	42.9	32.3	32.0	33.2	43.0	32.0	26.5	**53.8**
Cosmos [65]	69.2	62.8	64.0	53.7	33.8	56.8	62.6	55.2	**69.3**
RoboFAC [35]	70.1	46.4	65.4	47.0	54.2	54.6	60.2	40.2	**77.2**

Table 2: **General vision benchmark results.** Comparison with the base model Qwen3-VL-8B; higher is better.

Model	AI2D	SEED	POPE	MMMU	RealWQA	ChartQA	SciQA
Qwen3-VL-8B	**85.7**	**77.6**	88.1	52.0	**71.3**	**83.0**	92.3
Embodied-R1.5	83.3	76.5	**88.2**	**54.8**	69.4	**83.0**	**93.1**

RL gain and general capability retention. As shown in Sec. I.1, comparing with the SFT-only variant, RFT yields consistent gains: +3.8pp on Pointing, +2.7pp on Planning, +1.2pp on Cognition, with the largest improvement on Pointing where verifiable geometric rewards provide the strongest signal. On 7 standard vision benchmarks (Tab. 2), Embodied-R1.5 remains competitive with Qwen3-VL-8B: three scores improve, one is unchanged, and three decline. Gains on MMMU (+2.8pp) and ScienceQA (+0.8pp) coexist with decreases on AI2D (-2.4pp), SEED (-1.1pp), and RealWorldQA (-1.9pp).

6.2 Robot Manipulation Validation

Setup. For simulation, we build *Embodied-R1.5-VLA* by attaching a lightweight flow-matching action expert to the Embodied-R1.5 backbone and fine-tuning on benchmark data directly, examining whether a stronger embodied backbone can reduce the need for large-scale action data. We compare against leading VLAs including $\pi_0/\pi_{0.5}$ [66, 14], OpenVLA-OFT [67], GR00T-N1.5/N1.6 [68, 69], SpatialVLA [70], and Diffusion Policy [71]. For real-robot experiments, Embodied-R1.5 drives zero-shot task execution through pointing combined with motion planning.

SimplerEnv. SimplerEnv [75] evaluates manipulation with Google Robot and WidowX under visual matching (VM) and variant aggregation (VA) settings. We fine-tune on BridgeData and Fractal demonstrations (Sec. C.3). Baseline scores follow the corresponding reports, with Google scores for $\pi_{0.5}$ and GR00T-N1.6 from starVLA [76]; OpenVLA-OFT Google averages cover three reported

Table 3: **Visual trace results.** ↓: lower is better; Scores are scaled to 0–100. PIO-S3-Verified uses the revised LLM scoring protocol.

	ShareRobot-V		VABench-V		PIO-S3-Verified
Model	RMSE↓	DFD↓	RMSE↓	DFD↓	Score↑
GPT-5.4	23.81	33.55	18.20	26.50	44.90
Qwen3-VL-8B	22.58	32.47	15.30	21.80	46.60
RoboBrain 2.0	**13.23**	**19.63**	18.09	26.68	46.90
Embodied-R1	23.20	32.90	9.04	12.21	46.90
Embodied-R1.5-SFT	20.00	29.00	11.40	16.90	46.70
Embodied-R1.5	15.79	22.64	**7.00**	**9.83**	**49.56**

Table 4: **SimplerEnv results.** Reported success rate (%). Overall averages the three settings; † averages three reported tasks. See text for sources.

Method	**WidowX**	**Google VA**	**Google VM**	**Overall**
SpatialVLA	42.7	52.2	55.3	50.1
OpenVLA-OFT	41.8	45.5†	63.0†	50.1
π_0	27.1	54.7	71.4	51.1
π_0-FAST	32.1	44.3	46.4	40.9
$\pi_{0.5}$	57.1	68.4	72.7	66.1
GR00T-N1.5	62.0	44.5	35.2	47.2
GR00T-N1.6	57.1	65.3	67.7	63.4
Embodied-R1.5-VLA	**74.0**	**71.5**	**92.4**	**79.3**

Table 5: **LIBERO-Plus results.** Success rate (%) under distribution shifts.

Method	**Cam.**	**Robot**	**Lang.**	**Light**	**Bkgd.**	**Noise**	**Layout**	**Total**
OpenVLA	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
π_0	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
π_0-FAST	**65.1**	21.6	61.0	73.2	73.2	74.4	68.8	61.6
OpenVLA-OFT	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
ER1.5-VLA	53.2	**54.1**	**88.0**	**94.9**	**93.9**	**80.6**	**78.4**	**76.0**

Table 6: **ManiSkill Affordance.** Avg. success rate across object categories.

Method	**Seen (20)**	**Unseen (10)**
Where2Act [72]	0.26	0.21
Implicit3D [73]	0.46	0.41
ManipLLM [15]	0.59	0.54
ER1.5	**0.76**	**0.66**

tasks. In Tab. 4, Embodied-R1.5-VLA achieves 79.3% across the three settings, compared with 66.1% for $\pi_{0.5}$ (+13.2pp), including 92.4% on Google VM (+19.7pp).

LIBERO [77]. As shown in Tab. 7, Embodied-R1.5-VLA achieves 97.25% without action pretraining, exceeding $\pi_{0.5}$ without action pretraining by 3.70pp and performing comparably to action-pretrained OpenVLA-OFT (97.10%). This supports transferring embodied backbone capabilities to manipulation using benchmark-specific action fine-tuning.

LIBERO-Plus. LIBERO-Plus [78] (Tab. 5) evaluates robustness under seven distribution shifts. Embodied-R1.5-VLA achieves 76.0% total, outperforming OpenVLA-OFT (69.6%) by 6.4pp. Notably, on the Robot shift (54.1% vs. 31.9%) and Language shift (88.0% vs. 79.5%), the advantage is most prominent, suggesting that internalized reasoning provides stronger generalization under embodiment and instruction variations.

Table 7: **LIBERO results.** Success rate (%). **Pt.** denotes action pretraining. Bold/underline mark the best/second-best score in each column. Baseline results are from [74].

Method	**Pt.**	**Goal**	**Spatial**	**Object**	**Long**	**Overall**
With Action Pretraining						
OpenVLA	Y	79.2	84.7	88.4	53.7	76.50
π_0	Y	95.8	96.8	98.8	85.2	94.15
π_0-FAST	Y	88.6	96.4	96.8	60.2	85.50
$\pi_{0.5}$	Y	**98.0**	**98.8**	98.2	92.4	96.85
GR00T-N1.6	Y	97.5	97.7	98.5	94.4	96.99
OpenVLA-OFT	Y	97.9	97.6	98.4	**94.5**	97.10
Without Action Pretraining						
Diffusion Policy	N	68.3	78.3	92.5	50.5	72.40
π_0-FAST	N	89.0	87.0	63.0	48.0	71.75
OpenVLA-OFT	N	91.7	94.3	95.2	86.5	91.93
$\pi_{0.5}$	N	94.6	96.6	97.2	85.8	93.55
Embodied-R1.5-VLA	**N**	97.6	98.3	**99.2**	93.9	**97.25**

ManiSkill Affordance. We validate affordance capability on ManiSkill PartNet-Mobility [15] (Tab. 6). As a generalist EFM, Embodied-R1.5 achieves 0.76 on 20 seen categories and 0.66 on 10 unseen, surpassing the compared specialist affordance models, including ManipLLM (0.59/0.54). Full per-category results are in Sec. I.2.

Zero-shot real-robot experiments. Following the evaluation protocol of Embodied-R1 [2], we design five task categories to probe different EFM capabilities (Tab. 8): *Pick&Place* tests general grasping; *Tool Affordance* evaluates functional part recognition; *Spatial Reasoning* verifies spatial relation understanding; *Cup Disassembly* requires accurate task decomposition with per-step pointing; and *Door Open* demands articulated object manipulation. Detailed task specifications are provided in the supplementary material. Embodied-R1.5 achieves the best performance across all categories. Notably, on Cup Disassembly (66.6% vs. 0% for all baselines), Embodied-R1.5 successfully decomposes the

Table 8: **Zero-shot real-robot manipulation results.** Success rate (%) across five task categories on Xarm6 and ARX Lift2s (n=6 trials per task). Embodied-R1.5 uses PGC without task-specific fine-tuning.

Method	Pick&Place	Tool Affordance	Spatial Reasoning	Cup Disassembly	Door Open
Embodied-R1	83.3	33.3	50.0	0.0	0.0
RoboBrain 2.0	**100.0**	33.3	66.6	0.0	0.0
Embodied-R1.5	**100.0**	**100.0**	**83.3**	**66.6**	**16.6**

Table 9: **Zero-shot PGC evaluation and Corrector ablation.** Each variant is evaluated over 10 trials per task. Recovery statistics refer to full PGC and count events, rather than successful task trials.

	Task success		Recovery events (full PGC)		
Task	Full PGC	w/o Corrector	Deviations	Recovered	Rate
Sweep garbage	9/10	1/10	9	8	88.9%
Stack cups	5/10	4/10	6	4	66.7%
Make milk tea	3/10	1/10	9	5	55.6%
Total	**17/30**	**5/30**	**24**	**17**	**70.8%**

task into sub-steps and provides accurate pointing, while Door Open (16.6%) demonstrates emerging 3D trace generation capability for articulated objects.

Long-horizon closed-loop evaluation. We evaluate PGC on sweeping garbage, stacking cups, and making milk tea, with 10 trials per task for each variant. All tasks share the same Embodied-R1.5 model, PGC harness, and low-level execution interface, without task-specific fine-tuning or separate policies. We remove only the Corrector for the ablation. In addition to full-task success, we record recoverable deviation events (e.g., object displacement, inaccurate grasping, incomplete sweeping, or failed stacking). A recovery succeeds when correction returns execution to a feasible task-completion trajectory. The event-level recovery rate is the number of successful recoveries divided by the number of recoverable deviations; it is distinct from full-task success.

As shown in Tab. 9, full PGC achieves 17/30 task success (56.7%), compared with 5/30 (16.7%) without the Corrector. The largest gain occurs in sweeping, where repeated trajectory adjustments improve success from 1/10 to 9/10. Full PGC recovers from 17 of 24 recoverable deviations (70.8%). Stacking and milk-tea preparation remain difficult: detecting an error does not ensure physical recovery when grasp precision, object pose, or execution order constrains subsequent actions. These results support the contribution of correction in the evaluated tasks, while highlighting the remaining limitations of long-horizon execution.

Fig. 5 illustrates execution on these tasks and goods picking from shelves. Additional zero-shot RoboTwin results and qualitative recovery under human perturbation are provided in Secs. K and L, respectively.

6.3 Analysis

Does Embodied-R1.5 serve as a better VLM backbone for VLA? We compare Embodied-R1.5 and Qwen3-VL-8B as backbones with two action experts (GR00T and OFT) across training (Fig. 6). Embodied-R1.5 consistently outperforms, with the advantage most pronounced at early training (10K steps: 92.3% vs. 81.8%, +10.5pp) and persisting at convergence, supporting faster adaptation with the embodied backbone.

Which RL recipe works best? We ablate four RFT configurations (Fig. 7): vanilla GRPO, EMA-GRPO, DAPO without difficulty filtering, and the full recipe. The full recipe achieves scores of 70.2/65.3/72.8 on Cognition/Planning/Pointing and improves all three over the variant without difficulty filtering. It trades a small decrease in Cognition relative to vanilla GRPO (70.4) for stronger Planning and Pointing, rather than dominating every dimension.

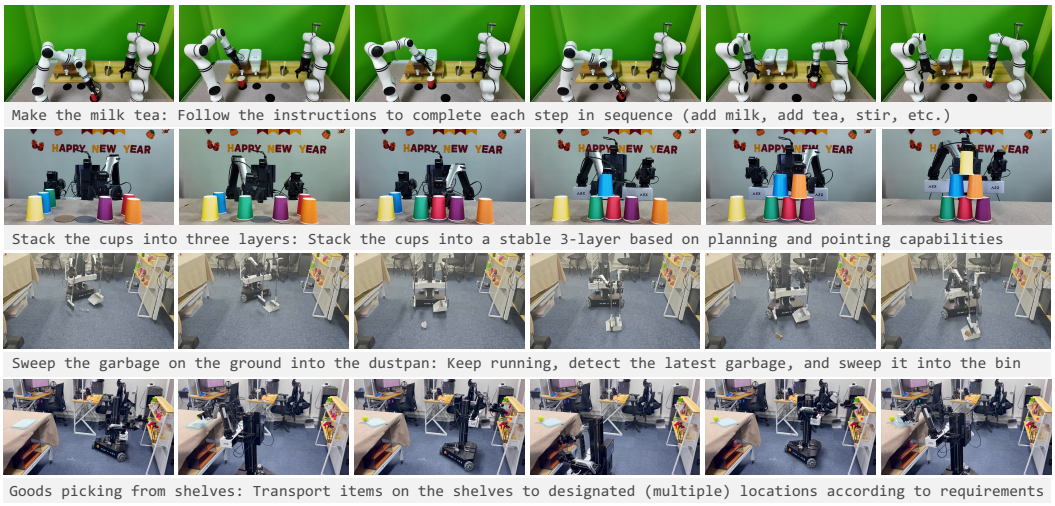


Figure 5: **Qualitative demonstrations of the PGC closed-loop framework.** Keyframes from four representative long-horizon tasks, covering sequential and cyclic execution with error detection and recovery.

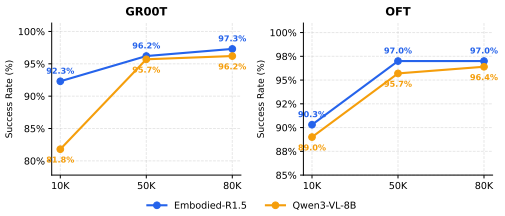


Figure 6: **VLM backbone comparison on LIBERO.** Overall success rate (%) of Embodied-R1.5 vs. Qwen3-VL-8B with two action experts at different training steps.

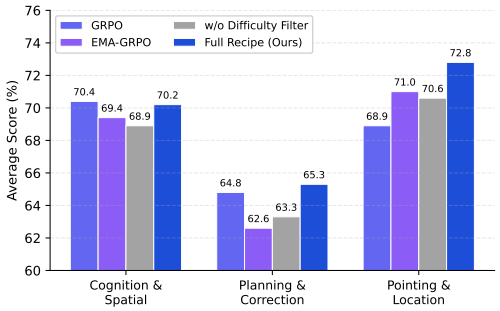


Figure 7: **RFT algorithm ablation.** Average score across three capability dimensions for different RL configurations.

7 Conclusion

We present Embodied-R1.5, a unified 8B-parameter Embodied Foundation Model that jointly internalizes spatial cognition, task planning with correction, and embodied pointing within a single architecture. Supported by a 15B+ token corpus combining existing open-source and newly constructed data, a two-stage SFT-to-RFT paradigm with multi-task balanced RL, and the Planner-Grounder-Corrector closed-loop framework, Embodied-R1.5 achieves the best results among compared baselines on 16 out of 24 embodied VLM benchmarks, surpassing Gemini-Robotics-ER-1.5. Without large-scale action pretraining, Embodied-R1.5-VLA achieves strong results on three simulation suites after action fine-tuning. Separate affordance and zero-shot real-robot evaluations demonstrate transfer through pointing and PGC, while the Corrector ablation quantifies the benefit of feedback in the evaluated tasks. Limitations and future directions are discussed in Sec. G.

Acknowledgement

This work is supported by the National Natural Science Foundation of China Youth Student Basic Research Project (Grant No. 625B2128). This work is also supported by the National Natural Science Foundation of China (Grant Nos. 62422605, 92370132). Yifu Yuan is also supported by the Young Science and Technology Scientists Sponsorship Program by CAST - Doctoral Student Special Plan.

References

[1] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini Robotics: Bringing AI into the physical world. *arXiv preprint arXiv:2503.20020*, 2025. doi: 10.48550/arXiv.2503.20020. URL https://arxiv.org/abs/2503.20020.

[2] Yifu Yuan, Haiqin Cui, Yaoting Huang, Yibin Chen, Fei Ni, Zibin Dong, Pengyi Li, Yan Zheng, and Jianye Hao. Embodied-R1: Reinforced embodied reasoning for general robotic manipulation. *arXiv preprint arXiv:2508.13998v1*, 2025. URL https://arxiv.org/abs/2508.13998v1.

[3] Fei Ni, Min Zhang, Pengyi Li, Yifu Yuan, Lingfeng Zhang, Yuecheng Liu, Peilong Han, Longxin Kou, Shaojin Ma, Jinbin Qiao, David Gamaliel Arcos Bravo, Yuening Wang, Xiao Hu, Zhanguang Zhang, Xianze Yao, Yutong Li, Zhao Zhang, Ying Wen, Ying-Cong Chen, Xiaodan Liang, Liang Lin, Bin He, Haitham Bou-Ammar, He Wang, Huazhe Xu, Jiankang Deng, Shan Luo, Shuqiang Jiang, Wei Pan, Yang Gao, Stefanos Zafeiriou, Jan Peters, Yuzheng Zhuang, Yingxue Zhang, Yan Zheng, Hongyao Tang, and Jianye Hao. Embodied arena: A comprehensive, unified, and evolving evaluation platform for embodied ai. *arXiv preprint arXiv:2509.15273*, 2025. doi: 10.48550/arXiv.2509.15273. URL https://arxiv.org/abs/2509.15273.

[4] Yifu Yuan, Haiqin Cui, Yibin Chen, Zibin Dong, Fei Ni, Longxin Kou, Jinyi Liu, Pengyi Li, Yan Zheng, and Jianye Hao. From seeing to doing: Bridging reasoning and decision for robotic manipulation. *arXiv preprint arXiv:2505.08548*, 2025.

[5] Yaoting Huang, Yifu Yuan, Linqi Han, Chengwen Li, Shuoheng Zhang, Xianze Yao, Hongyao Tang, Yan Zheng, and Jianye Hao. RoboPIN: Grounded embodied reasoning via pinned chain-of-thought. In *Proceedings of the ACM International Conference on Multimedia*, 2026. URL https://arxiv.org/abs/2606.15753.

[6] Pierre Sermanet, Tianli Ding, Jeffrey Zhao, Fei Xia, Debidatta Dwibedi, Keerthana Gopalakrishnan, Christine Chan, Gabriel Dulac-Arnold, Sharath Maddineni, Nikhil J. Joshi, Pete Florence, Wei Han, Robert Baruch, Yao Lu, Suvir Mirchandani, Peng Xu, Pannag Sanketi, Karol Hausman, Izhak Shafran, Brian Ichter, and Yuan Cao. Robovqa: Multimodal long-horizon reasoning for robotics. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pages 645–652. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10610216. URL https://doi.org/10.1109/ICRA57147.2024.10610216.

[7] Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, Xinda Xue, Qinghang Su, Huaihai Lyu, Xiaolong Zheng, Jiaming Liu, Zhongyuan Wang, and Shanghang Zhang. RoboBrain: A unified brain model for robotic manipulation from abstract to concrete. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1724–1734, 2025. doi: 10.1109/CVPR52734.2025.00168. URL https://openaccess.thecvf.com/content/CVPR2025/html/Ji_RoboBrain_A_Unified_Brain_Model_for_Robotic_Manipulation_from_Abstract_CVPR_2025_paper.html.

[8] BAAI RoboBrain Team, Mingyu Cao, Huajie Tan, Yuheng Ji, Xiansheng Chen, Minglan Lin, Zhiyu Li, Zhou Cao, Pengwei Wang, Enshen Zhou, Yi Han, Yingbo Tang, Xiangqi Xu, Wei Guo, Yaoxu Lyu, Yijie Xu, Jiayu Shi, Mengfei Du, Cheng Chi, Mengdi Zhao, Xiaoshuai Hao, Junkai Zhao, Xiaojie Zhang, Shanyu Rong, Huaihai Lyu, Zhengliang Cai, Yankai Fu, Ning Chen, Bolun Zhang, Lingfeng Zhang, Shuyi Zhang, Dong Liu, Xi Feng, Songjing Wang, Xiaodan Liu, Yance Jiao, Mengsi Lyu, Zhuo Chen, Chenrui He, Yulong Ao, Xue Sun, Zheqi He, Jingshu Zheng, Xi Yang, Donghai Shi, Kunchang Xie, Bochao Zhang, Shaokai Nie, Chunlei Men, Yonghua Lin, Zhongyuan Wang, Tiejun Huang, and Shanghang Zhang. Robobrain 2.0 technical report. *arXiv preprint arXiv:2507.02029*, 2025.

[9] Baiyu Pan, Daqin Luo, Junpeng Yang, Jiyuan Wang, Yixuan Zhang, Hailin Shi, and Jichao Jiao. Thinker: A vision-language foundation model for embodied intelligence. *CoRR*,

abs/2601.21199, 2026. doi: 10.48550/ARXIV.2601.21199. URL https://doi.org/10.48550/arXiv.2601.21199.

[10] Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *CoRR*, abs/2406.10721, 2024. doi: 10.48550/ARXIV.2406.10721. URL https://doi.org/10.48550/arXiv.2406.10721.

[11] Ronghao Dang, Jiayan Guo, Bohan Hou, Sicong Leng, Kehan Li, Xin Li, Jiangpin Liu, Yunxuan Mao, Zhikai Wang, Yuqian Yuan, Minghao Zhu, Xiao Lin, Yang Bai, Qian Jiang, Yaxi Zhao, Minghua Zeng, Junlong Gao, Yuming Jiang, Jun Cen, Siteng Huang, Liuyi Wang, Wenqiao Zhang, Chengju Liu, Jianfei Yang, Shijian Lu, and Deli Zhao. Rynnbrain: Open embodied foundation models. *CoRR*, abs/2602.14979, 2026. doi: 10.48550/ARXIV.2602.14979. URL https://doi.org/10.48550/arXiv.2602.14979.

[12] Kaituo Feng, Manyuan Zhang, Hongyu Li, Kaixuan Fan, Shuang Chen, Yilei Jiang, Dian Zheng, Peiwen Sun, Yiyuan Zhang, Haoze Sun, Yan Feng, Peng Pei, Xunliang Cai, and Xiangyu Yue. Onethinker: All-in-one reasoning model for image and video. *CoRR*, abs/2512.03043, 2025. doi: 10.48550/ARXIV.2512.03043. URL https://doi.org/10.48550/arXiv.2512.03043.

[13] Gemini Robotics Team, Abbas Abdolmaleki, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Ashwin Balakrishna, Nathan Batchelor, Alex Bewley, Jeff Bingham, et al. Gemini Robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer. *arXiv preprint arXiv:2510.03342*, 2025. doi: 10.48550/arXiv.2510.03342. URL https://arxiv.org/abs/2510.03342.

[14] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *CoRR*, abs/2504.16054, 2025. doi: 10.48550/ARXIV.2504.16054. URL https://doi.org/10.48550/arXiv.2504.16054.

[15] Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. ManipLLM: Embodied multimodal large language model for object-centric robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18061–18070, 2024.

[16] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV 2023*, pages 4195–4205, 2023. arXiv:2212.09748.

[17] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations, ICLR 2023*, 2023. arXiv:2210.02747.

[18] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. doi: 10.48550/arXiv.2503.14734. URL https://arxiv.org/abs/2503.14734.

[19] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems, RSS 2023*, 2023. arXiv:2304.13705.

[20] Jianke Zhang, Xiaoyu Chen, Qiuyue Wang, Mingsheng Li, Yanjiang Guo, Yucheng Hu, Jiajun Zhang, Shuai Bai, Junyang Lin, and Jianyu Chen. Vlm4vla: Revisiting vision-language-models in vision-language-action models. *arXiv preprint arXiv:2601.03309*, 2026.

[21] Hao Li, Ziqin Wang, Zi han Ding, Shuai Yang, Yilun Chen, Yang Tian, Xiaolin Hu, Tai Wang, Dahua Lin, Feng Zhao, Si Liu, and Jiangmiao Pang. Robointer: A holistic intermediate representation suite towards robotic manipulation. *arXiv preprint arXiv:2602.09973*, 2026.

[22] Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Dilin Wang, Zhicheng Yan, Hongyu Xu, Justin Theiss, Tianlong Chen, Jiachen Li, Zhengzhong Tu, Zhangyang Wang, and Rakesh Ranjan. VLM-3R: vision-language models augmented with instruction-aligned 3d reconstruction. *CoRR*, abs/2505.20279, 2025. doi: 10.48550/ARXIV.2505.20279. URL https://doi.org/10.48550/arXiv.2505.20279.

[23] Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis Brown, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, Daohan Lu, Rob Fergus, Yann LeCun, Li Fei-Fei, and Saining Xie. Cambrian-s: Towards spatial supersensing in video. *CoRR*, abs/2511.04670, 2025. doi: 10.48550/ARXIV.2511.04670. URL https://doi.org/10.48550/arXiv.2511.04670.

[24] Arijit Ray, Jiafei Duan, Reuben Tan, Dina Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A Plummer, Ranjay Krishna, Kuo-Hao Zeng, et al. Sat: Spatial aptitude training for multimodal language models. *arXiv preprint arXiv:2412.07755*, 3, 2024.

[25] Zhipeng Cai, Ching-Feng Yeh, Hu Xu, Zhuang Liu, Gregory Meyer, Xinjie Lei, Changsheng Zhao, Shang-Wen Li, Vikas Chandra, and Yangyang Shi. Depthlm: Metric depth from vision language models. *arXiv preprint arXiv:2509.25413*, 2025.

[26] Kaiyuan Chen, Shuangyu Xie, Zehan Ma, Pannag R. Sanketi, and Ken Goldberg. Robo2VLM: Visual question answering from large-scale in-the-wild robot manipulation datasets. *arXiv preprint arXiv:2505.15517*, 2025. URL https://arxiv.org/abs/2505.15517.

[27] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, K. Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, A. Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan C. Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, S. Levine, Yao Lu, U. Malla, D. Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, M. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, S. Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Q. Vuong, F. Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. *ArXiv*, abs/2212.06817, 2022.

[28] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023.

[29] Alexander Khazatsky, Karl Pertsch, S. Nair, Ashwin Balakrishna, S. Dasari, Siddharth Karamcheti, Soroush Nasiriany, M. K. Srirama, L. Chen, Kirsty Ellis, P. Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Ye Ma, Patrick Miller, Jimmy Wu, Suneel Belkhale, S. Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, K. Hatch, Shan Lin, Jingpei Lu, Jean-Pierre Mercat, Abdul Rehman, Pannag R. Sanketi, Archit Sharma, C. Simpson, Q. Vuong, H. Walke, Blake Wulfe, Ted Xiao, J. Yang, Arefeh Yavary, Tony Zhao, Christopher Agia, R. Baijal, Mateo Guaman Castro, D. Chen, Qiuyu Chen, T. Chung, Jaimyn Drake, E. P. Foster, Jensen Gao, D. Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Muhammad Zubair Irshad, Donovon Jackson, Charlotte Le, Yunshuang Li, K. Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, D. Morton, Tony Nguyen, Abigail O'Neill, R. Scalise, Derick Seale, V. Son, Stephen Tian, E. Tran, Andrew Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, O. Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, D. Jayaraman, Joseph J. Lim, Jitendra Malik, Roberto Mart'in-Mart'in, S. Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, T. Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset. *ArXiv*, abs/2403.12945, 2024.

[30] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546*, 2025.

[31] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded SAM: assembling open-world models for diverse visual tasks. *CoRR*, abs/2401.14159, 2024. doi: 10.48550/ARXIV.2401.14159. URL https://doi.org/10.48550/arXiv.2401.14159.

[32] Yi Chen, Yuying Ge, Yixiao Ge, Mingyu Ding, Bohao Li, Rui Wang, Ruifeng Xu, Ying Shan, and Xihui Liu. Egoplan-bench: Benchmarking multimodal large language models for human-level planning. *arXiv preprint arXiv:2312.06722*, 2023.

[33] Paul Pacaud, Ricardo Garcia, Shizhe Chen, and Cordelia Schmid. Guardian: Detecting robotic planning and execution errors with vision-language models. *CoRR*, abs/2512.01946, 2025. doi: 10.48550/ARXIV.2512.01946. URL https://doi.org/10.48550/arXiv.2512.01946.

[34] Zeyi Liu, Arpit Bahety, and Shuran Song. Reflect: Summarizing robot experiences for failure explanation and correction. *arXiv preprint arXiv:2306.15724*, 2023.

[35] Zewei Ye, Weifeng Lu, Minghao Ye, Tao Lin, Shuo Yang, Junchi Yan, and Bo Zhao. Robofac: A comprehensive framework for robotic failure analysis and correction. *arXiv preprint arXiv:2505.12224*, 2025.

[36] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh Nagaswamy Rajesh, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2024.

[37] Ricardo Garcia, Shizhe Chen, and Cordelia Schmid. Towards generalizable vision-language robotic manipulation: A benchmark and llm-guided 3d policy. In *IROS*, 2024.

[38] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. Sapien: A simulated part-based interactive environment. In *CVPR*, 2020.

[39] Abhay Deshpande, Yuquan Deng, Arijit Ray, Jordi Salvador, Winson Han, Jiafei Duan, Kuo-Hao Zeng, Yuke Zhu, Ranjay Krishna, and Rose Hendrix. Graspmolmo: Generalizable task-oriented grasping via large-scale synthetic data generation. *arXiv preprint arXiv:2505.13441*, 2025.

[40] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[41] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024.

[42] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

[43] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024.

[44] Shijie Lian, Changti Wu, Laurence Tianruo Yang, Hang Yuan, Bin Yu, Lei Zhang, and Kai Chen. Euclid's gift: Enhancing spatial perception and reasoning in vision-language models via geometric surrogate tasks. *arXiv preprint arXiv:2509.24473*, 2025.

[45] Yi Zhang, Bolin Ni, Xin-Sheng Chen, Heng-Rui Zhang, Yongming Rao, Houwen Peng, Qinglin Lu, Han Hu, Meng-Hao Guo, and Shi-Min Hu. Bee: A high-quality corpus and full-stack suite to unlock advanced fully open mllms, 2025.

[46] Shengyuan Ding, Shenxi Wu, Xiangyu Zhao, Yuhang Zang, Haodong Duan, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Mm-ifengine: Towards multimodal instruction following. *arXiv preprint arXiv:2504.07957*, 2025.

[47] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[48] Shuoheng Zhang, Yifu Yuan, Hongyao Tang, Yan Zheng, Qiaojun Yu, Pengyi Li, Guowei Huang, Helong Huang, Xingyue Quan, and Jianye Hao. ForceFlow: Learning to feel and act via contact-driven flow matching. In *Advances in Neural Information Processing Systems*, 2026. URL https://arxiv.org/abs/2605.11048.

[49] Xiaoshuai Hao, Lei Zhou, Zhijian Huang, Zhiwen Hou, Yingbo Tang, Lingfeng Zhang, Guang Li, Zheng Lu, Shuhuai Ren, Xianhui Meng, et al. Mimo-embodied: X-embodied foundation model technical report. *arXiv preprint arXiv:2511.16518*, 2025.

[50] Yi Zhang, Che Liu, Xiancong Ren, Hanchu Ni, Shuai Zhang, Zeyuan Ding, Jiayu Hu, Hanzhe Shan, Zhenwei Niu, Zhaoyang Liu, et al. Pelican-vl 1.0: A foundation brain model for embodied intelligence. *arXiv preprint arXiv:2511.00108*, 2025.

[51] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16488–16498, 2024.

[52] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Ziteng Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024.

[53] Mengfei Du, Binhao Wu, Zejun Li, Xuanjing Huang, and Zhongyu Wei. EmbSpatial-Bench: Benchmarking spatial understanding for embodied tasks with large vision-language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 346–355, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-short.33. URL https://aclanthology.org/2024.acl-short.33/.

[54] Arijit Ray, Jiafei Duan, Reuben Tan, D. Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A. Plummer, Ranjay Krishna, Kuo-Hao Zeng, and Kate Saenko. Sat: Dynamic spatial aptitude training for multimodal language models. 2024.

[55] Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. Robospatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics. *arXiv preprint arXiv:2411.16537*, 2024.

[56] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. *arXiv preprint arXiv:2404.12390*, 2024.

[57] 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.

[58] Enshen Zhou, Jingkun An, Cheng Chi, Yi Han, Shanyu Rong, Chi Zhang, Pengwei Wang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, and Shanghang Zhang. Roborefer: Towards spatial referring with reasoning in vision-language models for robotics. *arXiv preprint arXiv:2506.04308*, 2025.

[59] Yuhao Lu, Yixuan Fan, Beixing Deng, Fangfu Liu, Yali Li, and Shengjin Wang. Vl-grasp: a 6-dof interactive grasp policy for language-oriented objects in cluttered indoor scenes. *arXiv preprint arXiv:2308.00640*, 2023.

[60] Xiaoshuai Hao, Yingbo Tang, Lingfeng Zhang, Yanbiao Ma, Yunfeng Diao, Ziyu Jia, Wenbo Ding, Hangjun Ye, and Long Chen. Roboafford++: A generative ai-enhanced dataset for multimodal affordance learning in robotic manipulation and navigation. *arXiv preprint arXiv:2511.12436*, 2025.

[61] Long Cheng, Jiafei Duan, Yi Ru Wang, Haoquan Fang, Boyang Li, Yushan Huang, Elvis Wang, Ainaz Eftekhar, Jason Lee, Wentao Yuan, Rose Hendrix, Noah A. Smith, Fei Xia, Dieter Fox, and Ranjay Krishna. Pointarena: Probing multimodal grounding through language-guided pointing. *arXiv preprint arXiv:2505.09990*, 2025.

[62] Haotian Xue, Yunhao Ge, Yu Zeng, Zhaoshuo Li, Ming-Yu Liu, Yongxin Chen, and Jiaojiao Fan. Point-it-out: Benchmarking embodied reasoning for vision language models in multi-stage visual grounding. *arXiv preprint arXiv:2509.25794*, 2025.

[63] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 91–104, 2025.

[64] Lu Qiu, Yi Chen, Yuying Ge, Yixiao Ge, Ying Shan, and Xihui Liu. Egoplan-bench2: A benchmark for multimodal large language model planning in real-world scenarios. *arXiv preprint arXiv:2412.04447*, 2024.

[65] Alisson Azzolini, Junjie Bai, Hannah Brandon, Jiaxin Cao, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, et al. Cosmos-reason1: From physical common sense to embodied reasoning. *arXiv preprint arXiv:2503.15558*, 2025.

[66] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[67] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[68] NVIDIA. Gr00t n1.5: Advancing generalist robot foundation models. https://developer.nvidia.com/blog/advancing-generalist-robot-foundation-models-with-gr00t-n1-5/, 2025.

[69] NVIDIA GEAR Team. GR00T N1.6: An improved open foundation model for generalist humanoid robots. NVIDIA Research, December 2025. URL https://research.nvidia.com/labs/gear/gr00t-n1_6/. Released December 15, 2025; accessed November 7, 2026.

[70] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

[71] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *Robotics: Science and Systems*, 2023.

[72] Kaichun Mo, Leonidas J. Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. 2021.

[73] Chengliang Zhong, Yuhang Zheng, Yupeng Zheng, Hao Zhao, Li Yi, Xiaodong Mu, Ling Wang, Pengfei Li, Guyue Zhou, and Chao Yang. 3d implicit transporter for temporally consistent keypoint discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3869–3880, 2023.

[74] Zibin Dong, Yicheng Liu, Shiduo Zhang, Baijun Ye, Yifu Yuan, Fei Ni, Jingjing Gong, Xipeng Qiu, Hang Zhao, Yinchuan Li, and Jianye Hao. Actioncodec: What makes for good action tokenizers. *arXiv (Cornell University)*, 2026.

[75] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

[76] StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

[77] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[78] Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

[79] Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision–language–action models for embodied AI. *arXiv preprint arXiv:2405.14093*, 2024. URL https://arxiv.org/abs/2405.14093.

[80] Ziyi Wang, Xumin Yu, Yongming Rao, Yonggen Ling, Yunheng Li, Oran Wang, Mingqi Gao, Yuchen Zhou, Yves Liang, Zuyan Liu, Yani Zhang, Rui Huang, Xiaoran Xu, Bowen Yuan, Yifu Yuan, Xu Tan, He Zhang, Yufei Huang, Shenghao Zhang, Hongsheng Wu, Han Hu, and Zhengyou Zhang. Hy-Embodied-VLM-1.0: Efficient physical-world agents. *arXiv preprint arXiv:2607.12894*, 2026. URL https://arxiv.org/abs/2607.12894.

[81] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465, 2024.

[82] Kun Ouyang, Yuanxin Liu, Haoning Wu, Yi Liu, Hao Zhou, Jie Zhou, Fandong Meng, and Xu Sun. Spacer: Reinforcing mllms in video spatial reasoning. *arXiv preprint arXiv:2504.01805*, 2025.

[83] Zibin Dong, Fei Ni, Yifu Yuan, Yinchuan Li, and Jianye Hao. EmbodiedMAE: A unified 3D multi-modal representation for robot manipulation. *arXiv preprint arXiv:2505.10105*, 2025. URL https://arxiv.org/abs/2505.10105.

[84] Gen Luo, Ganlin Yang, Ziyang Gong, Guanzhou Chen, Haonan Duan, Erfei Cui, Ronglei Tong, Zhi Hou, Tianyi Zhang, Zhe Chen, et al. Visual embodied brain: Let multimodal large language models see, think, and control in spaces. *arXiv preprint arXiv:2506.00123*, 2025.

[85] Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *Advances in Neural Information Processing Systems*, 2024.

[86] Huajie Tan, Enshen Zhou, Zhiyu Li, Yijie Xu, Yuheng Ji, Xiansheng Chen, Cheng Chi, Pengwei Wang, Huizhu Jia, Yulong Ao, et al. Robobrain 2.5: Depth in sight, time in mind. *arXiv preprint arXiv:2601.14352*, 2026.

[87] Ziyang Gong, Zehang Luo, Anke Tang, Zhe Liu, Shi Fu, Zhi Hou, Ganlin Yang, Weiyun Wang, Xiaofeng Wang, Jianbo Liu, et al. Ace-brain-0: Spatial intelligence as a shared scaffold for universal embodiments. *arXiv preprint arXiv:2603.03198*, 2026.

[88] HY Team, Xumin Yu, Zuyan Liu, Ziyi Wang, He Zhang, Yongming Rao, Fangfu Liu, Yani Zhang, Ruowen Zhao, Oran Wang, et al. Hy-embodied-0.5: Embodied foundation models for real-world agents. *arXiv preprint arXiv:2604.07430*, 2026.

[89] Fei Ni, Zhuo Chen, Yifu Yuan, Zibin Dong, Xianze Yao, Shan Luo, Jianye Hao, Jiankang Deng, and Stefanos Zafeiriou. SemanticVLA: Towards semantic reasoning over action memorization via synergistic explicit trace and latent action planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12237–12247, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Ni_SemanticVLA_Towards_Semantic_Reasoning_over_Action_Memorization_via_Synergistic_Explicit_CVPR_2026_paper.html.

[90] Haoquan Fang, Jiafei Duan, Donovan Clay, Sam Wang, Shuo Liu, Weikai Huang, Xiang Fan, Wei-Chuan Tsai, Shirui Chen, Yi Ru Wang, Shanli Xing, Jaemin Cho, Jae Sung Park, Ainaz Eftekhar, Peter Sushko, Karen Farley, Angad Wadhwa, Cole Harrison, Winson Han, Ying-Chun Lee, Eli VanderBilt, Rose Hendrix, Suveen Ellawela, Lucas Ngoo, Joyce Chai, Zhongzheng Ren, Ali Farhadi, Dieter Fox, and Ranjay Krishna. Molmoact2: Action reasoning models for real-world deployment. *arXiv preprint arXiv:2605.02881*, 2025.

[91] Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, et al. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*, 2025.

[92] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 12–22. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00008. URL https://doi.org/10.1109/ICCV51070.2023.00008.

[93] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722, 2018.

[94] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021.

[95] Angel X. Chang, Angela Dai, T. Funkhouser, Maciej Halber, M. Nießner, M. Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *2017 International Conference on 3D Vision (3DV)*, pages 667–676, 2017.

[96] Delin Qu, Haoming Song, Qizhi Chen, Zhaoqing Chen, Xianqiang Gao, Xinyi Ye, Qi Lv, Modi Shi, Guanghui Ren, Cheng Ruan, et al. Eo-1: Interleaved vision-text-action pretraining for general robot control. *arXiv preprint arXiv:2508.21112*, 2025.

[97] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

[98] Baoqi Pei, Yifei Huang, Jilan Xu, Yuping He, Guo Chen, Fei Wu, Jiangmiao Pang, and Yu Qiao. Egothinker: Unveiling egocentric reasoning with spatio-temporal cot. *Advances in Neural Information Processing Systems*, 38:44140–44168, 2026.

[99] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024. URL https://arxiv.org/abs/2407.08693.

[100] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014.

[101] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pages 28085–28128, 2025.

[102] Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Rohun Tripathi, Sangho Lee, Mohammadreza Salehi, Jason Ren, Chris Dongjoo Kim, Yinuo Yang, et al. Molmo2: Open weights and data for vision-language models with video understanding and grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28652–28668, 2026.

[103] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019.

[104] Jierun Chen, Fangyun Wei, Jinjing Zhao, Sizhe Song, Bohuai Wu, Zhuoxuan Peng, S-H Gary Chan, and Hongyang Zhang. Revisiting referring expression comprehension evaluation in the era of large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 513–524, 2025.

[105] Andrew Guo, Bowen Wen, Jianhe Yuan, Jonathan Tremblay, Stephen Tyree, Jeffrey Smith, and Stan Birchfield. Handal: A dataset of real-world manipulable object categories with pose annotations, affordances, and reconstructions, 2023. URL https://arxiv.org/abs/2308.01477.

[106] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, Amir Mousavi, Yiwen Song, Abhimanyu Dubey, and Dhruv Mahajan. PACO: Parts and attributes of common objects. In *arXiv preprint arXiv:2301.01795*, 2023.

[107] Zifu Wan, Yaqi Xie, Ce Zhang, Zhiqiu Lin, Zihan Wang, Simon Stepputtis, Deva Ramanan, and Katia P Sycara. Instructpart: Task-oriented part segmentation with instruction reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24202–24227, 2025.

[108] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[109] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2432–2443. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.261. URL https://doi.org/10.1109/CVPR.2017.261.

[110] Afshin Dehghan, Gilad Baruch, Zhuoyuan Chen, Yuri Feigin, Peter Fu, Thomas Gebauer, Daniel Kurz, Tal Dimry, Brandon Joffe, Arik Schwartz, and Elad Shulman. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. 2021.

[111] Open X-Embodiment Collaboration. Open x-embodiment: Robotic learning datasets and RT-X models. *CoRR*, abs/2310.08864, 2023. doi: 10.48550/ARXIV.2310.08864. URL https://doi.org/10.48550/arXiv.2310.08864.

[112] Xinyu Huang, Yi-Jie Huang, Youcai Zhang, Weiwei Tian, Rui Feng, Yuejie Zhang, Yanchun Xie, Yaqian Li, and Lei Zhang. *Open-Set Image Tagging with Multi-Grained Text Supervision*. 2023.

[113] Ning Gao, Yilun Chen, Shuai Yang, Xinyi Chen, Yang Tian, Hao Li, Haifeng Huang, Hanqing Wang, Tai Wang, and Jiangmiao Pang. Genmanip: Llm-driven simulation for generalizable instruction-following manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12187–12198, 2025.

[114] Yang Tian, Yuyin Yang, Yiman Xie, Zetao Cai, Xu Shi, Ning Gao, Hangxu Liu, Xuekun Jiang, Zherui Qiu, Feng Yuan, et al. Interndata-a1: Pioneering high-fidelity synthetic data for pre-training generalist policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 976–985, 2026.

[115] Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM international conference on multimedia*, pages 11198–11201, 2024.

[116] Kaichen Zhang, Bo Li, Peiyuan Gao, Fanyi Zhang, Kairui Li, Jingkang Yan, and Ziwei Liu. Lmms-eval: Reality check on the evaluation of large multimodal models. *arXiv preprint arXiv:2407.12772*, 2024.

[117] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Haotong Zhang, and Ion Stoica. *Efficient Memory Management for Large Language Model Serving with PagedAttention*. 2023.

[118] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

A Related Work

The pursuit of general physical intelligence has driven rapid development of Embodied Foundation Models (EFMs) [79, 80]. Early efforts focus on individual capability dimensions, e.g., spatial reasoning [81, 82], 3D representations [83], physical scene understanding [84, 65], planning [6, 85], and embodied grounding [10, 60, 4, 2]. Recent works attempt broader unification: RoboBrain [8, 86] integrates spatial and temporal reasoning via chain-of-thought; MIMO-Embodied [49] and ACE-Brain-0 [87] fuse capabilities across driving and manipulation; Pelican-VL [50] and RynnBrain [11] target unified embodied reasoning with grounded pointing; and HY-Embodied [88] proposes iterative self-evolution. Embodied-R1.5 further unifies all three dimensions within a single model and proposes a multi-task balanced RL recipe to mitigate heterogeneous training interference. EFMs can also serve as the foundation for VLA training [89]. $\pi_{0.5}$ [14] co-trains with embodied reasoning data to mitigate forgetting, Gemini Robotics 1.5 [13] combines an embodied reasoning model with a VLA, and MolmoAct2 [90] injects adaptive reasoning into action generation. Embodied-R1.5-VLA demonstrates effective action fine-tuning from an embodied backbone without large-scale action pretraining.

B Detailed Capability Taxonomy

This appendix provides the complete sub-capability definitions for the three embodied reasoning dimensions.

B.1 Embodied Cognition & Spatial Reasoning

This dimension requires the model to perceive and understand the structure of the physical world, encompassing both static geometric relations and dynamic interaction possibilities. We decompose it into four sub-dimensions:

- **Spatial Relation Understanding.** Recognizing relative orientations between objects, support relations, containment, as well as occlusion and visibility from the current viewpoint. This is the foundation for all downstream planning and localization.
- **Metric Spatial Reasoning.** Reasoning about metric information in the physical world, including estimating distances between objects, judging object sizes, and determining whether a free space can accommodate a target object. This provides quantitative grounding for precise manipulation.
- **3D Scene Perception.** Performing monocular depth estimation, cross-view geometric reasoning, and maintaining spatial consistency across video frames, providing geometric priors for collision avoidance and precise manipulation.
- **Object & Scene Cognition.** Understanding scenes from a robot-centric or first-person perspective, distinguishing manipulation targets from obstacles and inferring the task context.

B.2 Embodied Planning & Correction

This dimension covers the entire life cycle of task execution. We organize it into two complementary aspects:

Task Planning.

- **Long-horizon Task Decomposition.** Takes the overall language instruction and current scene observation to output a structured sub-task sequence.
- **Next-step Planning.** Rolls out the next atomic action instruction during execution, conditioned on current progress and observation, directly serving closed-loop control.

Process Monitoring & Correction.

- **Process Detection.** Judges whether the current sub-task has been completed, determining when to advance to the next sub-task.

- **Error Localization.** Identifies the specific cause of failure from both execution and planning perspectives once a failure is detected.
- **Error Correction.** Generates concrete corrective suggestions (replanning sub-tasks or rolling back to retry), which feed directly as input context for the next round of planning.

B.3 Embodied Pointing & Location

Pointing serves as the critical interface connecting embodied reasoning to physical execution. Embodied-R1.5 substantially strengthens open-vocabulary pointing, with particularly significant advances in physical affordance understanding. The model treats coordinates as first-class symbols that can be repeatedly referenced and reasoned about, maintaining visual anchoring throughout multi-step reasoning. The pointing capability is organized into four types:

- **Referring Expression Grounding (REG).** Given a natural-language expression, the model outputs the corresponding object's point coordinate. This is the most basic form of grounding, serving both standalone object localization and as a building block for more complex grounding tasks.
- **Region Referring Grounding (RRG).** RRG handles regions rather than objects, for example identifying *"an empty region that can hold a plate"*. This capability is essential for object placement and spatial arrangement in manipulation tasks.
- **Object Functional Grounding (OFG).** Beyond object recognition, OFG further localizes an object's functional parts (spout, handle, button, etc.). Understanding which part of an object to interact with is critical for generating correct grasping strategies and manipulation plans.
- **Visual Trace Generation (VTG).** VTG generates ordered point sequences that describe complete manipulation trajectories. Embodied-R1.5 extends this to simultaneously support *object flow* (the expected motion path of the target object from its starting position to the goal position) and *end-effector flow* (the expected motion path of the robot end-effector). This dual-flow formulation enables the model to reason about both what should move and how the robot should move to achieve it.

C Training Details

C.1 SFT Training Configuration

Starting from the pretrained Qwen3-VL-8B-Instruct backbone, we perform full-parameter fine-tuning via the LLaMA-Factory[3] framework. We adopt the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and weight decay of 0.01. Training is conducted with bfloat16 precision, a peak learning rate of 2×10^{-6}, cosine learning rate decay with 10% warmup steps and a minimum learning rate of 2×10^{-7}. We set per-device batch size to 4 with 4 gradient accumulation steps, yielding an effective global batch size of 512. Gradient clipping is applied with a maximum norm of 1.0. We employ DeepSpeed ZeRO-2 for efficient distributed training. The maximum context sequence length is set to 8192 tokens. The input image resolution is limited to 3 million pixels, and video resolution is capped at 802,816 pixels. The vision encoder is fully tuned together with the LLM backbone, and only standard causal language modeling (CLM) loss is used for optimization. We train the model for one epoch on 32 NVIDIA H20-96G GPUs.

C.2 RFT Training Configuration

The RL stage is initialized from the Embodied-R1.5-SFT checkpoint produced by the SFT stage. We adopt the EasyR1 framework[4] for online reinforcement learning, extending it to support mixed video-and-image inputs as well as scoring from an external reward model (RM). Training is based on GRPO combined with the improvements described in the main text. The clip ratio lower/upper thresholds are set to 0.2/0.28; since embodied reasoning problems involve relatively short responses, the actual clipping rate during training is extremely low and this parameter has minimal effect. For KL divergence, we adopt the KL-in-reward formulation with a penalty coefficient of 0.01. We also enable online filtering, discarding samples whose overall reward falls outside the [0.01, 0.99] quantile

[3] https://github.com/hiyouga/LLaMA-Factory
[4] https://github.com/hiyouga/EasyR1

range to stabilize training. The entire training runs on 32 NVIDIA H20-96G GPUs for 2 epochs with approximately 200K samples per epoch, taking roughly 2 days in total. The reward function design is detailed in Sec. D. Each prompt is sampled with $n = 8$ parallel rollouts. The rollout batch size is 1024, the generation-phase mini rollout batch size is 256, and the actor update global batch size is 512; the per-device micro batch size is 4 for the update phase and 8 for experience collection. We use AdamW with a learning rate of 3×10^{-6}, weight decay of 0.01, and gradient clipping at 1.0. The vision encoder is jointly trained (not frozen), gradient checkpointing is enabled to reduce memory footprint, and the actor employs ZeRO-2 for distributed training. The maximum output response length is 2048 tokens.

C.3 Embodied-R1.5-VLA Training Configuration

Embodied-R1.5-VLA adopts a dual-system architecture comprising a VLM backbone and a flow-matching action head. The VLM backbone processes visual observations and language instructions, producing hidden representations of dimension 2048. The action head employs a DiT-B architecture (16 layers, 768-dim, 12 attention heads) that uses VLM hidden states as cross-attention context, combined with 32 learnable future query tokens, to generate 7-dimensional continuous action trajectories via flow matching. The total model size is 8B parameters, all trained end-to-end without freezing. During training, timesteps are sampled from a Beta$(1.5, 1.0)$ distribution and discretized into 1000 buckets. During inference, actions are generated from Gaussian noise via 4-step Euler integration. Common training hyperparameters are as follows: AdamW optimizer (β_1=0.9, β_2=0.95), VLA-to-VLM loss weight ratio of 1.0:0.1, action model learning rate 10^{-4}, and VLM learning rate 10^{-5}. The total batch size is 128 on 8 NVIDIA H20-96GB GPUs using DeepSpeed ZeRO-2 with BF16 precision and gradient clipping threshold of 1.0. All VLA experiments are conducted using the starVLA framework[5].

For the LIBERO environment, the action prediction horizon is 8 steps with delta joint position (delta_qpos) actions and min-max normalization. Observations include dual-view images (primary camera and wrist camera) at 224×224 resolution, and training data combines four LIBERO subsets (spatial, object, goal, and libero_10). During evaluation, we run 50 episodes per task with maximum step limits of 220/280/300/520 depending on the subset. For the SimplerEnv benchmark, the action prediction horizon is 16 steps with delta end-effector (delta_ee) actions and q99 normalization. Observations consist of a single camera view at 224×224 resolution, and training data is a mixture of Bridge and Fractal (RT-1) datasets.

D Multi-Task Reward Design Details

Following the principle that *reward structure should match output structure*, we design five families of reward functions for the heterogeneous output types in embodied reasoning. For distance-based point and trajectory rewards, we employ a shared **piecewise-linear decay** framework: given a distance metric d and threshold pair (τ_p, τ_z),

$$\phi(d;\, \tau_p, \tau_z) = \mathrm{clip}\left(\frac{\tau_z - d}{\tau_z - \tau_p},\ 0,\ 1\right), \tag{1}$$

which assigns full credit when $d < \tau_p$, zero credit when $d \geq \tau_z$, and linearly decays in between. This partial-credit design provides dense gradient signals for RL training, avoiding the sparsity inherent in binary rewards.

(1) Exact-match reward (deterministic-answer tasks). For multiple-choice questions, numerical comparisons, and mathematical derivations that admit a single correct answer, we use a binary reward $R = \mathbb{1}[\mathrm{match}(\hat{y},\, y^*)]$, outputting $\{0, 1\}$. Multiple-choice grading employs a flexible matching mechanism that accepts answer variants such as "A", "A.dog", while treating ordering questions (e.g., "1-3-2") as strict exact matches. Numerical answers are rounded to one decimal place before comparison. Math problems use symbolic equivalence verification to determine whether two mathematical expressions are equivalent.

[5] https://github.com/starVLA/starVLA

(2) IoU reward (spatial grounding tasks). For tasks whose output is a bounding box, we compute the standard 2D Intersection-over-Union between the predicted box b and the ground-truth box b^*: $R_{\text{IoU}} = |b \cap b^*|/|b \cup b^*|$, yielding a reward in $[0, 1]$. The continuous-valued IoU naturally provides partial credit.

(3) Point-distance reward (point localization tasks). For REG, OFG, and other point localization tasks, we compute the *average nearest-neighbor distance* d_{nn} between the predicted and ground-truth point sets, then apply the piecewise-linear decay: $R = \phi(d_{\text{nn}};\, 40,\, 150)$ (thresholds in pixels). When the ground truth is a segmentation polygon or bounding box, we switch to a *point-in-region* check and reward the fraction of predicted points that fall within the target region. An additional count penalty $\delta_c = 0.3$ is applied when the number of predicted points does not match the ground truth.

(4) Trajectory-RMSE reward (trajectory prediction tasks). For 2D and 3D trajectory tasks, we first align predicted and ground-truth trajectories to an equal number of sampling points via linear interpolation, then use per-point RMSE as the distance metric: $R_{\text{2D}} = \phi(\text{RMSE}_{\text{2D}};\, 50,\, 120)$. For 3D trajectories, the depth dimension is independently evaluated via MAE: $R_{\text{depth}} = \phi(\text{MAE}_d;\, 0.1,\, 0.4)$, and the final reward is an equal-weight average $R = 0.5\, R_{\text{2D}} + 0.5\, R_{\text{depth}}$. A length-mismatch penalty $\delta_l = 0.35$ is imposed when the predicted and ground-truth trajectory lengths differ, and single-point outputs are assigned zero reward to prevent reward hacking.

(5) Semantic-similarity reward (open-ended text tasks). For open-ended outputs such as error correction suggestions and long-horizon plans, we use Skywork-Reward-V2-Qwen3-4B [91] as the default reward model for semantic quality scoring; the raw reward-model score is mapped to $[0, 1]$ via sigmoid temperature normalization. When reward-model inference fails, we fall back to the average of BLEU-1, BLEU-2, and BLEU-3 scores to measure semantic consistency, yielding a reward in $[0, 1]$.

Format reward. Across all tasks, we include a unified format-checking reward that jointly verifies two conditions: (1) the output contains the required `<answer>...</answer>` tag structure, and (2) the content inside the tags conforms to the task-specific structural format (e.g., point tasks require `point_2d` as a two-element numeric list; trajectory tasks require a `depth` field; spatial grounding tasks require `boxes` as a four-element numeric list). The format reward is binary and combined with the accuracy reward via a weighted sum:

$$R = (1 - \lambda)\, R_{\text{acc}} + \lambda\, R_{\text{fmt}}, \quad \lambda = 0.1. \tag{2}$$

E Training Data Composition Details

E.1 SFT Data Composition

This appendix provides the full data composition details described in Section 3.

E.1.1 Embodied Cognition & Spatial Reasoning Data

Multi-view spatiotemporal reasoning. Spatial reasoning and scene cognition are foundational for embodied VLMs to perceive the physical environment. Since inputs in embodied scenarios are often temporal and multi-view, we integrate VLM-3R [22], Cambrian-S [23], and SAT [24] datasets to strengthen the model's spatiotemporal reasoning under diverse viewpoints. For video inputs, we additionally incorporate multiple video spatial reasoning datasets [82, 12]. These datasets collectively cover object counting, relative distance, relative direction, spatial topological relations (above/inside/below/between, etc.), depth ordering, occlusion judgment, appearance order, object size, and absolute distance. They require the model to jointly reason about spatial information from different observation angles, establishing a foundation for 3D perception in embodied scenes.

Depth estimation. Depth perception is a critical channel for embodied VLMs to understand the 3D structure of the physical environment. We adopt indoor scene depth estimation data from DepthLM [25], which extracts sensor-level ground-truth depth values from high-quality indoor 3D datasets including ScanNet++ [92], Taskonomy [93], HM3D [94], and Matterport3D [95], constructing depth QA pairs based on specified image coordinates. The data encompasses both

absolute metric depth and relative depth: the former provides precise distance values, while the latter enhances the model's understanding of scene depth hierarchies. During data construction, the point sampling strategy explicitly excludes pixels at object boundaries, at infinity, or in physically inconsistent regions, and normalizes camera focal lengths across all images for cross-source scale unification. We deliberately select only high-quality indoor datasets, and filter unreliable depth supervision before training.

Embodied scene cognition. To adapt the model to the distinctive viewpoints and semantic requirements of robotic manipulation scenarios, we curate approximately 106K samples from Robo2VLM [26] and RoboBrain 2.0 [8], covering scene understanding question-answering from the robot's first-person perspective. This data helps the model establish environment cognition from the robot's manipulation viewpoint.

Tabletop spatial reasoning. To address the gap in fine-grained tabletop spatial reasoning, we construct the *ER1.5-Spatial* dataset ($\sim$20K) from Fractal [27], BridgeData V2 [28], and DROID [29] datasets via a fully automated 3D scene annotation pipeline. The pipeline takes a single RGB image as input and produces a structured 3D semantic scene graph, from which spatial reasoning QA pairs are programmatically generated covering spatial relations, distance metrics, scene cognition, and appearance order. Full pipeline implementation details are provided in Appendix F.1.

E.1.2 Embodied Planning Data

Embodied planning requires the model to assess the current execution state, comprehend the target task objective, and perform task decomposition and next-step planning in long-horizon tasks. We compile planning data from multiple sources, covering diverse planning demands across manipulation, navigation, and egocentric scenarios. We first integrate approximately 950K task planning QA pairs from open-source datasets including RoboVQA [6], EO-Data [96], AgiBot-World [97], and Cosmos-Reason [65], encompassing long-horizon task planning capabilities in complex embodied scenarios and forming the large-scale robot visual QA planning data. In addition, we incorporate diverse samples from EgoPlan-IT [32] and EgoRe [98], which are extracted from first-person videos and require the model to predict subsequent action sequences based on observed manipulation progress.

E.1.3 Embodied Correction Data

Error correction is critical for closed-loop autonomous execution. Existing robotic datasets predominantly contain successful demonstrations, while failure data with structured error annotations remains severely insufficient. We first incorporate 78K video question-answering pairs from the RoboFAC [35] dataset, which covers fault understanding and correction across different robots. To address comprehensive capability requirements, we draw upon the failure taxonomy established in prior work [35, 33, 34] and construct the *ER1.5-Correction* dataset, a large-scale failure correction QA dataset covering the complete pipeline from task planning to execution. Through an automated perturbation generation pipeline that enables large-scale expansion, the dataset comprises 800K samples. The dataset is organized along two orthogonal dimensions: by *capability level*, it spans failure detection (binary classification), failure localization (identifying the specific failure type and location), and failure correction (generating recovery plans from erroneous states); by *stage*, it covers planning failures (step omission, redundancy, swap, object error) and execution failures (execution interruption, wrong manipulation target, incorrect action). The combination of these two dimensions yields six QA task types. Data sources encompass diverse real-robot datasets (the CoT-annotated version of BridgeData V2 [28, 99], RoboFail [34]) and simulation datasets (ManiSkill [36], GEMBench [37]).

Specifically, for planning failures, we start from correct sub-task plans and apply five structured perturbation operators to automatically generate erroneous plans. For each perturbed plan, we simultaneously generate QA pairs at multiple capability levels, while retaining the correct plan as a positive sample in the training data to prevent overfitting. For execution failures, we adopt three complementary strategies: truncating successful demonstration videos to simulate execution interruption, replacing sub-task descriptions to simulate object/action errors, and injecting physical perturbations in ManiSkill [36] simulation to simulate manipulation failures. For real failure records already present in RoboFail [34], we directly extract and reformat them. All data types undergo human sampling verification, and the construction pipeline and QA template design are described in Appendix F.2.

E.1.4 Embodied Pointing Data

Pointing is the signature capability of Embodied-R1.5. We organize pointing data according to the four pointing capabilities defined in Section 2. Beyond extensively integrating existing open-source data, we also significantly expand data through automated pipelines in the directions of functional affordance and trajectory generation.

REG (Referring Expression Grounding). Referring expression grounding requires the model to precisely localize target objects based on natural language descriptions. We construct massive training data from multiple sources including RefCoco [100], SAM2 [101], Pixmo-Points [63], CoSyn-Point [102], LVIS [103], RoboRefit [59], and Ref-L4 [104].

RRG (Region Referring & Grounding). Region grounding is a core capability for object placement and spatial arrangement in embodied manipulation, requiring the model to identify suitable free regions for placement and assess placement feasibility. We integrate RoboPoint [10] covering diverse robotic manipulation pointing scenarios, RefSpatial [58] providing region-level spatial referring and grounding, and ER1-Point inherited and extended from Embodied-R1. Additionally, we extract large-scale region grounding data from massive human interaction videos, providing diverse real-world placement scenarios. Notably, in Embodied-R1.5 we generate Regular Rearrangement samples via synthetic simulation data (RegularRearrangement). These tasks present scenes with blocks arranged in regular patterns (triangles, squares, etc.) with 1–2 blocks removed; the model must reason about where to point to complete the pattern, combining spatial reasoning with pointing capability.

OFG (Object Functional Grounding). Functional grounding integrates visual localization with physical property understanding, requiring the model to localize functional parts of objects (e.g., cup handles, kettle spouts, tool grips), which is critical for correct grasping strategies in embodied manipulation. Since part-level annotation in the real world is extremely expensive, we construct OFG data from three directions: (1) open-source affordance datasets: HandAL [105] provides human hand-functional part interaction annotations, PACO-LVIS [106] provides part-level affordance annotations, and InstructPart [107] provides instruction-conditioned part grounding; (2) simulation-based automatic extraction: PRISM [39] renders diverse articulated objects and automatically generates functional grounding annotations, and PartNet-Maniskill extracts part-level data covering 17 categories of articulated objects from ManiSkill [36]; (3) human interaction data: we extract human grasping patterns on functional parts from large-scale hand-object interaction datasets as supervision signals. See Appendix F.3 for pipeline details.

VTG (Visual Trace Generation). Trajectory prediction is crucial for internal planning in embodied tasks, depicting expected motion paths of objects or end-effectors as 2D or 3D feasibility curves. We automatically extract trajectory annotations from large-scale self-constructed simulation data and human interaction data, covering two trajectory types: robot-centric traces (describing end-effector motion paths) and object-centric traces (describing the motion trajectories of manipulated objects). See Appendix F.3 for pipeline details.

E.1.5 General Knowledge Data

To maintain general vision-language capability and prevent catastrophic forgetting, we include general knowledge and reasoning samples from LLaVA [108], HONEY [45], MM-IF [46], and EUCLID [44], covering diverse VQA, captioning, reasoning, and multi-modal instruction following tasks.

F Data Construction Pipeline Details

This appendix provides full details of the three automated data construction pipelines described in Sec. 3. Sec. F.1 describes the 3D scene annotation pipeline for spatial reasoning data (ER1.5-Spatial). Sec. F.2 describes the failure-aware data construction pipeline for correction data (ER1.5-Correction). Sec. F.3 describes the functional affordance and trajectory data construction pipelines (ER1.5-Pointing).

F.1 Pipeline 1: 3D Scene Annotation for Spatial Reasoning Data

This section details the automated 3D scene annotation pipeline used to construct the **ER1.5-Spatial** dataset ($\sim$20K samples). The pipeline takes single RGB images as input and produces structured 3D

semantic scene graphs, from which tabletop-level spatial reasoning QA pairs are programmatically generated.

While several open-source 3D datasets provide geometric, semantic, and instance-level metadata — including ScanNet [109], ScanNet++ [92], and ARKitScenes [110] — these datasets predominantly operate at room-level granularity typically used for navigation scenarios, which is complementary to our data. Our goal is to construct spatial reasoning QA data at a finer *tabletop manipulation* granularity: given an RGB image of a tabletop operation scene, automatically infer the complete 3D scene information (object categories, spatial positions, inter-object relations) and generate spatial reasoning QA training data. Early efforts create small-scale spatial QA datasets via manual or semi-automated annotation, an approach that does not scale. Our key insight is that once a detailed 3D semantic scene graph can be automatically reconstructed from a single RGB image, large-scale spatial reasoning QA data can be generated programmatically.

Data source. Our input images are drawn from large-scale real-robot datasets in Open X-Embodiment [111], including **BridgeData V2** [28] and **DROID** [29], which provide diverse tabletop operation RGB images spanning varied object compositions, table layouts, and lighting conditions. We sample representative frames from these datasets as pipeline input. Let the input image set be $\mathcal{I} = \{I_i\}_{i=1}^{N}$. The pipeline processes each image independently and outputs a structured annotation record $\mathcal{S}_i$ containing: semantic instance labels $\{c_k\}$, 2D instance masks $\{M_k\}$, a world-frame point cloud $\mathbf{P}_i^w$, per-instance 3D bounding boxes $\{B_k\}$, and the camera-to-world transformation $\mathbf{T}_i$. The pipeline proceeds through multiple stages: semantic understanding, geometry estimation with 2D segmentation, 3D lifting, and coordinate normalization. Quality control mechanisms are embedded within each stage.

Semantic understanding. Before geometric processing, the pipeline extracts object category labels for each image. The input image is first resized so that both height and width are multiples of 14, satisfying the patch-based tokenizer constraint of downstream ViT models. We employ two complementary semantic annotation backends (used solely as annotation tools, distinct from the model being trained): **Qwen3-VL** leverages the open-domain understanding capability of a multimodal large language model to generate rich semantic descriptions, particularly effective for long-tail objects and functional characterizations; **RAM++** [112] provides broader category coverage and higher recall at faster inference speed. The raw outputs of both backends are passed through a VLM-based *semantic normalization module* that identifies and merges synonym groups (e.g., "cup" and "mug"), removes redundant categories, and produces a normalized global label table $\mathcal{L} = \{(l_j, \mathrm{id}_j)\}$ together with a text prompt $\mathcal{T}$ that conditions the subsequent detection and segmentation models. This normalization step is critical for reducing spurious detections in the downstream segmentation stage. As quality control at the semantic stage, we perform multi-frame consistency checks and filter out images with fewer than 2 valid detected objects (insufficient for generating spatial relation QA), as well as scenes where semantic ambiguity is detected or the robot arm itself is misidentified as a manipulable object.

Geometry estimation and 2D instance segmentation. Building on the semantic labels, we perform geometry estimation and instance segmentation in parallel. For geometry estimation, we use the **metric-scale variant of MoGe-2** [30] to jointly predict, from a single RGB image: a dense depth map $D_i \in \mathbb{R}^{H\times W}$ in absolute metric units (meters), a per-pixel surface normal map $\mathbf{N}_i \in \mathbb{R}^{H\times W\times 3}$, and camera intrinsics $\mathbf{K}_i \in \mathbb{R}^{3\times 3}$. Since MoGe-2 directly outputs absolute-scale depth values, the subsequent 3D reconstruction and distance computation require no additional scale recovery step — this is the key prerequisite enabling programmatic generation of distance estimation QA data. Through extensive experimentation, we found that MoGe-2 consistently outperforms Depth Anything V3 in reconstruction accuracy on tabletop-level scenes. For instance segmentation, we adopt **Grounded-SAM-2** [31] for open-vocabulary segmentation: GroundingDINO generates candidate detection boxes $\{b_k\}$ conditioned on the text prompt $\mathcal{T}$, and SAM2 refines each detection box into a precise binary mask M_k, yielding the complete per-image instance annotation $\{(M_k, c_k, \mathrm{conf}_k)\}$. Compared to using a single global prompt, conditioning on per-image labels significantly reduces false positives from categories absent in the scene. At this stage, we filter out instances with detection confidence below 0.3, as well as instances whose mask area is less than 0.5% or greater than 50% of the image area (the former are typically noise detections; the latter are typically background misdetections such as the table surface itself). Fig. 8 illustrates the geometry estimation outputs on a tabletop scene.

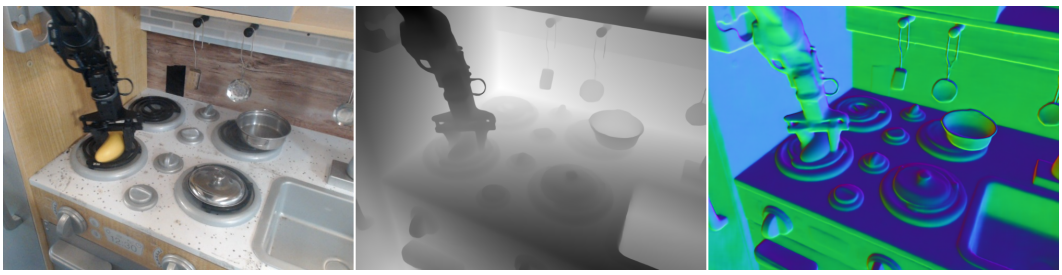


Figure 8: **Geometry estimation.** MoGe-2 metric-scale depth and surface normal predictions on a BridgeData V2 tabletop scene. From left to right: input RGB, predicted depth map, and predicted surface normals.

3D lifting (inverse perspective projection). Using the depth map D_i, intrinsic matrix $\mathbf{K}_i$, and instance masks $\{M_k\}$, each foreground pixel (u, v) belonging to instance k is lifted to a 3D point in the camera coordinate frame via inverse perspective projection:

$$\mathbf{p}^c = D_i(u, v) \cdot \mathbf{K}_i^{-1}[u,\ v,\ 1]^\top. \tag{3}$$

To suppress depth discontinuity artifacts at object boundaries, we apply edge-aware filtering that removes pixels whose depth gradient exceeds 3 standard deviations of the local neighborhood mean, and perform voxel downsampling (voxel size 2 mm) to control point cloud size. Additionally, we apply Statistical Outlier Removal (SOR) to each instance's 3D point cloud, ensuring reliable downstream bounding box estimation.

Horizontal plane alignment. In tabletop manipulation scenarios, the dominant plane is the table surface. We transform the camera coordinate frame to a world coordinate frame in which the table surface lies at $z = 0$ and the z-axis points vertically upward. First, we estimate the dominant plane normal $\hat{\mathbf{n}}_{\text{best}}$ via RANSAC on the per-pixel normal map $\mathbf{N}_i$: a seed normal $\vec{n}_{\text{seed}}$ is randomly sampled, and all normals with dot product $S_j = \vec{n}_{\text{seed}} \cdot \vec{n}_j > 0.996$ (angular deviation $< 5°$) are counted as inliers; this is repeated for $K = 100$ iterations, selecting the estimate with the largest inlier count. As a quality check, if the best inlier ratio is below 30%, we deem the scene to lack a clear dominant plane (e.g., a cluttered non-tabletop scene) and discard it. Second, we determine the table height via a projection histogram: all camera-frame points are projected onto $\hat{\mathbf{n}}_{\text{best}}$ to obtain $d_j = \hat{\mathbf{n}}_{\text{best}} \cdot \mathbf{p}_j^c$, and the histogram peak d_{peak} identifies the table surface height (Fig. 9a–b). Third, we construct the 4×4 camera-to-world transformation: setting $\mathbf{z}_{\text{new}} = \hat{\mathbf{n}}_{\text{best}}$, we build the remaining axes via Gram-Schmidt orthogonalization to obtain the rotation matrix $\mathbf{R} = [\mathbf{x}_{\text{new}},\ \mathbf{y}_{\text{new}},\ \mathbf{z}_{\text{new}}]^\top$, set the translation $\mathbf{t} = [0,\ 0,\ -d_{\text{peak}}]^\top$, and form:

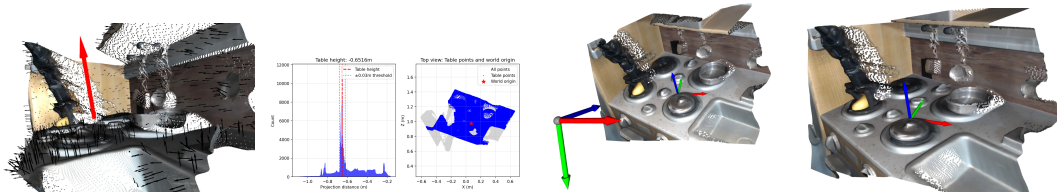
$$\mathbf{T}_{c\to w} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}. \tag{4}$$

The transformation is applied to all point clouds ($\mathbf{P}_i^w = \mathbf{T}_{c\to w} \cdot \mathbf{P}_i^c$) and camera poses, ensuring that the z-axis of all processed images is consistently aligned with the scene's gravity direction (Fig. 9c–d). After transformation, we further verify that the z-coordinates of most object bounding box bottoms lie within 7 cm of $z = 0$; scenes that fail this sanity check are flagged as low-quality and removed.

Spatial reasoning QA generation. All intermediate outputs are aggregated into a unified per-scene metadata record $\mathcal{S}_i$ containing: image, normalized semantic label table, per-instance 2D masks and 3D point cloud segments, estimated intrinsics, per-instance 3D bounding boxes, and the world-coordinate transformation matrix. Leveraging this dense spatial annotation, we programmatically generate QA pairs covering key task types including spatial relations, distance metrics, scene cognition, and appearance order, constituting the tabletop-level spatial reasoning QA dataset ER1.5-Spatial (∼20K samples) that directly feeds into the training data system described in Sec. 3.

F.2 Pipeline 2: Failure-Aware Data Construction for Correction Data

This section details the systematic construction of the **ER1.5-Correction** dataset (∼800K+ samples), a large-scale failure-aware QA dataset designed to endow the embodied foundation model with failure



(a) RANSAC normals (b) Height histogram (c) World frame axes (d) Aligned point cloud

Figure 9: **Horizontal plane alignment (Example 1).** (a) RANSAC identifies the dominant plane normal from the predicted normal map. (b) The projection histogram determines the table surface height d_{peak}. (c) The constructed world coordinate frame with the z-axis aligned to gravity. (d) The resulting point cloud after transformation, with the table surface at $z = 0$.

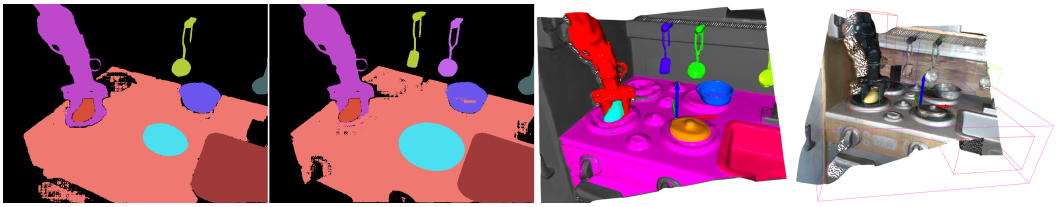


Figure 10: **Instance segmentation and 3D scene reconstruction.** From left to right: raw instance segmentation masks, fused instance masks, per-instance 3D point clouds with semantic coloring, and fitted 3D bounding boxes in the gravity-aligned world frame.

perception and autonomous correction capabilities throughout the robot task execution lifecycle. Motivated by the failure taxonomy frameworks of RoboFAC [35] and Guardian [33], the dataset covers both simulation and real-world scenarios, organizing failures along two orthogonal dimensions: by execution stage and by cognitive level.

F.2.1 Failure Taxonomy

We organize robot task execution failures along two orthogonal dimensions: by *stage* into planning failures and execution failures, and by *cognitive level* into detection, localization, and correction.

Planning failures. These originate from inaccurate task decomposition or language instruction misunderstanding, and include five types:

- **Step omission:** A necessary subtask is skipped, resulting in an incomplete plan.
- **Step redundancy:** An unnecessary duplicate subtask is introduced.
- **Step swap:** The execution order of two subtasks is reversed.
- **Object error:** The manipulation target in a subtask is replaced with another object present in the scene.
- **Action/location error:** The action verb or spatial location descriptor in a subtask is altered to an incorrect alternative.

Execution failures. These originate from insufficient spatial reasoning, instruction-to-action mapping errors, or physical execution imprecision, and include four types:

- **Execution interruption:** The robot completes only a subset of subtasks before halting.
- **Wrong manipulation object:** The robot manipulates a non-target object in the scene.
- **Wrong action:** The robot executes an action that does not match the instruction.
- **Operation failure:** The robot attempts the correct action but fails due to physical reasons (e.g., object slipping during grasp).

Three cognitive levels. For each failure type above, we construct QA at three progressive levels:

- **Failure detection:** Binary judgment (yes/no) of whether the plan or execution is correct.
- **Failure localization:** Identification of the specific failure type and the location (which subtask) where it occurs.
- **Failure correction:** Generation of a repair plan that transitions from the erroneous state to the correct one.

F.2.2 Data Sources

We collect raw data from multiple complementary sources covering both simulation and real-world scenarios.

For simulation, **RoboFail** [34] provides multi-type operation failure execution videos across 10 simulated tasks with structured failure annotations (task name, action sequence, failure cause, temporal location), directly usable for constructing execution-stage failure QA. On the **ManiSkill** [36] platform, we design 26 perturbation schemes across 11 tasks; by injecting directed perturbations during execution (e.g., opposing forces during approach, positional offsets during grasping), we generate $\sim$3.5K failure samples (2,560 planning failures, 740 execution failures, 220 correct samples). **GEMBench** [37] covers 31 simulated tasks (16 task types) with keyframe images, task instructions, and subtask lists, used for planning-stage failure QA.

For real-world data, **RoboFail** [34] additionally provides 30 failure episodes across 10 real-world tasks with complete execution videos and JSON-formatted structured failure descriptions. **BridgeData V2 (CoT-annotated version [99])** contains over 60K real-world robot execution episodes, each with complete image sequences, language instructions, per-timestep action annotations, scene object detection results, and step-by-step reasoning information. **RoboFAC** [35] provides failure execution videos across 6 real-world tasks with QA annotations for failure detection, determination, and localization.

F.2.3 Planning Failure Data Construction

The core idea is to apply structured perturbations to correct subtask plans. The input consists of an initial-frame image and a language instruction, with the correct subtask plan $\mathcal{P} = \{s_1, s_2, \ldots, s_n\}$ obtained from dataset annotations or VLM generation. We construct planning failures via five perturbation operators:

- **Step deletion $\mathcal{D}_{\text{del}}$:** Randomly select subtask s_k and remove it: $\mathcal{P}' = \mathcal{P} \setminus \{s_k\}$.
- **Step duplication $\mathcal{D}_{\text{dup}}$:** Randomly select subtask s_k and insert a duplicate: $\mathcal{P}' = \{s_1, \ldots, s_k, s_k, \ldots, s_n\}$.
- **Step swap $\mathcal{D}_{\text{swap}}$:** Randomly select two subtasks s_i, s_j $(i \neq j)$ and exchange their positions.
- **Object replacement $\mathcal{D}_{\text{obj}}$:** Select a non-target object o' from the scene object list and replace the target object o in subtask s_k: $s'_k = \text{Replace}(s_k, o, o')$. The scene object list is obtained from bounding box detection results or object annotations.
- **Action/location replacement $\mathcal{D}_{\text{act}}$:** Using predefined action synonym dictionaries or spatial relation dictionaries, replace the action verb or location descriptor in subtask s_k with a semantically similar but functionally incorrect alternative.

For BridgeData, we leverage its rich step-by-step reasoning annotations and object detection results to precisely locate the manipulation target and action description in each subtask, enabling fine-grained perturbation construction. For ManiSkill and GEMBench, we perform perturbations based on task structure and keyframe information.

Each perturbation simultaneously generates three levels of QA: the detection QA reference answer is directly set to "No"; the localization QA reference answer is generated via template inversion of the perturbation operator (e.g., for $\mathcal{D}_{\text{swap}}$, the answer template is "The subtask 's_j' should be executed before 's_i'."); the correction QA reference answer contains the complete correct subtask sequence $\mathcal{P}$. To prevent overfitting to failure patterns, we simultaneously retain original correct plans as positive samples with detection QA (reference answer: "Yes").

F.2.4 Execution Failure Data Construction

The core idea is to exploit inconsistencies between video content and subtask instructions to construct failure samples. Depending on the data source, we employ three complementary strategies:

- **Truncation (execution interruption):** Original successful execution videos are truncated at a subtask completion boundary, so the input video contains only the first portion of task execution. The model must judge that the task is incomplete and identify the unexecuted subtasks. This method applies to all data sources with complete execution videos (BridgeData, ManiSkill).
- **Description replacement (object/action errors):** The language description of a subtask is modified in terms of the manipulation object or action verb, creating a mismatch with the actual video content. For object errors, semantic descriptions of other scene objects are used for substitution; for action errors, antonyms or near-synonyms of the action verb are used (e.g., "open" replaced with "close"). This method is primarily applied to BridgeData.
- **Perturbation injection (operation failure):** In ManiSkill simulation, physical perturbations are injected at specific subtask execution phases (e.g., external forces during grasping, directional offsets during movement) to produce execution failure videos caused by physical factors. Perturbation parameters are calibrated to ensure the robot exhibits "attempted but failed" behavior rather than completely unrelated random actions.

For RoboFail and RoboFAC datasets, which natively contain real failure execution records and structured failure annotations, we directly extract relevant information and reformat it into unified QA data. For RoboFail's real-world data, Qwen-VL-32B-generated reference answers are used for failure correction QA.

F.2.5 Quality Control

All QA templates and VLM-generated answers undergo human sampling review and correction. We implement the following additional quality control measures:

- For planning failure perturbations, we verify that the perturbed subtask sequence genuinely cannot complete the original task (e.g., omitting a critical step indeed makes the task unreachable).
- For execution truncation, we ensure the truncation point falls at a subtask boundary rather than mid-subtask execution, avoiding ambiguity.
- For description replacement, we ensure that the substituted object or action is semantically plausible but executionally incorrect (e.g., avoiding obviously unreasonable combinations like "use the cup to grasp the table").
- For all binary detection QA, we enforce positive/negative sample balancing to prevent the model from learning answer preferences.

F.2.6 Dataset Statistics

The final ER1.5-Correction dataset comprises $\sim$800K+ samples. BridgeData contributes $\sim$802K samples (covering 19 QA subtypes), ManiSkill contributes $\sim$3.5K samples (11 tasks, 26 perturbation schemes), RoboFail contributes failure annotations from both simulation and real-world scenarios, RoboFAC contributes failure QA from 6 real-world tasks, and GEMBench contributes planning failure data from 31 simulated tasks. While BridgeData contributes the majority of samples due to its rich annotation structure, failure type diversity is ensured by combining five perturbation operators with three execution failure strategies across both simulation and real-world scenarios. By cognitive level, the three tiers (failure detection, failure localization, and failure correction) are approximately balanced in sample count.

F.3 Pipeline 3: Functional Affordance & Trajectory Data Construction

This section describes two components of the third automated pipeline that generate functional affordance data (for OFG) and trajectory data (for VTG).

F.3.1 Object Functional Affordance Data

Object Functional Grounding (OFG) requires the model to localize functional parts of objects (handles, spouts, buttons, etc.) and understand their action affordances. Part-level annotation in the real world is extremely expensive, creating a bottleneck for scaling OFG data. We construct OFG data automatically from two complementary sources:

Simulation-based extraction. We leverage the advantage that part-level semantic annotations in simulation environments are essentially free. Using PRISM [39] and PartNet-Mobility assets in ManiSkill [36], we render diverse articulated objects from multiple viewpoints and lighting conditions, and automatically generate functional grounding annotations by mapping semantic part labels to image coordinates.

Human interaction data extraction. We extract functional part grounding from large-scale hand-object interaction datasets, where human grasping patterns naturally indicate functional affordance locations. By tracking contact regions between human hands and objects, we obtain supervision signals for where and how objects should be grasped for different manipulation intents.

F.3.2 Trajectory Data

Visual Trace Generation (VTG) generates ordered point sequences that describe complete manipulation trajectories, supporting both 2D and 3D trace formats. While many robot demonstration datasets contain action sequences, they lack the explicit 2D/3D visual trajectory annotations required for VTG training. We design automated trajectory extraction pipelines to bridge this gap. We distinguish *robot-centric traces* (end-effector flow), which describe how the robot should move, from *object-centric traces* (object flow), which describe the desired motion of the manipulated object. The following procedures construct these complementary forms of supervision.

End-effector trajectory extraction (with pose metadata). For robot demonstration datasets that record full 3D end-effector poses (e.g., DROID [29] and BridgeData V2 [28]), we leverage the camera intrinsic and extrinsic matrices provided by each dataset to project per-frame 3D end-effector TCP positions onto the 2D image plane via perspective projection. Specifically, for each episode we extract the TCP position sequence across keyframes, apply the camera coordinate transformation, and obtain a continuous 2D pixel coordinate trajectory. For 3D traces, we extract end-effector positions in world coordinates and normalize them relative to the workspace. After projection, we apply trajectory smoothing and outlier removal to ensure the resulting 2D traces are visually coherent and physically plausible.

End-effector trajectory extraction (without pose metadata). For datasets that do not provide 3D pose information, we train a dedicated end-effector detector based on Detectron2 [40], fine-tuned on OXE data to precisely localize the robot gripper in each frame. For each episode's frame sequence, the detector outputs per-frame gripper center coordinates, which are concatenated to form the 2D end-effector motion trajectory. To improve robustness, we introduce temporal consistency constraints and Kalman filter smoothing to handle occlusion and detection jitter.

Object motion tracking. For scenarios requiring object motion flow generation, we leverage advanced point tracking models such as Co-Tracker3 [41] to track manipulated object motion across video frames. The pipeline first identifies the target object region in the initial frame via semantic segmentation or detection, then uniformly samples tracking points on the object surface, propagates these points across frames using Co-Tracker3, and finally takes the centroid trajectory of all tracked points as the object motion flow annotation. This method is primarily applied to hand-object interaction datasets (e.g., HOI4D), capturing the complete motion path of objects during grasping, transport, and placement.

Articulated object 3D trajectories. For articulated object manipulation (e.g., opening doors, pulling drawers, turning faucets), we leverage the rich articulated object assets and simulation environments provided by RoboCasa [42] to generate 3D trajectory data. In simulation, both joint states and end-effector poses can be precisely obtained; we simultaneously record the end-effector path and object articulated motion trajectory during manipulation, retain the 3D trajectories, and project both onto the 2D image plane for 2D trace supervision. Since articulated object motion

patterns are strongly structured (along arcs or lines), such trajectory data is particularly valuable for helping the model understand kinematic constraints of articulated manipulation.

Both trace types undergo quality filtering (removing trajectories with excessive noise, insufficient motion, or physically implausible paths), coordinate normalization, and are paired with natural-language instructions describing the intended action. Data sources span robot demonstration and simulation datasets (e.g., DROID, GenManip [113], InternData-A1 [114]) and human interaction video datasets, collectively providing diverse motion patterns across manipulation primitives. The resulting ER1.5-Pointing dataset comprises approximately 400K samples covering affordance and trajectory annotations.

G Broader Impacts and Limitations

Broader Impacts. This work aims to build an embodied foundation model with strong reasoning capabilities, advancing robotics from task-specific data collection toward a unified model-driven paradigm. Positive impacts include: (1) lowering the barrier to robot deployment by reducing per-task data requirements through a unified model; (2) improving execution safety via closed-loop error correction, which reduces the need for human intervention upon failure; and (3) fully open-sourcing all models, data, code, and the evaluation framework to facilitate reproducible research in the community. Target application scenarios span household assistance, industrial support, and other beneficial directions.

Regarding potential risks, general-purpose manipulation capabilities could theoretically be applied to unintended scenarios. We mitigate this through the following measures: generated plans and correction messages can be inspected, and we release the model with usage guidelines. These outputs do not guarantee faithful access to internal computation or safe physical execution. Evaluation is restricted to tabletop manipulation and does not establish reliability outside the tested settings.

Limitations. This work has several limitations. First, the model operates on 2D image inputs via a VLM backbone and lacks native 3D perception (e.g., direct point cloud input), which may degrade spatial reasoning accuracy under severe occlusion. Second, the current design decouples VLM reasoning from VLA action generation; tighter end-to-end integration remains a promising direction for future work. Third, the PGC closed-loop framework has been validated on task horizons of 5 to 10 sub-steps, and its performance under significantly longer horizons requires further investigation. Finally, real-robot experiments are limited in scale due to hardware constraints. The PGC evaluation uses 10 trials per task and variant, and we do not report multi-seed confidence intervals. Recovery depends on perception, calibration, and low-level execution; detecting a failure does not guarantee that it can be physically corrected. Future work will extend evaluation to more diverse robot morphologies and larger-scale physical experiments.

H EmbodiedEvalKit

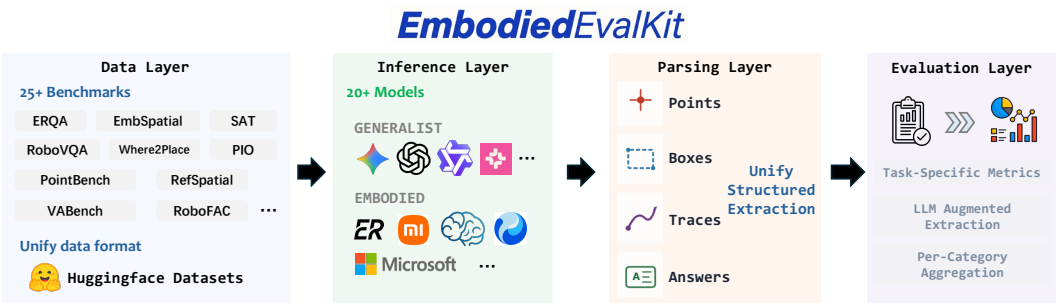


Figure 11: **Architecture of EmbodiedEvalKit.** A four-layer modular framework supporting 25+ embodied benchmarks and 20+ models in a single reproducible evaluation pipeline.

The general-purpose VLM community benefits from mature evaluation frameworks such as VLMEvalKit [115] and lmms-eval [116], yet the embodied VLM domain lacks comparable infrastructure. Each paper implements its own evaluation scripts with different grounding format

Table 10: **Full embodied VLM benchmark results: Cognition, Spatial Reasoning, and Planning.** Models are grouped into general-purpose VLMs, embodied VLMs, and ours (including ablation variants).

	Cognition & Spatial Reasoning									*Planning & Correction*				
Model	ERQA	OpenEQA	CV-B	EmbSp	SAT	RoboSp	BLINK	VSI	**Avg**	RoboVQA	EgoPl	Cosmos	RoboF	**Avg**
General-Purpose VLMs														
GPT-4o	32.5	51.1	78.6	71.9	66.7	44.4	64.5	43.6	56.7	34.5	41.8	53.3	48.9	44.6
GPT-5	54.5	64.4	84.5	78.3	83.3	54.1	75.0	52.9	68.4	33.1	36.9	56.2	69.0	48.8
GPT-5.4	50.5	69.3	76.7	73.2	74.7	53.1	87.9	52.6	67.3	31.8	44.1	69.2	70.1	53.8
Gemini-2.5-Pro	55.7	65.8	84.6	78.7	76.7	59.9	85.5	43.4	68.8	33.9	42.9	62.8	46.4	46.5
Gemini-Robotics-ER-1.5	48.5	50.5	83.6	73.4	62.7	40.1	85.9	39.9	60.6	32.6	32.0	53.7	47.0	41.3
Qwen2.5-VL-7B	39.3	22.0	73.8	70.3	52.8	44.3	75.0	37.3	51.8	52.8	39.7	53.7	53.2	49.9
Qwen3-VL-8B	41.8	65.1	86.3	78.5	65.3	65.4	78.2	55.1	67.0	55.0	32.3	64.0	65.4	54.2
InternVL3.5-8B	41.0	42.9	81.5	70.3	55.3	51.1	75.0	55.7	59.1	28.6	31.6	48.2	48.6	39.2
Molmo-D-7B	40.6	35.1	67.9	58.7	54.0	36.0	75.0	6.7	46.8	23.0	26.9	22.2	11.5	20.9
Embodied VLMs														
RoboBrain-2.0-7B	38.5	24.2	85.8	76.3	75.3	54.2	79.0	36.1	58.7	57.5	33.2	33.8	54.2	44.7
VeBrain	37.3	27.8	79.7	70.5	58.0	42.5	71.8	36.2	53.0	42.4	27.3	58.8	58.8	46.8
Mimo-Embodied	46.8	53.6	88.8	76.2	54.6	61.8	94.4	48.5	65.6	62.0	43.0	56.8	54.6	54.1
Pelican-VL-1.0	39.8	54.7	78.9	73.2	54.6	57.5	78.2	52.8	61.2	58.5	32.0	62.6	60.2	53.3
Magma	25.4	32.7	65.9	64.6	71.3	33.7	58.9	17.1	46.2	14.3	4.1	32.8	18.1	17.3
Embodied-R1	35.2	34.5	82.7	67.4	76.3	47.4	76.6	26.6	55.8	51.8	26.5	55.2	40.2	43.4
Ours														
Embodied-R1.5-SFT	42.3	63.0	86.5	78.9	76.7	68.3	83.1	52.6	68.9	61.6	49.9	62.2	76.8	62.6
Embodied-R1.5-GRPO	44.3	62.8	86.1	77.3	79.3	67.7	90.3	55.2	70.4	60.9	54.4	67.3	76.6	64.8
Embodied-R1.5-EMA-GRPO	44.5	62.6	85.6	76.6	76.7	66.9	86.3	56.1	69.4	60.2	50.7	62.2	77.2	62.6
Embodied-R1.5-No-Filter	42.8	62.6	85.6	77.3	74.7	67.1	85.5	55.8	68.9	60.3	51.6	66.2	75.4	63.3
Embodied-R1.5	46.0	62.6	86.9	78.1	74.7	69.7	87.1	56.1	70.2	61.0	53.8	69.3	77.2	65.3

conventions and often evaluates on non-overlapping benchmark subsets, making cross-paper comparison unreliable. Moreover, evaluation data is scattered across disparate sources, making it laborious to set up a comprehensive evaluation system. To bridge this gap, we develop and open-source **EmbodiedEvalKit**, a unified evaluation framework designed for embodied VLMs.

As shown in Fig. 11, EmbodiedEvalKit adopts a four-layer modular design. The *data layer* reorganizes 25+ embodied benchmarks spanning all three capability dimensions into a standardized HuggingFace Parquet format, enabling a single data loading pipeline for all tasks. The *inference layer* provides a model-agnostic interface supporting vLLM [117], HuggingFace Transformers, and API backends, with a *backbone* abstraction that automatically handles model-specific differences in coordinate systems (e.g., 0–1000 normalized vs. absolute pixel vs. 0–1 normalized) and prompt templates, allowing both open-source and proprietary models to be evaluated within the same framework. The *parsing layer* detects and normalizes heterogeneous grounding outputs (points, bounding boxes, trajectories, free-form text) into a unified structured representation. Finally, the *evaluation layer* computes official metrics for each benchmark and aggregates results by capability category. EmbodiedEvalKit is publicly available and extensible: adding a new benchmark requires only defining a data adapter and an evaluator following standardized templates. We release pre-processed evaluation data for all supported benchmarks to ensure exact reproducibility.

I Additional Experimental Results

I.1 Full Model Comparison on Embodied VLM Benchmarks

Tabs. 10 and 11 present the complete comparison across all evaluated models, including additional baselines not shown in the main paper: GPT-4o, GPT-5, Qwen2.5-VL-7B, InternVL3.5-8B, Molmo-D-7B, VeBrain, Magma, and the SFT-only variant of Embodied-R1.5. Models are grouped into general-purpose VLMs, embodied VLMs, and ours (including ablation variants).

I.2 Full Results on ManiSkill-PartNet Benchmark

Tab. 12 presents the complete per-category results on ManiSkill-PartNet, covering 20 train and 10 test object categories.

Table 11: **Full embodied VLM benchmark results: Pointing & Location.** Models are grouped into general-purpose VLMs, embodied VLMs, and ours (including ablation variants).

	Pointing & Location									
Model	VAbench-P	Where2Place	RefSpatial	Part-Afford	RoboRefit	RoboAfford	PointBench	PIO	Pixmo-Pt	**Avg**
General-Purpose VLMs										
GPT-4o	13.7	20.4	8.8	13.3	14.2	20.5	29.5	13.4	13.5	16.4
GPT-5	31.1	39.7	21.6	16.9	44.7	30.0	49.2	23.6	23.6	31.2
GPT-5.4	25.7	34.7	15.7	17.2	29.2	29.4	37.9	19.3	27.5	26.3
Gemini-2.5-Pro	21.7	49.6	36.5	25.5	38.4	23.4	62.8	23.2	12.7	32.6
Gemini-Robotics-ER-1.5	41.6	48.3	39.7	27.0	80.0	57.6	70.7	54.0	52.5	52.4
Qwen2.5-VL-7B	25.5	31.1	38.0	42.7	74.9	60.7	54.0	48.3	54.9	47.8
Qwen3-VL-8B	42.3	64.0	42.2	48.7	83.6	69.9	65.6	56.8	59.9	59.2
InternVL3.5-8B	23.0	34.8	16.8	22.7	30.8	31.5	32.4	23.1	44.1	28.8
Molmo-D-7B	38.0	39.3	45.4	73.2	81.1	61.7	61.6	44.3	51.9	55.2
Embodied VLMs										
RoboBrain-2.0-7B	41.0	63.6	32.5	45.6	70.4	51.5	60.4	54.0	54.7	52.6
VeBrain	1.7	12.3	0.3	32.1	32.2	2.1	20.2	24.1	16.8	15.8
Mimo-Embodied	46.9	63.6	48.0	65.5	82.3	69.8	50.2	27.5	42.4	55.1
Pelican-VL-1.0	14.5	57.8	37.5	28.8	74.9	63.4	36.4	32.9	26.1	41.4
Magma	6.6	10.9	4.5	0.3	5.1	13.4	23.4	4.6	12.8	9.1
Embodied-R1	66.0	69.5	39.7	56.6	85.6	67.2	49.7	44.4	49.4	58.7
Ours										
Embodied-R1.5-SFT	62.3	75.0	46.6	77.4	87.0	79.5	71.5	57.8	63.5	69.0
Embodied-R1.5-GRPO	70.7	65.0	50.9	82.3	89.4	78.9	72.8	59.1	51.4	68.9
Embodied-R1.5-EMA-GRPO	73.7	71.0	52.4	82.8	87.5	75.9	74.9	60.3	60.3	71.0
Embodied-R1.5-No-Filter	71.3	68.0	53.1	82.4	87.6	78.8	74.6	59.6	60.1	70.6
Embodied-R1.5	73.7	74.0	54.2	82.9	88.4	80.0	74.6	62.6	64.8	72.8

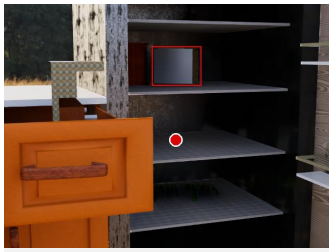
Table 12: **ManiSkill-PartNet: Full per-category results.** Success rate on 20 seen (train) and 10 unseen (test) articulated object categories. **Bold**: best result per category.

	Seen Categories (Train)															
Method																
Where2Act [72]	0.26	0.36	0.19	0.27	0.23	0.11	0.15	0.47	0.14	0.24	0.13	0.12	0.56	0.68	0.07	0.40
Implicit3D [73]	0.53	0.58	0.35	0.55	0.28	0.66	0.58	0.51	0.52	**0.57**	0.45	0.34	0.41	0.54	0.39	0.43
ManipLLM [15]	0.68	0.64	0.36	0.77	0.43	0.62	0.65	**0.61**	0.65	0.52	0.53	0.40	**0.64**	0.71	**0.60**	**0.64**
Embodied-R1.5	**0.96**	**0.92**	**0.92**	**0.88**	**0.90**	**0.80**	**0.96**	0.59	**0.94**	0.33	**1.00**	**0.50**	0.28	**1.00**	0.58	0.62

	Seen (cont.)					**Unseen Categories (Test)**										
Method					**Avg$_{seen}$**											**Avg$_{unseen}$**
Where2Act [72]	0.13	0.18	0.13	0.40	0.26	0.18	**0.35**	0.38	0.28	0.05	0.21	0.17	0.20	0.15	0.15	0.21
Implicit3D [73]	0.27	0.65	0.20	0.33	0.46	0.45	0.17	0.80	0.53	0.15	0.69	0.41	0.31	0.30	0.31	0.41
ManipLLM [15]	0.41	**0.75**	0.44	0.67	0.59	0.38	0.22	**0.81**	**0.86**	0.38	**0.85**	0.42	0.83	0.26	0.38	0.54
Embodied-R1.5	**0.82**	0.68	**0.47**	**1.00**	**0.76**	**0.67**	0.25	0.73	0.76	**0.94**	0.22	**0.92**	**0.90**	**0.69**	**0.53**	**0.66**

J Qualitative Visualizations

This section provides qualitative visualizations of Embodied-R1.5: embodied pointing across diverse scenes (Fig. 12), embodied spatial reasoning (Fig. 13), and generated chain-of-thought responses across pointing, planning, correction, and action understanding (Figs. 14 and 15). The reasoning examples show intermediate explanations followed by task-specific outputs within `<answer>` tags. Zero-shot RoboTwin manipulation and recovery under human perturbation are presented separately in Secs. K and L.



Question: Locate one spot in the vacant space below the item that has been highlighted.



Question: Locate at least a single point on an item that is positioned in front of the item that is bordered in red.



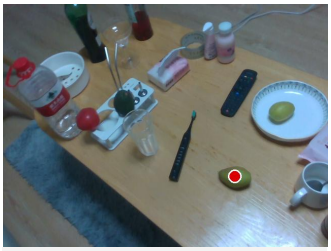
Question: Locate one point located upon an item that is resting on the marked plane.



Question: Locate several points situated on an item that rests upon the outlined plane.



Question: The task instruction is: put beet in pot sink. Use 2D points to mark the target location.



Question: Provide one point of objects region this sentence describes: pick up the small green pear at the bottom.



Question: Provide one point coordinate of objects region this sentence describes: pass me the remote control.



Question: The task instruction is: put lid on pot or pan. Use 2D points to mark the target location.

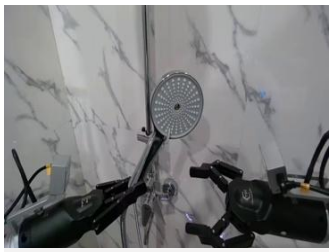


Question: The task instruction is: move blue spoon on the other side of pan on table. Use 2D points to mark the target location.

Figure 12: **Pointing examples of Embodied-R1.5.** Qualitative visualizations covering referring expression grounding, region-level free-space identification, and functional part localization across diverse manipulation scenes.



Question: What elements are present in the current visual?
Answer: The robotic arm holds the showerhead near its holder mounted on the tiled wall. The holder is positioned slightly above and to the right of the arm's grip.



Question: Detail the current scene you are perceiving.
Answer: The robotic arm holds the showerhead near its wall-mounted holder. The holder is attached to a marble wall, positioned slightly above the arm's grip.



Question: Could you describe the scene that is currently visible?
Answer: The robotic arm holds the showerhead near the wall-mounted holder. The holder is attached to the tiled wall, ready to receive the showerhead.



Question: What activity did you just finalize?
Answer: pick up the whole wheat toast from the slot after taking the white toaster.



Question: What activity did you just finalize?
Answer: pick up the whole wheat toast from the front card slot green toaster with left arm.



Question: What activity did you just finalize?
Answer: place the toast on the plate.



Question: If bowl was smaller, how must cup insertion strategy change?
Answer: Robot must angle the cup carefully to fit within the smaller bowl's dimensions.



Question: If the cup were glass, how would gripping strategy change?
Answer: Glass requires gentler grip to avoid slipping or breaking compared to ceramic.

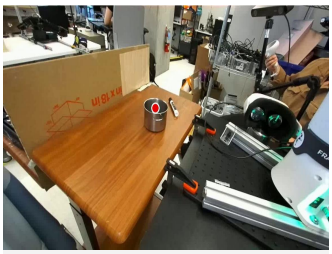


Question: If bowl was smaller, how must cup insertion adjust?
Answer: Cup insertion would need precise alignment and reduced depth to fit within the smaller bowl.

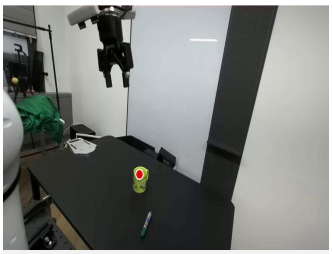
Figure 13: **Spatial reasoning examples of Embodied-R1.5.** Qualitative visualizations of multi-view spatiotemporal reasoning, depth and metric understanding, and object-level spatial relation reasoning under diverse embodied scenes.



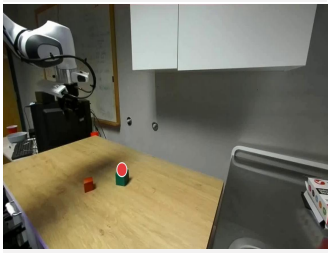
Question: You are currently a robot performing robotic manipulation tasks. The task instruction is: move the blue cube on top of yellow cube. Use 2D points to mark the target location where the object you need to manipulate in the task should ultimately be moved.
Answer: To complete the task, I need to move the blue cube to the top of the yellow cube. The current position of the blue cube is above the red cube, so I will adjust its position to align with the yellow cube.
<answer>[{\"point_2d\": [750, 748]}]</answer>



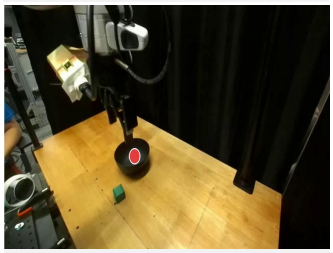
Question: You are currently a robot performing robotic manipulation tasks. The task instruction is: Place the black marker inside the pot. Use 2D points to mark the target location where the object you need to manipulate in the task should ultimately be moved.
Answer: I need to place the black marker inside the pot. I will move the marker towards the pot and position it inside.
<answer>[{\"point_2d\": [467, 419]}]</answer>



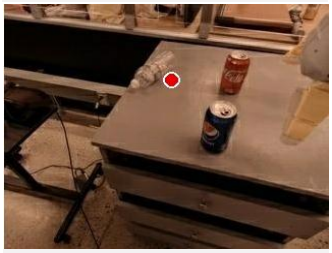
Question: You are currently a robot performing robotic manipulation tasks. The task instruction is: Put the green marker inside the yellow mug. Use 2D points to mark the target location.
Answer: To complete the task of putting the green marker inside the yellow mug, I need to identify the current position of the green marker and determine its target location within the mug. The green marker is currently outside the mug, so I will move it towards the mug's opening.
<answer>[{\"point_2d\": [418, 690]}]</answer>",



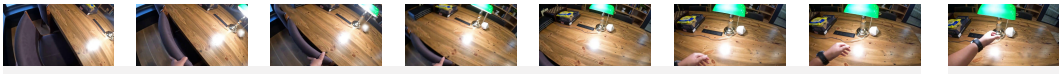
Question: You are currently a robot performing robotic manipulation tasks. The task instruction is: Place the orange block on the green one. Use 2D points to mark the target location.
Answer: To complete the task of placing the orange block on the green one, I need to identify the current position of the orange block and determine its target position above the green block. The green block is already positioned on the table, so the orange block needs to be moved to rest on top of it.
<answer>[{\"point_2d\": [367, 674]}]</answer>",



Question: You are currently a robot performing robotic manipulation tasks. The task instruction is: Put the green block in the black bowl. Use 2D points to mark the target location where the object you need to manipulate.
Answer: To complete the task of putting the green block in the black bowl, I need to identify the current position of the green block and determine its target location within the bowl. The green block is currently outside the bowl, so my next step is to move it towards the bowl's interior.
<answer>[{\"point_2d\": [401, 619]}]</answer>",

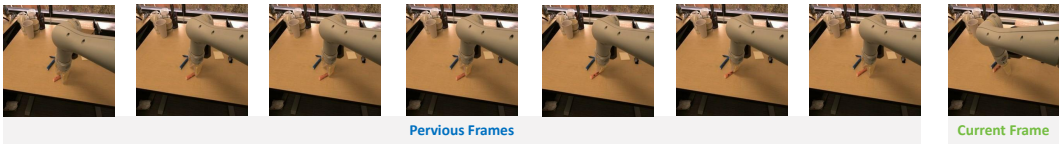


Question: You are currently a robot performing robotic manipulation tasks. The task instruction is: move pepsi can near water bottle. Use 2D points to mark the target location.
Answer: To complete the task of moving the Pepsi can near the water bottle, I need to identify the current position of the Pepsi can and determine a suitable target location close to the water bottle. The water bottle is located at a specific position on the table, and I will calculate a point that is adjacent to it.
<answer>[{\"point_2d\": [512, 308]}]</answer>",

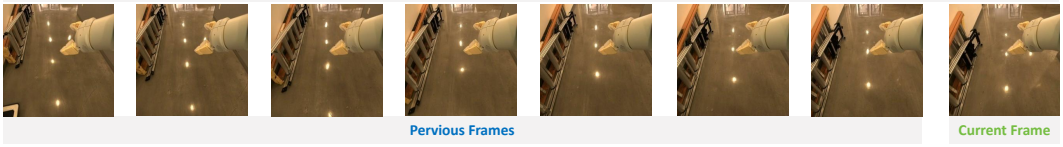


Question: The agent in the video was given the instruction - switch off the lamp. Is it possible for the agent to execute the task specified in the instruction?
A. no B. yes
Answer: To turn off the lamp, I need to locate the switch. The switch is on the lamp's base, and I am currently reaching my hand toward the switch, so I can turn it off.
<answer>B</answer>

Figure 14: **Chain-of-thought reasoning examples of Embodied-R1.5 (1/2).** Pointing over single-image manipulation scenes and feasibility planning over egocentric video, with the final decision emitted within an `<answer>` tag.



Question: place orange marker into pen holder. is it satisfied?
Answer: I need to check if the orange marker is placed into the pen holder. Observing the image, the orange marker is not inside the pen holder; it's still on the table. Therefore, the condition of placing the orange marker into the pen holder is not satisfied.
<answer>no</answer>



Question: what just happened?
Answer: I observed that the robot arm moved forward along a straight path while keeping its hand extended towards the ground. This indicates that the robot was advancing in a linear direction.
<answer>move forward</answer>

Figure 15: **Chain-of-thought reasoning examples of Embodied-R1.5 (2/2).** Task-state verification and action understanding over observation frame sequences, with the final decision emitted within an `<answer>` tag.

K Zero-Shot RoboTwin Manipulation

We evaluate Embodied-R1.5 on seven RoboTwin2.0 [118] tasks under the `demo_randomized` configuration. No RoboTwin data are used in any training stage. The model predicts pointing locations, which are converted into robot actions through a shared motion-execution interface. In contrast, RDT, π_0, and $\pi_{0.5}$ are fine-tuned with 400 demonstrations per task. This comparison assesses transfer under different training regimes, rather than performance under matched training data.

Table 13: **RoboTwin2.0 results.** Success rate (%) on seven tasks. Baselines use 400 demonstrations per task; Embodied-R1.5 uses no RoboTwin training data.

	Fine-tuned			**Zero-shot**
Task	**RDT**	$\boldsymbol{\pi_0}$	$\boldsymbol{\pi_{0.5}}$	**Embodied-R1.5**
Click Bell	9	3	66	99
Click Clock	12	11	89	56
Move Stapler	0	2	42	42
Place Mouse	0	1	39	52
Shake Bottle	45	60	97	89
Move Card	11	22	84	67
Place Shoe	7	6	93	50
Average	12.0	15.0	72.9	65.0

Embodied-R1.5 achieves 65.0% average success, compared with 72.9% for fine-tuned $\pi_{0.5}$, 15.0% for π_0, and 12.0% for RDT. Its 99% success on Click Bell illustrates effective functional grounding, while lower results on tasks such as Place Shoe indicate remaining task-dependent limitations. These results provide complementary evidence that pointing can support zero-shot manipulation transfer. Fig. 16 shows qualitative examples.

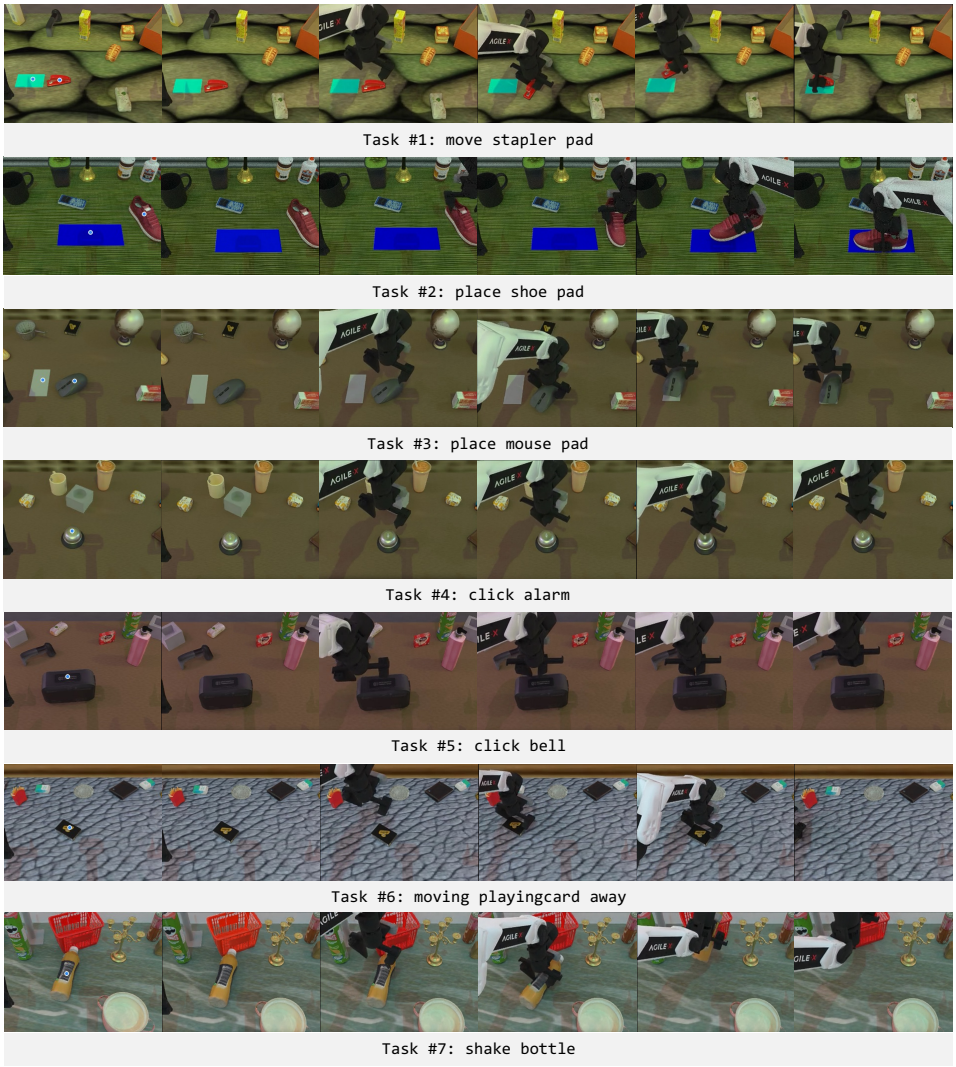


Figure 16: **Zero-shot RoboTwin manipulation examples.** Predicted pointing locations guide motion execution across manipulation tasks. These examples illustrate grasp, placement, and functional-part grounding without RoboTwin training data.

L Recovery under Human Perturbation

We further illustrate PGC's correction behavior under active human disturbances. During a pick-and-place task, a human operator relocates the destination plate and displaces the target object during execution. The Corrector identifies the deviation from the expected state, and the Grounder re-localizes the relevant object and destination before execution resumes. The demonstrated sequence recovers after repeated disturbances and completes the task without a manual reset (Fig. 17). Human interventions here deliberately perturb the scene rather than supply corrective commands.

This qualitative demonstration illustrates the recovery process underlying PGC. It is separate from the three-task quantitative evaluation in Tab. 9; we do not infer an aggregate success rate under human perturbation from this example.

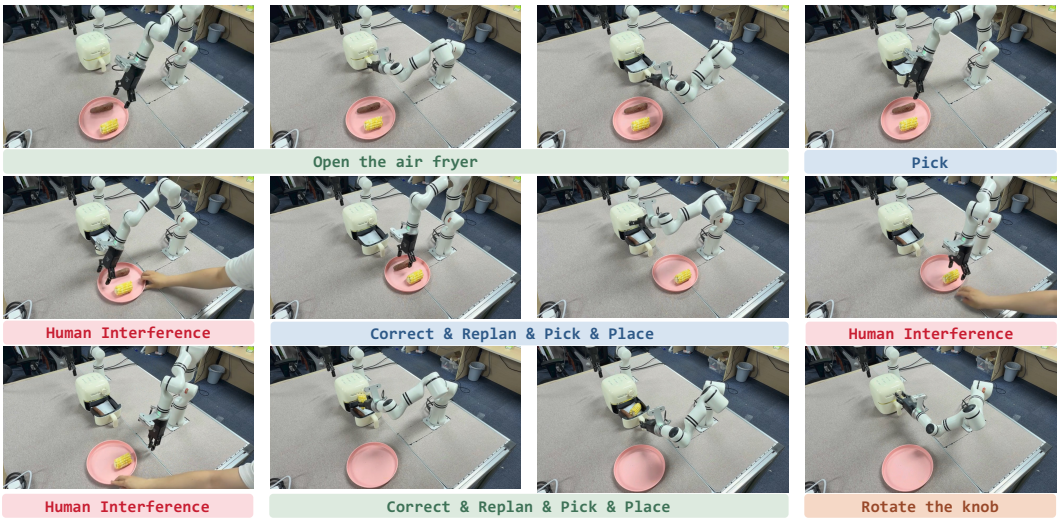


Figure 17: **Correction under human perturbation.** The sequence shows normal execution followed by deliberate scene disturbances. PGC detects the changed state, re-localizes the target and destination, and resumes execution to complete the demonstrated task.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction distinguish embodied VLM evaluation, action-fine-tuned VLA benchmarks, affordance grounding, and zero-shot robot execution. Claims are scoped to the compared baselines and evaluated settings; quantitative results and the Corrector ablation appear in Sec. 6.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Sec. G, including 2D perception, task horizons, physical recovery constraints, and limited real-robot evaluation scale.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper does not include theoretical results or proofs. It is primarily an empirical study.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Model checkpoints, training data, and code provide a route to reproduce the model results. Architecture, training, reward definitions, and data construction are described in Secs. 2 to F; Sec. H describes the released evaluation framework. Robot protocols and trial counts are reported in Secs. 6.2 and K.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The project website below the author block and the links in the introduction provide access to model weights, training datasets, training code, and EmbodiedEvalKit. The public repositories include installation, training, inference, and benchmark-evaluation instructions. Baseline scores taken from prior reports are identified in the experimental section.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Secs. C and D report architecture, optimizer settings, batch sizes, learning rates, and reward definitions. Secs. C.3, 6 and K describe evaluation settings and training regimes; Tab. 9 reports 10 trials per task and variant and distinguishes task success from event-level recovery.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We report benchmark scores and evaluation trial counts, including 10 trials per task and variant for PGC, but do not provide error bars, confidence intervals, or significance tests for the main comparisons. This limitation is acknowledged in Sec. G.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Sec. C reports GPU types, counts, memory, and the approximate RFT duration (32 H20-96GB GPUs for two days), as well as SFT and VLA hardware configurations. However, we do not provide execution times for every experiment or an estimate of total project compute, including preliminary runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that our research fully complies with it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Sec. G discusses potential benefits for robotics research and deployment, unintended uses, and the limits of physical-execution reliability. Generated plans and correction messages are inspectable, but do not guarantee safe execution.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We assess the release as research-oriented tabletop manipulation rather than a high-risk deployment. This does not imply risk-free operation: physical-execution limitations and potential unintended uses are discussed in Sec. G.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: We cite the source datasets, models, and software in the paper, but do not provide a complete asset-by-asset list of license names and terms of use. Citation alone does not satisfy the explicit license-documentation requirement of this item.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The public repositories linked in the introduction document training and inference, task-specific output formats, and benchmark evaluation. Released models and datasets are linked from the project website, with data construction and model limitations described in Secs. E to G.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The reported experiments evaluate robot and model performance, not human participants. The human interventions in Sec. L are deliberate scene disturbances used to test robot recovery, not a user study or crowdsourcing experiment.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The reported work does not study human participants. The disturbance demonstration in Sec. L evaluates robot recovery rather than human behavior or outcomes.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Secs. 2 and C describe the VLM backbone and its adaptation. Sec. F describes model-assisted data annotation and construction; Sec. D identifies the semantic reward model. The visual-trace evaluation identifies the revised LLM scoring protocol for PIO-S3-Verified in Tab. 3.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.