

RynnBrain: Open Embodied Foundation Models

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, Deli Zhao

DAMO Academy, Alibaba Group

*Core contributors in alphabetical order, †Project lead, ‡Correspondence

Despite rapid progress in multimodal foundation models, embodied intelligence community still lacks a unified, physically grounded foundation model that integrates perception, reasoning, and planning within real-world spatial-temporal dynamics. We introduce **RynnBrain**, an open-source spatiotemporal foundation model for embodied intelligence. RynnBrain strengthens four core capabilities in a unified framework: comprehensive egocentric understanding, diverse spatiotemporal localization, physically grounded reasoning, and physics-aware planning. The RynnBrain family comprises three foundation model scales (2B, 8B, and 30B-A3B MoE) and four post-trained variants tailored for downstream embodied tasks (i.e., **RynnBrain-Nav**, **RynnBrain-Plan**, and **RynnBrain-VLA**) or complex spatial reasoning tasks (i.e., **RynnBrain-CoP**). In terms of extensive evaluations on 20 embodied benchmarks and 8 general vision understanding benchmarks, our RynnBrain foundation models largely outperform existing embodied foundation models by a significant margin. The post-trained model suite further substantiates two key potentials of the RynnBrain foundation model: (i) enabling physically grounded reasoning and planning, and (ii) serving as a strong pretrained backbone that can be efficiently adapted to diverse embodied tasks.



<https://alibaba-damo-academy.github.io/RynnBrain.github.io>



<https://github.com/alibaba-damo-academy/RynnBrain>



<https://huggingface.co/collections/Alibaba-DAMO-Academy/rynnbrain>



https://www.modelscope.cn/collections/DAMO_Academy/RynnBrain

Date: February 14, 2026

1 Introduction

The advent of advanced robotic embodiments [95, 54] and general-purpose vision-language models (VLMs) [42, 93] has created a growing anticipation for versatile robots capable of adaptively performing diverse and complex tasks, which is often referred to as “embodied intelligence”. A central challenge in embodied intelligence is achieving behavioral and cognitive generalization: enabling robotic agents to transfer knowledge across environments, tasks, and interaction regimes.

Despite the strong generalization capabilities, existing VLMs are not intrinsically grounded in physical dynamics and thus struggle with spatio-temporal consistency, physical reasoning, and actionable planning. Conversely, embodied models trained primarily on action-centric data often sacrifice high-level semantic abstraction and lose the broad generalization capabilities inherited from large-scale multimodal pretraining. We argue that progress toward general-purpose embodied intelligence requires a unified foundation model that preserves the semantic breadth of VLMs while being explicitly structured around physical space, temporal dynamics, and embodiment constraints. Such an embodied foundation model should serve as a high-level cognitive “brain” for perception, reasoning, and decision-making, while remaining adaptable to downstream control systems. This report primarily examines how to develop a generalizable foundation model for embodied tasks and explores its generalization capacity and post-training potential in multiple dimensions.



Object Understanding

What material is <object> <frame 0> (637, 653), (912, 956) </object> made of?

It is made of leather.



Counting

How many sofa are there in the video?

3.



Spatial Understanding

Which is higher? object0 <object> <frame2> (430,150),(1000,1000) </object> or object1 <object> <frame3> (0,54),(1000,750) </object> ?

Object0 is higher.

Measuring from the closest point of each object, what is the distance between the table and the door?

2.2 meters.



OCR

Which brand name is written on the white cup I keep near the stove while I'm cooking?

CHASE.

General Egocentric Understanding

What did the user add to the dish?

Salt.

Area Location



Identify several points in the unoccupied region on the top surface of the toilet lid.

<area> <frame1> (890, 371), (890, 370) </area>.

Affordance Location



Push the drawer of the nightstand.

<affordance> <frame2> (630, 602) </affordance>.

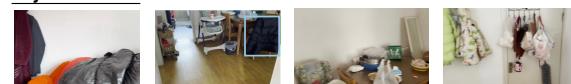
Trajectory Location



Move the blue box from the center of the bed to the drawer on the right.

<trajectory> <frame2> (714, 800), (721, 775), (735, 725), (999, 540) </trajectory>.

Object Location



If I need something to wear for warmth while sitting at the dining table, what should I pick up from the chair?

<object> <frame1> (805, 95),(1000,560) </object>.

Grasp Pose Location



Pick up the bottle at its neck.

<grasp pose> (441, 339), (456, 374), (412, 330), (403, 287) </grasp pose>.

Cognition Location

Trajectory Prediction Reasoning



Pick up the white tissue box on the table and place it on the wooden sofa.

The starting point of the trajectory is the object to be moved, which is the [white tissue box] <object> <frame 1> ((808, 708), (906, 843)) </object> located on the [table] <object> <frame 2> ((0, 697), (576, 997)) </object>. The destination is the target location for the placement action, which is the [wooden sofa] <object> <frame 0> ((59, 322), (862, 997)) </object>. Therefore, the trajectory is <trajectory> <frame 3>, (381, 721), ..., (32, 662) </trajectory> .

Counting Reasoning



How many drawers are near <object> <frame0> (450,405),(550,998) </object>?

Based on our inspection of the environment, the objects that fit the criteria are <object> <frame3> (255,355),(410,509) </object>, <object> <frame3> (410,225),(510,442) </object> and <object> <frame3> (510,135),(602,359) </object>. So the answer is 3.

Reasoning



Planning



Manipulation

Task: Open the fridge to get the drink.



Open the <affordance> (420, 350) <affordance> of the <object> (405, 14), (805, 480) <object> with the left arm.

Pick up the <affordance> (870, 390) <affordance> of <object> (851, 358), (890, 601) <object> with the left arm.

Place the <object> (804, 250), (905, 470) </object> on the <area> (330, 890) </area>.

Navigation

Walk straight, and when you reach the end of the table turn right. Wait by the brass chairs.



→ ↑ ↑ ↑
↑ ↑ ↑ ↑
↑ ↑ → →
→ ↑ ↑ STOP

Figure 1 Overview of the RynnBrain embodied foundation model. RynnBrain integrates four core capabilities: egocentric cognition, spatio-temporal localization, physically grounded reasoning, and physics-aware planning. On the input side, RynnBrain processes multimodal signals including images, videos, and spatio-temporal coordinates. On the output side, it jointly produces natural language and explicit spatial grounding primitives such as points, bounding boxes, and trajectories, enabling coherent perception, reasoning, and planning in physical environments.

Several recent efforts [92, 28, 117, 37, 5] have initiated exploration of embodied foundation models. For instance, RoboBrain 2.0 [92] unifies understanding, localization, and planning within a single VLM to facilitate complex embodied tasks, while Robix [28] emphasizes more natural human–robot interaction during execution. Despite these advances, existing embodied “brain” models exhibit three key limitations. First, their egocentric cognitive capabilities remain narrow, as training is typically confined to limited task categories or perception modalities, restricting robustness in complex environments. Second, spatial reasoning is often grounded in static image inputs, lacking coherent spatio-temporal representations necessary for global scene awareness and mobile manipulation. Third, high-level reasoning and planning are frequently conducted in a purely textual space, leading to hallucinations and inconsistencies with physical constraints.

To advance the role of embodied “brains” in complex real-world tasks, we propose **RynnBrain**, a spatio-temporal foundation model explicitly grounded in physical environments. As illustrated in Figure 1, RynnBrain demonstrates robust capabilities in four key dimensions:

1. **Comprehensive egocentric understanding:** RynnBrain excels in spatial comprehension, embodied question answering, egocentric counting, egocentric OCR, etc. Notably, it also introduces fine-grained video understanding—a capability previously overlooked by existing embodied brains.
2. **Diverse spatio-temporal localization:** RynnBrain can locate objects, target areas, and even predict trajectories across its entire episodic memory, thereby endowing robots with global spatial awareness.
3. **Physically grounded reasoning:** Instead of conventional textual reasoning, RynnBrain employs an interleaved reasoning strategy that alternates between textual and spatial localization, ensuring that its reasoning traces are firmly grounded in the physical environment.
4. **Physics-aware planning:** To provide downstream policy models with more accurate planning instructions, RynnBrain integrates the location information of affordance, areas, and objects directly into its planning outputs. Consequently, even highly intricate and fine-grained tasks can be effectively addressed within our hierarchical system architecture.

We build RynnBrain on top of Qwen3-VL [6]. To accommodate varying computational resource constraints, we release two dense variants (2B and 8B) and one mixture-of-experts (MoE) model (30B-A3B). Given comparable inference latency, RynnBrain surpasses all existing embodied brain models in terms of comprehension, localization, and planning capabilities. Beyond foundational pretraining, we explore four post-training directions: RynnBrain-CoP, RynnBrain-Nav, RynnBrain-Plan, and RynnBrain-VLA. RynnBrain-CoP introduces chain-of-point reasoning, an interleaved reasoning mechanism that alternates between textual reasoning and spatial grounding, enabling physically grounded prediction. This design yields superior performance on tasks requiring precise localization, counting, and other embodied perceptual reasoning capabilities. RynnBrain-Nav demonstrates that adopting RynnBrain as a backbone substantially elevates performance ceilings across various embodied tasks. RynnBrain-Plan validates the effectiveness of the fine-grained manipulation-planning paradigm that alternates between textual reasoning and localization. Finally, RynnBrain-VLA shows that embodiment-agnostic foundational pretraining under the RynnBrain paradigm benefits downstream VLA models that directly predict low-level actions.

A fundamental bottleneck for embodied foundation models is the scarcity of high-quality training data. We observe that more realistic and diverse data can substantially enrich and deepen RynnBrain’s capabilities in real-world scenarios. To this end, we design dedicated data pipelines tailored to key competencies, including OCR, spatio-temporal localization, action planning, and physically grounded reasoning. Importantly, our data construction framework strategically leverages the priors of pretrained foundation models, introducing human supervision only at critical decision points. This human–model collaborative data flywheel improves annotation efficiency and data quality under constrained labeling budgets, enabling the training corpus to scale over 20 million samples.

We extensively evaluate the proposed RynnBrain models in multiple dimensions. Also, recognizing that existing open-source benchmarks inadequately assess fine-grained understanding and spatio-temporal localization, we introduce **RynnBrain-Bench**, a curated benchmark with carefully filtered and manually verified annotations to ensure robustness and reliability. Across 28 benchmarks, RynnBrain demonstrates strong egocentric cognition, including spatial and temporal understanding, OCR, and robot question answering, as well as

diverse localization capabilities spanning objects, areas, affordances, and trajectories. Meanwhile, it retains competitive general-purpose visual understanding and instruction-following capabilities.

We further evaluate four post-trained variants across distinct embodied domains: spatio-temporal reasoning (RynnBrain-CoP), vision-and-language navigation (RynnBrain-Nav), manipulation planning (RynnBrain-Plan), and vision-language-action modeling (RynnBrain-VLA). The interleaved grounding–reasoning paradigm of RynnBrain-CoP improves performance on complex spatio-temporal tasks (e.g., trajectory prediction) by approximately 7%. On the R2R [3] and RxR [49] benchmarks, RynnBrain-Nav achieves state-of-the-art results and consistently surpasses Qwen3-VL-based counterparts across model scales. For manipulation planning, RynnBrain-Plan adopts two online evaluation protocols, VLMs-UMI and VLMs-VLA. VLMs-UMI directly measures the accuracy and efficiency of high-level planning, while The VLMs-VLA framework evaluates how RynnBrain’s physics-aware, spatially explicit plans enhance downstream VLA execution, thereby strengthening the robustness of the hierarchical embodied architecture. In high-complexity grasping scenarios, RynnBrain-VLA consistently outperforms models fine-tuned from $\pi_{0.5}$ [9], indicating that strong scene understanding and embodied grounding form a critical foundation for generalizable VLA systems.

All code, model checkpoints, and benchmarks are publicly released to facilitate reproducibility and further research. We envision RynnBrain as a foundational step toward physically grounded general intelligence, where unified spatio-temporal reasoning and physics-aware planning enable embodied agents to operate robustly across diverse real-world settings.

2 Overview

2.1 Model Architecture

An overview of the RynnBrain architecture is shown in [Figure 2](#). RynnBrain adopts a decoder-only vision–language architecture following the design principles of Qwen3-VL [6]. It comprises a vision encoder, a vision-language projector, and a large language model (LLM) backbone initialized from Qwen3-VL variants (Qwen3-VL-2B/8B/30B-A3B-Instruct). In addition, we also employ the techniques of DeepStack [68] and Interleaved MRoPE [41] to better integrate multimodal information.

2.2 Infrastructure

As a general-purpose embodied foundation model, the training data of RynnBrain consists of multiple modalities—including video, image, and text—across a wide range of tasks. These tasks range from short-response tasks, such as localization and spatial perception, to long-form tasks involving detailed multimodal captioning and complex reasoning. This inherent task diversity results in sequence length distributions characterized by high variance and a pronounced long-tail profile. Since computational complexity scales with sequence length, a naive distribution of samples across a data parallel (DP) training environment induces a severe straggler effect, where workers assigned heavy workloads become throughput bottlenecks.

To mitigate this, we implement an online load-balancing pipeline. Specifically, we first estimate the sequence lengths of all samples according to pre-computed image sizes and the numbers of text tokens. During the batch sampling phase of the training process, rather than assigning an equal number of samples to each DP worker, we aggregate all samples across the DP group and redistribute them based on the objective of minimizing the maximum cumulative sequence length within each DP worker. To solve this redistribution efficiently, we adopt a greedy approximation algorithm that prioritizes longer sequences: we initialize buffers equal to the DP world size, sort sequences in descending order of length, and iteratively assign each to the buffer with the smallest current total length. This process is executed during data prefetching; under the Single Program, Multiple Data (SPMD) framework, stable sorting ensures that global data distribution remains consistent across all workers. This fast and dynamic allocation prevents training stalls while maintaining flexibility, eliminating the need for costly data pre-processing when hyperparameters or datasets change.

To maintain convergence stability after sample redistribution, the global number of tokens is required under

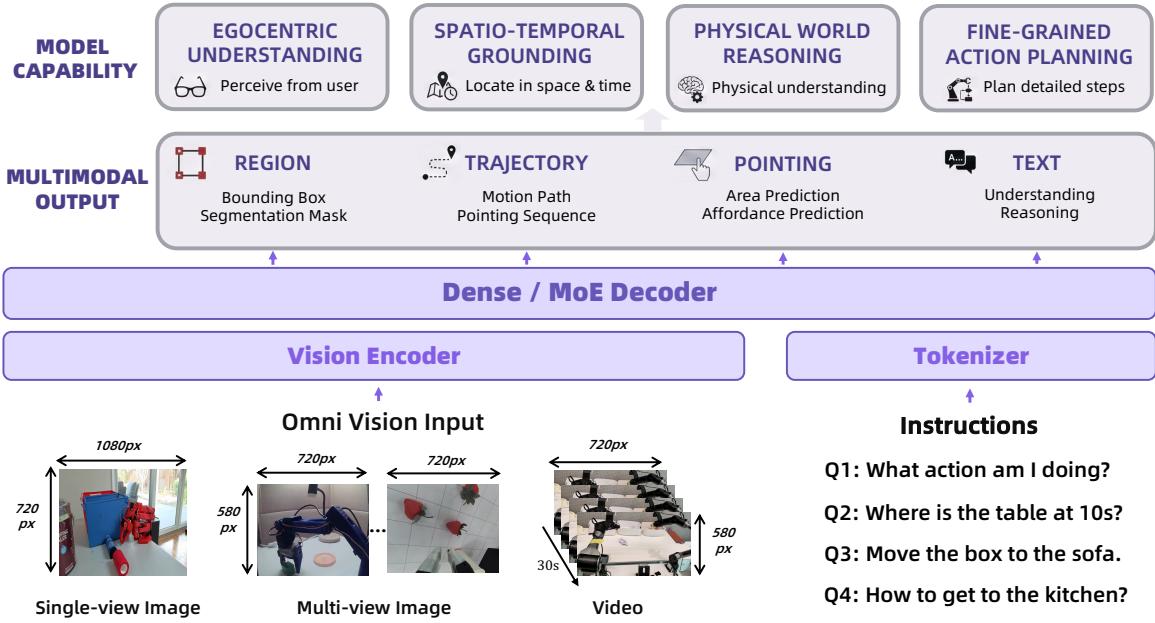


Figure 2 Overview of the RynnBrain architecture. RynnBrain processes omni vision inputs, including single view images, multi view images, and videos, together with language instructions. A shared dense or mixture of experts decoder generates aligned multimodal outputs, including text, regions, trajectories, and pointing signals. This unified output space supports egocentric understanding, spatiotemporal grounding, physically grounded reasoning, and fine grained action planning in real world environments.

the traditional per-token loss formulation:

$$\mathcal{L} = \frac{1}{\sum_{i=1}^n \sum_{j=1}^{b_i} s_{ij}} \sum_{i=1}^n \sum_{j=1}^{b_i} \sum_{k=1}^{s_{ij}} l_{ijk}, \quad (1)$$

where n is the DP world size, b_i is the local batch size on i -th worker, s_{ij} is the sequence length of the j -th sequence, and l_{ijk} is the per-token loss. However, calculating the global token count (the denominator) requires an additional all-gather operation across the DP group, which introduces synchronization overhead and reduces training efficiency. To circumvent this, we adopt a per-sample loss reduction strategy:

$$\mathcal{L} = \frac{1}{b} \sum_{i=1}^n \sum_{j=1}^{b_i} \frac{1}{s_{ij}} \sum_{k=1}^{s_{ij}} l_{ijk}, \quad (2)$$

where b is the global batch size. Since b is a constant known to each worker, this strategy eliminates the need for extra communication and improves efficiency. The holistic approach doubles training efficiency while preserving model stability and convergence properties.

To accommodate the models within the memory constraints of a single GPU, we employ the ZeRO-1 optimizer [77] and per-block gradient checkpointing for training RynnBrain-2B and RynnBrain-8B. Considering the large memory consumption of the logits, we selectively filter out tokens that do not require loss calculation—such as multimodal tokens—during the forward pass of the output head. For the larger RynnBrain-30B-A3B model, we employ the ZeRO-2 optimizer [77] and expert parallel (EP) with a world size of 2 to partition and fit the model in a single GPU. To optimize computational throughput, we implement the grouped linear operation for MoE layers with packed inputs and weights based on the kernel templates from NVIDIA CUTLASS ¹. Cross-GPU token dispatching for EP is facilitated via DeepEP [57]. For broad accessibility and extensibility,

¹<https://github.com/NVIDIA/cutlass>

our training and inference frameworks are based on the HuggingFace Transformers [101] library and have been released as open-source.

3 Physics-Aware Spatio-temporal Pretraining

Enabling generalizable robots to interact naturally with real-world environments requires two fundamental capabilities: (1) Spatio-temporal Memory: Through historical visual memory, the robot must establish multi-dimensional representations encompassing space, location, events, trajectories, and e.g., thereby enabling adaptation to complex and dynamic environments. (2) Physical World Grounding: All robotic cognitive processes must be fundamentally rooted in the objective reality of the physical world. This chapter primarily introduces the pretraining methodology of RynnBrain, which is explicitly guided by these two insights.

3.1 Training Recipe

To equip RynnBrain with spatio-temporal memory and physical grounding, we adopt a unified pretraining framework that maps multimodal inputs into a shared semantic representation space. The training recipe is structured around two core components: a unified input–output representation and a physics-aware optimization strategy.

Unified Spatio-temporal Representation. To support spatio-temporal memory, we treat images and videos as a unified visual modality. Formally, a visual input \mathbf{V} is represented as a sequence of frames $\{I_t\}_{t=1}^T$, where $T = 1$ for static images and $T > 1$ for videos. For videos, frames are uniformly sampled to preserve temporal continuity. Each frame is encoded into visual tokens and augmented with temporal positional embeddings to encode frame order. This representation enables RynnBrain to capture temporal dependencies, motion patterns, and trajectory dynamics across extended visual sequences.

Physically Grounded Output Space. To ensure physical world grounding, we explicitly structure the output space to bridge high-level reasoning and low-level execution. Unlike conventional vision–language models that treat spatial quantities as free-form text, we introduce discrete coordinate tokens to represent physical locations. All spatial entities, including bounding boxes \mathcal{B} , points \mathcal{P} , and trajectory waypoints \mathcal{T} , are normalized to the range $[0, 1000]$ and encoded as integer tokens. This discretization converts continuous spatial prediction into a classification problem, allowing the model to generate precise and physically meaningful spatial outputs using the same autoregressive mechanism as language generation.

Optimization. RynnBrain is trained end to end using a standard next-token prediction objective. The training loss is defined as:

$$\mathcal{L} = - \sum_{i=1}^L \log P(y_i | y_{<i}, \mathbf{V}, \Theta), \quad (3)$$

where \mathbf{V} denotes the visual input, \mathbf{y} is the mixed sequence of textual and coordinate tokens, and Θ represents the model parameters. Optimization hyperparameters are adjusted across model scales based on pilot experiments conducted on a representative subset of the pretraining data. Detailed training configurations are reported in Table 1.

Table 1 Hyperparameters of the pretraining stage for RynnBrain model series.

Parameter	RynnBrain-2B	RynnBrain-8B	RynnBrain-30B-A3B
Base Model	Qwen3-VL-2B-Instruct	Qwen3-VL-8B-Instruct	Qwen3-VL-30B-A3B-Instruct
Optimizer	AdamW	AdamW	AdamW
Learning Rate	$5e^{-6}$	$2e^{-6}$	$2e^{-6}$
Learning Rate Vision	$1e^{-6}$	$2e^{-6}$	$2e^{-6}$
Global Batch Size	512	1024	1024
Warmup Ratio	0.03	0.03	0.03

Table 2 Pretraining data mixture statistics for RynnBrain

Category	Sub-Task	Data Sources	Samples (M)
General MLLM	General	LLaVA-OV-SI [52], LLaVA-Video [118], ShareGPT-4o-video [15], VideoGPT-plus [61], FineVideo [29], CinePile [79], ActivityNet [10], YouCook2 [123], LLaVA-SFT [58]	4.80
Cognition	Object Understanding	RynnBrain-Object, RefCOCO [110], Google Refexp [64], Osprey-724K [112], DAM [55], VideoRefer-700k [113]	1.10
	Spatial Understanding	Sensenova-SI-800K [12], VSI-590k [106], VLM-3R [27], RynnBrain-Spatial	2.50
	Counting	RynnBrain-Counting, Molmo2 [20]	0.30
	OCR	RynnBrain-OCR	1.00
	Egocentric Task Understanding	EgoRe-5M [73], Egotaskqa [44], Env-QA [1], QAEgo4d [34], RoboVQA [85], Robo2vlm [13], ShareRobot [43]	2.77
Localization	Object Localization	ADE20K [121], COCOSTuff [11], Mapillary [70], PACO-LVIS [76], PASCAL-Part [16], VG [48] RoboAfford-Object [36], RynnBrain-Grounding	1.20
	Area Localization	RefSpatial [122], RoboAfford-Area [36], Molmo2 [20], RynnBrain-Area	3.37
	Affordance Localization	RynnBrain-Affordance, RoboAfford-Affordance [36]	1.13
	Trajectory Prediction	RynnBrain-Trajectory, FSD [111]	0.56
Planning	Grasp Pose Prediction	Grasp-Anything [96]	1.00
	Manipulation	AgibotWorld [22], Open X-Embodiment [21], RynnBrain-Planning	0.16
Total			19.89

3.2 Pretraining Data

Table 2 summarizes the data sources and corresponding data volumes used for pretraining RynnBrain. Below, we describe each dataset grouped by category.

3.2.1 General MLLM Data

To retain broad multimodal understanding, we construct a general-purpose MLLM pretraining corpus spanning both images and videos across diverse domains. The corpus aggregates publicly available datasets, including LLaVA-OV-SI [52], LLaVA-Video [118], ShareGPT-4o-video [15], VideoGPT-plus [61], FineVideo [29], CinePile [79], ActivityNet [10], YouCook2 [123], LLaVA-SFT [58], and VideoLLaMA 3 [114]. Together, these datasets support open-vocabulary object recognition, conversational video understanding, long-horizon temporal reasoning, and image–text supervision. In total, the corpus comprises 4.8M samples.

3.2.2 Multi-Dimensional Cognition Data

Object Understanding. The object understanding dataset is designed to enhance fine-grained object recognition and object-centric reasoning. Each sample focuses on a specific object annotated with a bounding box in a single frame, formatted as `<object> <frame n>: (coordinates) </object>`, with questions conditioned on the indicated object. The dataset covers object attributes such as category, color, shape, function, spatial position, and related properties.

We combine publicly available datasets [110, 64, 112, 55, 113] with self-collected egocentric data, yielding over 1.1M samples. For the egocentric subset, we construct an object-centric QA generation pipeline on indoor videos. Objects are first identified using Qwen2.5-VL [7], detected in key frames with Grounding DINO 1.5 [81], and segmented and tracked using SAM2 [78]. To reduce redundancy, we limit each video to at

most two instances per object category. Object-centric QA pairs are then generated using Qwen2.5-VL and manually filtered for quality, resulting in 712K high-quality QA samples.

Spatial Understanding. Spatial reasoning is critical for embodied tasks such as navigation and manipulation, yet remains a weakness of many existing VLMs. To address this limitation, we curate over 2.5M spatial instruction samples spanning two categories: general spatial understanding and fine-grained object-centric spatial reasoning.

General spatial understanding data are sourced from publicly available datasets, including Sensenova-SI-800K [12], VLM-3R [27], and VSL-590K [106]. For fine-grained spatial annotations, we process self-collected indoor images and videos using MASt3R-SLAM [69], which reconstructs 3D point clouds and estimates camera extrinsics from RGB video. Instance-level segmentations are projected into the reconstructed 3D space, and the point cloud is realigned using RANSAC [31] to detect the ground plane and enforce a gravity-aligned world coordinate system.

Based on these calibrated 3D scenes, we generate spatial QA pairs requiring reasoning about metric distances, relative positions, heights, and other 3D relationships. QA generation follows a template-based scheme, where missing attributes are computed directly from the underlying geometry. This process yields 855K video-based and 272K image-based spatial QA samples.

Counting. The counting dataset is designed to improve robust estimation of object quantities in complex visual scenes. We combine publicly available data with egocentric indoor videos. The public component consists of the Molmo2 counting subset [20], comprising 222K samples with diverse scenes and reliable annotations. To incorporate embodied perspectives, we further curate 42K counting QA pairs from self-collected egocentric videos. All annotations are manually verified to ensure accuracy and consistency.

OCR. The OCR dataset equips the model with scene text recognition and grounding capabilities essential for text-rich embodied environments. We construct approximately 1M OCR QA samples from egocentric videos sourced from Ego4D [34], Charades-Ego [88], and EPIC-KITCHENS [23]. Scene text is detected using GoMatching [38, 39], and videos are segmented based on text appearance patterns into clips of 3 to 15 seconds, yielding 85,324 text-containing segments.

For each segment, human annotators label the first appearance frame, the clearest frame, text transcription, and bounding polygons. QA pairs are generated using two complementary strategies: (i) GPT-5.2 [71] produces goal-oriented, first-person questions grounded in practical text understanding, yielding 256K contextual QA samples; (ii) template-based generation produces structured questions covering text reading, temporal localization, verification, and multiple-choice recognition, yielding 722K samples. GPT-generated questions are filtered to ensure visual perception is required.

The OCR dataset is provided in two formats: normal video QA (893K samples), where the model predicts textual answers from video input, and area prediction QA (85K samples), where the model outputs frame indices and normalized bounding coordinates.

Egocentric Task Understanding To support broad egocentric task comprehension, we construct an egocentric task understanding dataset comprising 2.77M video-text pairs. The dataset aggregates publicly available resources, including Env-QA [1], EgoTaskQA [44], RoboVQA [85], EgoRe-5M [73], QAEgo4D [72], Robo2VLM [13], and ShareRobot [43]. Videos shorter than 3 seconds are excluded to ensure sufficient temporal context for task-level reasoning.

3.2.3 Spatio-Temporal Location Data

Object Location. Object localization enables the model to interpret language instructions and identify target objects in images and videos. Each sample is represented as $(\mathbf{V}, Q, \mathcal{B}, t)$, where $\mathbf{V} = \{I_t\}_{t=1}^T$ denotes a sequence of T frames ($T = 1$ for static images), Q is a textual query describing the target object, $\mathcal{B} = \{(x_0, y_0, x_1, y_1)\}$ is the bounding box of the target with normalized coordinates in $[0, 1000]$, and t denotes the key frame where the object is most clearly observed.

We aggregate 900K samples from publicly available grounding datasets, including ADE20K [121], COCO [56], Mapillary [70], PACO-LVIS [76], PASCAL-Part [16], VG [48], and RoboAfford++ [36]. To strengthen egocentric localization, we further construct 300K egocentric samples using the same segmentation pipeline as object understanding. Referring expressions are generated using Qwen3 [104], including simple expressions based on category or position and situational expressions that require task-level inference. All samples are manually filtered for quality.

Area Location. Area localization equips the model to identify non-object regions, such as surfaces, empty spaces, or functional areas, in images and videos. Each sample is represented as $(\mathbf{V}, Q, \mathcal{P}, t)$, where $\mathcal{P} = \{(x_i, y_i)\}_{i=1}^n$ denotes a set of normalized points indicating the target area, and t is the keyframe index.

The dataset is constructed from multiple sources. We annotate 6K egocentric house-touring video segments using LLM-generated instructions with human-selected point annotations. To enhance temporal coverage, we incorporate 222K video samples from Molmo2-VideoPoint [20]. For static scenes, we curate 448K image-area samples from indoor images using a similar pipeline. Additionally, we include 2.2M image-based samples from RoboAfford++ [36] and RefSpatial [122] to increase domain diversity.

Affordance Location. Affordance localization focuses on identifying actionable points, such as handles, buttons, or interaction hotspots, on objects or surfaces. Each sample is represented as (\mathbf{V}, Q, p, t) , where $p = (x, y)$ is a normalized affordance point and t denotes the key frame index where the affordance is most relevant..

We follow a construction pipeline similar to area localization. For spatiotemporal data, we annotate 6K video segments with LLM-generated instructions and human-labeled affordance points. For static images, we derive 476K affordance samples from 500K indoor images. To improve generalization, we further include 260K affordance samples from RoboAfford++ [36], focusing on actionable interactions.

Trajectory Location Trajectory localization trains the model to predict plausible two-dimensional manipulation trajectories for object interaction tasks. Each sample is represented as $(\mathbf{V}, Q, \mathcal{T}, t_s)$, where $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^m$ is an ordered set of up to 10 normalized trajectory points, and t_s denotes the starting frame.

We construct 6K spatiotemporal samples with LLM-generated instructions and human-annotated trajectories, emphasizing cross-frame reasoning. For static images, we generate 507K image-trajectory samples from indoor scenes. To further diversify manipulation scenarios, we include 13K trajectory samples from FSD [111].

Grasp Pose Location The grasp pose location dataset equips the model with the ability to predict precise robotic grasp poses for target objects. Each sample is represented as (I, Q, \mathcal{G}) , where I denotes a single RGB image, Q is a textual query specifying the target object and grasping task, and $\mathcal{G} = \{(x_i, y_i)\}_{i=1}^4$ denotes four ordered corner points defining an oriented grasp rectangle.

We construct this dataset from Grasp-Anything [96], which provides grasp annotations for everyday objects in tabletop scenes using oriented rectangles parameterized by center (c_x, c_y) , dimensions (w, h) , and rotation angle θ . We process approximately 995K images at 416×416 resolution, each containing one or more annotated grasp candidates. For each object, we select the highest-scoring grasp and convert the parameterized representation into four corner points via rotation. This representation explicitly captures grasp orientation and gripper alignment, supporting spatially precise manipulation planning.

To promote linguistic diversity, we generate instruction prompts using a weighted template strategy: 40% object-centric prompts, 30% scene-aware prompts incorporating scene descriptions, and 30% task-oriented prompts emphasizing manipulation intent. Grasp pose outputs are expressed using multiple concise response templates to improve robustness to varied linguistic formulations.

Following this pipeline, we construct a static-image grasp pose dataset comprising 1.3M training samples derived from 945K images, with an average of 1.44 samples per image. This dataset enables RynnBrain to learn orientation-aware and spatially grounded grasp pose prediction, a key capability for robotic manipulation.

3.2.4 Physics-Aware Planning Data

To support precise manipulation planning, we design a structured planning data schema. Following Hi Robot [87], we adopt atomic actions as the minimal units of planning. Long-horizon tasks are decomposed

into temporally ordered sub-tasks using an in-house model and subsequently verified by human annotators.

To enable fine-grained spatial grounding, each sub-task is annotated with a unified grounding schema that includes target object bounding boxes, placement area points, and affordance points. Formally, each training sample is represented as $(\mathbf{V}, Q, \mathcal{M})$, where $\mathbf{V} = \{I_t\}_{t=1}^T$ denotes the visual context preceding the current step, Q is a high-level task instruction (e.g., “Please help me tidy up the sink.”), and \mathcal{M} denotes the current sub-task plan, expressed as a mixed sequence of textual tokens and grounding annotations (bounding boxes \mathcal{B} , area points \mathcal{P} , and affordance point p).

We incorporate publicly available datasets, including AgibotWorld Alpha [22] and Open X-Embodiment [21], formatted as single-turn planning dialogues. To strengthen physical grounding, we further augment these data with spatial annotations on randomly sampled frames. This design enables RynnBrain to integrate object, region, and affordance information directly into planning outputs, providing downstream manipulation policies with precise and physically grounded guidance.

4 Physically Grounded Chain-of-Point Reasoning

Most existing multimodal reasoning models [30, 59, 51] rely on purely textual reasoning paradigms. Although several approaches [60, 120, 33, 108] incorporate auxiliary tools such as region zooming to alleviate visual recognition challenges, their reasoning processes remain largely detached from physical spatial structure, limiting generalization beyond narrowly defined tasks. Alternative methods that explore visual imagination during reasoning [35, 103, 26] further suffer from hallucinated visual content, undermining physical consistency.

For embodied agents operating in real-world environments, reasoning must be grounded in observable physical evidence. To this end, we introduce *Chain-of-Point* (**CoP**) reasoning in RynnBrain, an interleaved reasoning paradigm that integrates explicit spatial grounding with textual inference over egocentric video streams. By anchoring intermediate reasoning steps to concrete spatial references, CoP bridges language-based cognition and physical perception, enabling reasoning that remains consistent with the underlying environment. This section presents the design and explorations of CoP reasoning for physically grounded embodied intelligence.

4.1 Cold-Start Supervised Fine-Tuning

4.1.1 Training Recipe

The training of the CoP reasoning model, i.e., RynnBrain-CoP, begins with the pretrained RynnBrain model, which establishes a strong foundation in general embodied understanding. We perform full-parameter supervised fine-tuning using the AdamW optimizer with a cosine learning rate schedule. We set the peak learning rate to 1×10^{-5} for the language model and projector, and 2×10^{-6} for the vision encoder, with a 3% warmup period. The model is trained for 1 epoch with a global batch size of 128. To effectively process long-horizon egocentric videos, we sample frames at 2 FPS (up to 2048 frames) and set the maximum context length to 16,384 tokens. We utilize DeepSpeed ZeRO-1 to optimize memory efficiency during training.

4.1.2 Data

To develop the model’s CoP reasoning capability, we construct a specialized dataset that explicitly interleaves textual reasoning with visual grounding. This process is based on the core spatio-temporal location datasets (Area, Affordance, and Trajectory Location). Each sample is enriched with a “Thinking” field that bridges high-level task understanding with low-level spatial localization.

The data generation follows a structured pipeline: First, given the original task instruction and video frames, we use Qwen3-VL-235B to pre-generate a step-by-step textual reasoning chain. This chain includes key reasoning steps and explicitly marks potential entities (e.g., objects or areas) using square brackets (e.g., [white flower-patterned wallpaper]). These entities are candidates for visual grounding. Next, an in-house model is employed to classify each marked entity as either “area” or “object” based on the textual context. Finally, human annotators review the reasoning chain and entity classifications. For each identified entity, they select the most relevant and clear frame from the video sequence and perform precise annotation:

for entities classified as “area”, they annotate a set of representative points; for those classified as “object”, they annotate a 2D bounding box. The grounding results are then inserted back into the reasoning text in the structured format `<object/area> <frame n>: ...; (coordinates) </object/area>`, creating a seamless interleaving of textual reasoning and spatial grounding.

This process results in a CoT-style dataset where the model’s internal thinking process is not merely abstract, but is continually anchored to specific visual evidence in the physical space. Formally, a sample extends the base tuple to $(V, Q, \mathcal{P}_{final}, t_s, R)$, where R represents this interleaved reasoning chain. This dataset is fundamental for training RynnBrain-CoP, enabling RynnBrain to perform transparent, grounded, and hallucination-resistant reasoning essential for reliable operation in embodied scenarios.

4.2 Reinforcement Learning

4.2.1 Training Recipe

We employ Group Relative Policy Optimization (GRPO) [86] to align the model with physically grounded reasoning tasks. Unlike standard PPO [84] which requires a value function (critic) to estimate the advantages, GRPO estimates the baseline from the group scores of multiple sampled outputs generated from the same prompt. This significantly reduces memory usage and training complexity.

Formally, for each query q , we sample a group of G outputs $\{o_1, o_2, \dots, o_G\}$ from the old policy $\pi_{\theta_{old}}$. The optimization objective is defined as follows:

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E} \left[\frac{1}{G} \sum_{i=1}^G (\min(\rho_i A_i, \text{clip}(\rho_i, 1 - \epsilon, 1 + \epsilon) A_i) - \beta \mathbb{D}_{KL}(\pi_\theta(o_i|q) \parallel \pi_{\text{ref}}(o_i|q))) \right] \quad (4)$$

where $\rho_i = \frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}$ is the importance sampling ratio, and β is the coefficient for the KL divergence penalty with respect to the reference model π_{ref} . The advantage A_i for each output is computed by normalizing the rewards within the group:

$$A_i = \frac{r_i - \text{mean}(\{r_1, \dots, r_G\})}{\text{std}(\{r_1, \dots, r_G\}) + \epsilon} \quad (5)$$

The training is initialized from our cold-start SFT model. We utilize the SGLang [119] inference engine for efficient rollout generation with a group size of $G = 5$. The training runs for 10 epochs with a batch size of 128. We optimize the policy using a cosine learning rate schedule starting at 2×10^{-6} with a 3% warmup. To ensure stability, we set the clipping range ϵ to $[0.2, 0.28]$ and the KL coefficient $\beta = 0.02$. The maximum sequence length is set to 16,384 tokens to accommodate long-context egocentric video reasoning.

4.2.2 Reward Design

We design task-specific rule-based reward functions to strictly anchor the model’s reasoning in the physical world. All spatial coordinates are normalized to the unit interval $[0, 1]$ prior to reward computation.

Trajectory. The trajectory reward evaluates the shape and sequential alignment of the predicted path. First, both the predicted sequence $\mathcal{P} = (p_1, \dots, p_M)$ and the ground truth sequence $\mathcal{G} = (g_1, \dots, g_N)$ are resampled to have the same number of points uniformly spaced by arc length. We then calculate the Discrete Fréchet Distance (DFD), defined recursively. Let $c(i, j)$ be the coupling distance between prefixes $p_{1:i}$ and $g_{1:j}$:

$$c(i, j) = \max(\|p_i - g_j\|_2, \min(c(i-1, j), c(i, j-1), c(i-1, j-1))) \quad (6)$$

with $c(0, 0) = \|p_1 - g_1\|_2$. The final distance is $D_F = c(M, N)$. The reward decays exponentially with this distance:

$$r_{\text{traj}} = \exp(-\lambda_{\text{traj}} \cdot D_F) \quad (7)$$

Affordance. For affordance, we measure the set similarity between predicted interaction points \mathcal{P} and ground truth points \mathcal{G} using the Bidirectional Mean Euclidean Distance, a variant of the Chamfer distance. This

metric jointly captures precision, by penalizing invalid predictions, and recall, by measuring coverage of all annotated affordance regions:

$$D_{\text{bidir}}(\mathcal{P}, \mathcal{G}) = \frac{1}{2} \left(\frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \min_{g \in \mathcal{G}} \|p - g\|_2 + \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \min_{p \in \mathcal{P}} \|p - g\|_2 \right) \quad (8)$$

The reward is defined as $r_{\text{aff}} = \exp(-\lambda_{\text{aff}} \cdot D_{\text{bidir}})$.

Area. For area identification, we treat the task as a point-retrieval problem within a valid polygon. Let $S_{\mathcal{G}}$ denote the geometric region defined by the ground truth polygon. The reward is the strict accuracy of the generated points \mathcal{P} :

$$r_{\text{area}} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \mathbb{I}(p \in S_{\mathcal{G}}) \quad (9)$$

where $\mathbb{I}(\cdot)$ is the indicator function.

4.2.3 Reinforcement Learning Data

To support efficient and high-quality policy exploration, we construct a curated reinforcement learning dataset based on the spatiotemporal localization data used during pretraining, covering area, affordance, and trajectory tasks. These tasks provide essential supervision for visual evidence localization and physics-aware reasoning.

We apply a difficulty-aware filtering strategy to remove trivial samples that do not require grounded reasoning, as well as excessively noisy or ambiguous cases. Each candidate sample is scored by a pretrained SFT model using the evaluation metrics described in Section 6.1, and only samples of intermediate difficulty (scores between 40 and 80) are retained. To further improve temporal localization, we additionally include a subset of failure cases in which the SFT model incorrectly selects key frames.

This refinement process yields a high-quality dataset of 30K training samples. By constraining exploration within a structured and physically grounded regime, the dataset reduces hallucinations and promotes more reliable reasoning during reinforcement learning.

5 Post-training for Embodied Tasks

5.1 Vision-Language Navigation

The RynnBrain foundation model is pretrained to enhance its general understanding capabilities. To validate the benefits of this pretraining for the task of Vision-Language Navigation (VLN), we subsequently fine-tune the model on a VLN dataset. The fine-tuned RynnBrain model, namely **RynnBrain-Nav** is then deployed as an agent to perform navigation tasks.

Problem Formulation. In the VLN task, an embodied agent is tasked with interpreting a natural language instruction Q . At time t , based on a sequence of visual observations $O = \{o_0, o_1, \dots, o_t\}$ and language instruction Q , the agent must generate a corresponding action a_t to follow the instruction and reach the target destination. Each observation $o_i \in \mathbb{R}^{3 \times H \times W}$ is an RGB image from the agent’s current perspective. The discrete action space is defined as $\mathcal{A} = \{\uparrow, \leftarrow, \rightarrow, \text{STOP}\}$, representing the low-level movements of moving forward by 30 centimeters, turning left or right by 15 degrees, and halting the episode, respectively.

Data Formulation. We adopt a multi-turn conversational format, which is efficient for both training and inference. Following the methodology of StreamVLN [100], the training data is organized as a sequence of observation-action pairs, $d_i = (o_i, a_i)$. The training objective is to predict the next action a_i based on current visual observation o_i and the preceding conversational history. This formulates each VLN trajectory into an interleaved image-text sequence, represented as:

$$\{o_0, a_0, o_1, a_1, \dots, o_n, a_n\} \quad (10)$$

Data Collection. We curated a large-scale, navigation-specific training dataset using the Habitat simulator [83] and using the ground truth action to generate the image-text interleaved VLN dataset. The primary component

consists of 450K video clips generated from R2R [3], R2R-EnvDrop [91], and RxR [49] trajectories across 60 Matterport3D (MP3D) [75] environments. To enhance scene diversity and improve generalization, this dataset was augmented with an additional 300K samples from a subset of the ScaleVLN [99] dataset. We further implement multi-turn DAgger [82] to further collect the data to improve the performance of the model.

Fine-tuning Settings. We perform full-parameter supervised fine-tuning using the AdamW optimizer with a cosine learning rate schedule. We set the peak learning rate to 2×10^{-5} for the language model and projector, and 2×10^{-6} for the vision encoder, with a 3% warmup period. The model is trained for 1 epoch with a global batch size of 256. To effectively process long-horizon egocentric videos, we sample frames at 2 FPS (up to 2048 frames) and set the maximum context length to 16,384 tokens. We utilize DeepSpeed ZeRO-1 to optimize memory efficiency during training.

5.2 Manipulation Planning

Since the pretraining corpus incorporates planning-centric data, the foundation model already possesses inherent planning capabilities. However, adapting this capability to complex, long-horizon manipulation tasks requires the model to maintain effective memory. To address this issue, we utilize a tiny in-house dataset formatted as multi-turn dialogues, where the interaction history functions as an explicit memory buffer to preserve historical reasoning results. This structure enables the model to bridge individual planning steps into a coherent long-horizon strategy. Crucially, to align with this sequential inference, grounding annotations were applied exclusively to the final frame of each dialogue turn, ensuring current decisions are conditioned on both the immediate observation and the accumulated memory. Empirically, we find that this approach is highly data-efficient: fine-tuning with only a few hundred samples is sufficient to endow the model with robust long-horizon planning and generalization capabilities. Further details and quantitative evaluations are provided in [Section 6](#).

5.3 VLA

Model Architecture. To bridge the gap between planning and physical execution, we propose **RynnBrain-VLA**, which translates fine-grained plans into executable robot actions. We build RynnBrain-VLA upon RynnBrain-2B to utilize the large-scale pretraining on fine-grained object references and precise spatial localization while maintaining low inference latency. The overall model architecture is shown in [Figure 3](#). Generally, we adopt a flow matching framework to predict an action chunk [8] at each step. The VLM backbone is served as a single-stream Diffusion Transformer (DiT) taking a single packed sequence containing the condition and the noisy actions as input. To make the VLM compatible with this framework, we add three linear projections to align the dimension of the input noises, input timestamp embeddings, and the output actions to the VLM hidden size. To preserve the VLM’s inherent instruction-following capabilities, we utilize its native conversation format for organizing the input sequence. Consistent with the planning model, pointing information is passed in a text-based format, with the task’s initial frame prepended to the input sequence.

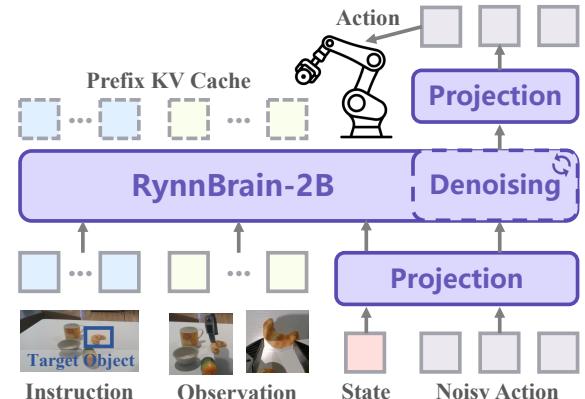


Figure 3 RynnBrain-VLA architecture.

```

<|im_start|>user
INSTRUCTION:
<start_frame>
Pick the <affordance>  $(x, y)$  </affordance> of the <object>  $(x_0, y_0), (x_1, y_1)$  </object>
OBSERVATION:
<camera_1><camera_2><camera_3>
STATE:
<state>
What action should the robot take ?<|im_end|>
<|im_start|>assistant
<action>

```

Following π_0 [8], actions are positioned at the end of the sequence to enable the KV cache during inference.

Fine-tuning. To equip the model with robotic control capabilities, we curated a dataset comprising six pick-and-place tasks involving three distinct objects, collected via manual teleoperation on a Franka Emika arm. Following data collection, each episode was annotated with its target object or placement location and aligned with the output format of the planning model. We then fine-tuned the RynnBrain-2B model on this dataset for 60k steps, using a learning rate of 2e-5 and a batch size of 32. All images were proportionally resized to a short-side dimension of 384 pixels. Experimental results show that, by leveraging RynnBrain’s embodied understanding and precise localization, this simple adaptation achieves accurate interpretation of point-based instructions and reliable grasping.

6 Evaluation

6.1 RynnBrain-Bench

For embodied brains operating in physical reality, the fine-grained spatio-temporal understanding across the entire episodic memory is essential for performing intricate embodied tasks. While existing benchmarks primarily focus on either static scene understanding with text-referenced objects [62, 25] or spatial pointing tasks with single-frame input [43], they fall short of adequately evaluating models’ capabilities in this domain. We introduce **RynnBrain-Bench**, a high-dimensional evaluation suite designed to holistically benchmark the cognition and localization capabilities of embodied understanding models in complex household environments. Advancing beyond existing benchmarks, RynnBrain-Bench features a unique emphasis on fine-grained understanding and precise spatio-temporal localization within episodic video sequences.

6.1.1 Overview

We present an overview framework of RynnBrain-Bench in Figure 4, highlighting its core dimensions and sample tasks. RynnBrain-Bench systematically measures spatio-temporal embodied understanding across four foundational pillars: *Object Cognition*, *Spatial Cognition*, *Grounding*, and *Pointing*. Covering 21 specialized sub-capabilities ranging from detailed object attributes (e.g., color, shape) to affordance points prediction, the benchmark comprises 3,616 video clips consisting of 577,998 frames, and 12,000 meticulously curated open-ended questions for comprehensive evaluation. Our data construction starts from the self-collected egocentric indoor videos and object-centric Q/A pairs initially generated with foundation model priors, followed by the rigorous human-in-the-loop annotation pipeline detailed in Section 3.2. The annotated data is cross-validated by human annotators across multiple rounds to ensure its correctness and high quality. We perform internal data balancing across sub-capabilities within each foundational dimension to ensure fairness and objectivity for evaluation. To ensure high fidelity, questions related to objects are further balanced against real-world object distributions for better authenticity.

6.1.2 Evaluation Dimensions

RynnBrain-Bench defines a new form of spatio-temporal evaluation paradigm, requiring models to perform instruction-guided cognition and localization anchored to precise spatial and temporal coordinates.

Object Cognition challenges models with fine-grained object perception and counting of region-level targets across dynamic image sequences. We assess nine core object attributes (i.e., category, color, material, shape, state, position, function, surface detail, and size)—plus a distinct object counting capability. Models are required to provide responses conditioned on questions with precise spatio-temporal positions (i.e., frame index and spatial coordinates). *Evaluation Metrics*: During evaluation, responses are scored by GPT-4o on a scale from 0 to 1, utilizing either a binary scheme or a multi-level system with 0.2-point increments.

Spatial Cognition requires models to derive 3D spatial awareness from egocentric video streams, spanning two primary perspectives: Ego-centric and World-centric. While ego-centric cognition examines the embodied agent’s evolving relationship (e.g., rotation, direction) with the environment over time, world-centric cognition evaluates the comprehension of objective 3D layouts and physical properties, such as size scale, distance, and position. *Evaluation Metrics*: For numerical questions, we apply mean relative accuracy (MRA) and rotational accuracy (RoA) to measure the score following RynnEC [24]. For textual questions, we use the binary or fine-grained scores from GPT-4o as described above.

Grounding evaluates the capability for precise spatio-temporal localization, representing a key link for anchoring understanding in reality. This task requires the brain model to (1) pinpoint the critical temporal key frame and then (2) predict the object’s spatial coordinates within that frame. We distinguish between Direct Grounding, which involves locating objects based on explicit descriptions, and Situational Grounding, which necessitates context-aware reasoning to identify and localize targets within complex scenarios. *Evaluation Metrics*: We apply the Acc@0.5 to calculate the score [45]. Specifically, the prediction is considered correct only if the model selects a frame t that contains a valid ground truth ($\mathcal{G}_t \neq \emptyset$) and the Intersection over Union (IoU) between the predicted box \mathcal{B} and \mathcal{G}_t exceeds 0.5. Let $\mathbb{I}(\cdot)$ be the indicator function, the metric is:

$$\text{Acc}@0.5 = \mathbb{I}(\mathcal{G}_t \neq \emptyset \wedge \text{IoU}(\mathcal{B}, \mathcal{G}_t) > 0.5) \quad (11)$$

Pointing aims to predict target *areas*, spatio-temporal *trajectories*, or *affordance* points across the entire episodic memory, serving as a critical bridge for robot-physical world interaction. Departing from previous benchmarks, we extend the evaluation scope to the spatio-temporal domain, where models must demonstrate the dual capacity to locate the key frame and predict corresponding task-relevant point sequences. *Evaluation Metrics*: For pointing tasks, the score is set to zero if the model-predicted frame does not contain a valid ground truth ($\mathcal{G}_t = \emptyset$). Otherwise, (1) For trajectory prediction, we apply the Discrete Fréchet Distance (DFD) distance [43] between the predicted point sequence $\mathcal{P} = (p_1, \dots, p_M)$ and the ground truth sequence $\mathcal{G} = (g_1, \dots, g_N)$. We resample both sequences to 15 points uniformly distributed along the arc length and compute the DFD according to Equation 7. (2) For area prediction, we calculate the proportion of predicted points \mathcal{P} falling within the ground truth polygon $S_{\mathcal{G}}$ according to Equation 9. (3) For affordance prediction, we evaluate spatial proximity using the exponential decay of the Euclidean distance from each predicted point $p \in \mathcal{P}$ to its nearest neighbor in the ground-truth set \mathcal{G} :

$$D(\mathcal{P}, \mathcal{G}) = \exp\left(-\frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \min_{g \in \mathcal{G}} \|p - g\|_2\right) \quad (12)$$

6.2 Embodied Cognition Capability

To assess RynnBrain’s embodied cognition capabilities, we evaluate it on a diverse suite of benchmarks, including VSI-Bench [105], MMSI [107], ERQA [94], RoboSpatial [90], EgoTaskQA [44], EgoTextVQA [124], Open-X VQA [13], MindCube [109], RynnBrain-Object and RynnBrain-Spatial. As shown in Table 3, our RynnBrain-8B outperform the base model Qwen3-VL-8B on 9 of 11 embodied cognition tasks. RynnBrain-8B delivers substantial gains across a variety of tasks. For instance, on the spatial reasoning benchmark VSI-Bench, RynnBrain-8B achieves 71.0 score, surpassing the previous best result of 60.3, and on RoboSpatial it exceeds the previous top-performing method by 11.3%. RynnBrain-8B also attains strong performance on RynnBrain-Object and RynnBrain-Spatial, indicating robust improvements in object-centric and spatially grounded reasoning. Similarly, we evaluate RynnBrain-30B (A3B) model on various embodied cognition benchmarks, with results summarized in Table 4. From the table, it is clear that RynnBrain-30B (A3B)

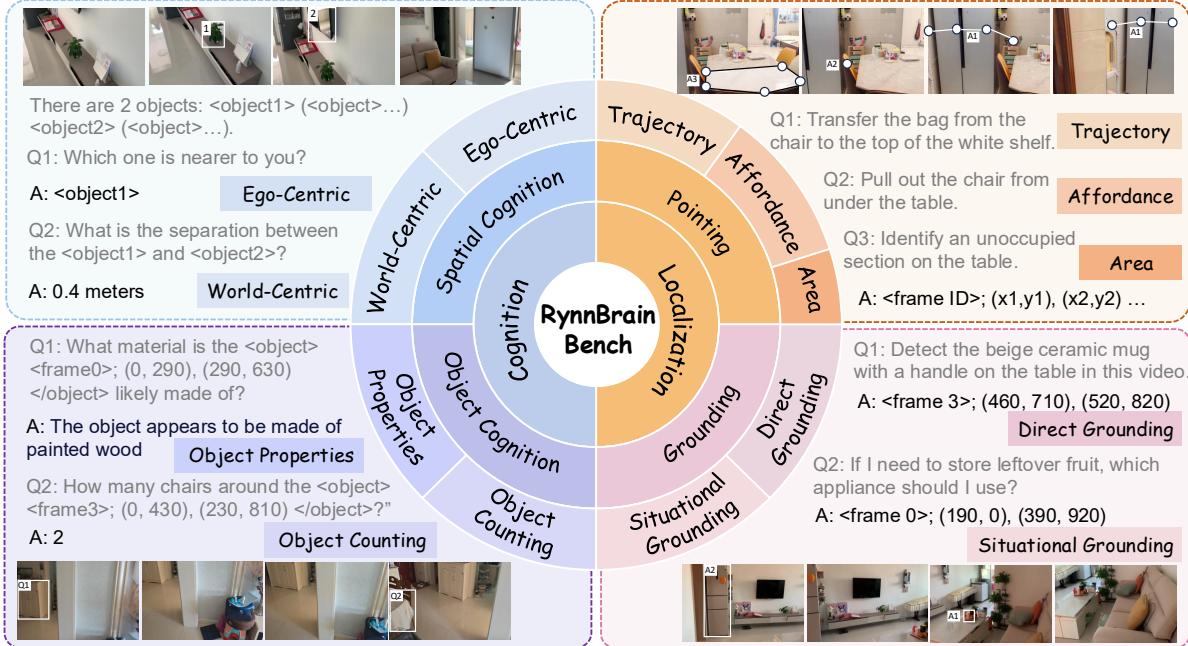


Figure 4 Overview of evaluation dimensions in RynnBrain-Bench. RynnBrain-Bench includes two subsets: cognition and location, evaluating a total of 21 spatio-temporal fine-grained embodied abilities.

outperforms prior models on most benchmarks. Notably, on VSI-Bench it improves over the previous best by 8.7%, on EgoTaskQA it yields a 10.5% gain, on Open-X VQA it surpasses prior methods by 6.6%, on RynnBrain-Object it improves by 20.2%, and on RynnBrain-Spatial it achieves a 25.1% gain. These results collectively demonstrate RynnBrain’s strong ability to perform embodied cognition and spatial reasoning across diverse tasks and environments.

6.3 Embodied Location Capability

We evaluate RynnBrain’s spatial grounding abilities across five key location tasks: object location, area location, affordance location, trajectory location, and grasp pose location. Our models are benchmarked against state-of-the-art methods on public benchmarks including RefSpatial-Bench [122], ShareRobot-Affordance [43], ShareRobot-Trajectory [43], Cornell-Grasp [19], and VMRD-Grasp [115]. As shown in Table 3, RynnBrain-8B achieves leading performance across all location benchmarks except ShareRobot-Trajectory, where RynnBrain-2B performs best. On RefSpatial-Bench, it achieves 59.2, surpassing the base model (Qwen3-VL) by 5.8%. It attains 44.7 on ShareRobot-Affordance, outperforming the closest competitor by 7.7%. For grasp pose location, RynnBrain-8B achieves 26.6 on Cornell-Grasp and 14.1 on VMRD-Grasp, significantly exceeding other 8B-scale models. Moreover, on our internal RynnBrain-Grounding and RynnBrain-Affordance benchmarks, RynnBrain-8B reaches 81.6 and 90.4, respectively, demonstrating its strong capability in precise spatio-temporal joint localization. The advantages are kept at the 30B (A3B) scale (Table 4). It achieves the best results on Cornell-Grasp (33.6), VMRD-Grasp (14.5), RynnBrain-Grounding (83.9), and RynnBrain-Affordance (90.5). Moreover, RynnBrain-30B (A3B) significantly outperforms all other models of the same scale on RefSpatial-Bench, RynnBrain-Area, and RynnBrain-Trajectory, approaching the performance of the much larger Gemini 3 Pro. These results demonstrate RynnBrain’s strong spatial grounding capabilities across multiple embodied location tasks and model scales.

6.4 General Visual Understanding

We further evaluate the general visual understanding ability of RynnBrain to assess its overall generality and generalization. To cover both static images and dynamic videos, we benchmark RynnBrain on a suite of general VQA datasets, including image-based AI2D [46], ChartQA [65], DocVQA [66], RealWorldQA [102], and InfoVQA [67], as well as video-based MVBench [53], EgoSchema [63], and VideoMME [32]. As shown in

Table 3 Comparison between models with parameter scales below 8B. * denotes results obtained from our own reproduction.

Benchmark		Model	RynnBrain	RynnBrain	MiMo-Embodied	RoboBrain 2.0	Pelican-VL	Cosmos-reason2	Qwen3-VL
			2B	8B	7B	7B	7B	8B	8B
Embodied Cognition	VSI-Bench		70.5	71.0	48.5	36.1	52.8	53.7*	60.3
	MMSI		34.1	39.6	30.2*	24.8*	26.2*	31.3*	29.6
	ERQA		42.3	46.8	46.8	36.5*	39.8*	46.0*	44.8
	RoboSpatial		65.7	73.1	61.8	54.2	57.5	59.0*	58.2
	EgoTaskQA		73.9	72.5	58.7*	51.1*	50.0*	55.7*	57.8
	EgoTextVQA _{indoor}		27.7/2.08	31.6/2.28	28.7/2.17*	22.0/1.79*	30.3/2.24*	26.5/1.96*	38.9/2.64*
	Open-X VQA		71.0	74.0	41.5*	44.6	44.1*	55.0*	59.8
	QAEgo4D		43.9	43.9	39.0*	39.7*	26.1*	46.9*	44.0
	MindCube		50.1	56.6	43.1*	38.9	33.7*	43.9*	36.0
Embodied Location	RynnBrain-Object		70.7	71.2	39.0	24.7	30.8	37.2	41.8
	RynnBrain-Spatial		57.2	59.9	28.3	13.5	20.5	31.4	35.0
	RefSpatial-Bench		52.7	59.2	48.0	32.5	22.3	33.1*	53.4
	ShareRobot-Affordance		43.3	44.7	35.8*	28.1	11.3	37.1*	37.0
	ShareRobot-Trajectory ↓		0.34	0.35	0.41*	0.55	0.42*	0.36*	0.37
	Cornell-Grasp		20.9	26.6	0.2*	0.0*	0.0*	18.1*	21.2*
	VMRD-Grasp		13.0	14.1	2.8*	0.5*	0.0*	13.7*	7.1*
	RynnBrain-Grounding		79.1	81.6	49.8	18.6	3.5	60.0	62.8
	RynnBrain-Area		54.6	56.2	49.4	38.0	46.5	37.6	30.0
General Visual Understanding	RynnBrain-Affordance		89.4	90.4	84.4	73.5	81.4	83.9	82.9
	RynnBrain-Trajectory		66.6	64.5	61.3	57.6	59.2	64.0	63.4
	AI2D		79.4	86.3	84.2	70.3*	83.8*	83.0*	85.7
	ChartQA		78.2	86.5	85.2*	82.4*	87.5*	84.3*	89.6
	DocVQA _{val}		93.0	96.2	94.9*	93.1*	94.5*	95.0*	96.4
	MVBench		67.3	69.5	57.9*	50.6*	67.7	67.0*	68.7
	RealWorldQA		60.4	67.3	66.1*	51.2*	67.1*	69.3*	71.5
	InfoVQA _{test}		71.2	83.4	72.0*	77.6*	81.1*	78.3*	83.1
	EgoSchema		64.0	69.7	58.2*	54.2*	73.3	63.5*	69.7
	VideoMME _{w/o sub}		61.4	70.7	65.0*	52.3*	63.3	71.9	71.4

Table 3, RynnBrain maintains the general visual understanding performance of the base model Qwen3-VL on both images and videos, and notably achieves state-of-the-art results on AI2D, MVbench, and InfoVQA, demonstrating the effectiveness of our training strategy. **Table 4** shows consistent trends for the 30B models, confirming that RynnBrain provides strong general visual capability alongside its embodied cognition strengths. This generalization advantage enables RynnBrain to serve as a central component of an embodied agent system, accommodating diverse task requirements.

6.5 Physically Grounded Reasoning

To rigorously evaluate the physically grounded reasoning capabilities of our model, we conducted a comparative analysis of RynnBrain-CoP-8B against several state-of-the-art multimodal baselines. The comparison includes leading open-source models such as InternVL3.5-8B [97], MiMo-Embodied-7B [37], and Qwen3-VL [6] (8B and 30B variants), alongside powerful proprietary models like GPT-5.2 and Gemini-3-Pro. Our evaluation focuses on three core embodied tasks—affordance prediction, area prediction, and trajectory prediction—which require the model to ground complex spatial intent into precise coordinates.

As shown in **Table 5**, RynnBrain-CoP-8B achieves superior performance across all evaluated metrics, setting a new state-of-the-art for embodied reasoning. On average, our 8B model attains a score of 73.8, surpassing

Table 4 Comparison between models with parameter scales above 30B. * denotes results obtained from our own reproduction.

Benchmark	Model	RynnBrain	RoboBrain 2.0	Pelican-VL	GPT-5.2	Gemini 3 Pro	Claude Sonnet 4.5	Qwen3-VL
		30B (A3B)	32B	72B	-	-	-	30B (A3B)
Embodied Cognition	VSI-Bench	74.5	42.7	57.3	46.6*	48.8*	42.5*	65.8
	MMSI	39.5	28.5*	30.7*	38.2*	49.2	28.9*	21.1
	ERQA	46.3	46.0	43.0	45.3*	70.5	60.0	43.0
	RoboSpatial	70.0	72.4	55.4	54.7*	56.0*	40.9*	55.4
	EgoTaskQA	78.9	59.9*	64.8*	59.6*	68.4*	50.9*	64.2*
	EgoTextVQA _{indoor}	34.6/2.39	30.5/2.24*	37.8/2.59*	49.6/3.02 *	45.5/2.87*	36.6/2.55*	41.3/2.74*
	Open-X VQA	83.4	28.6*	48.0*	43.6*	56.0*	41.9*	76.8*
	QAEgo4D	47.3	40.3*	24.6*	46.8*	42.1*	35.0*	47.3*
	MindCube	63.4	29.2	32.5*	61.7	70.8	58.3	39.0*
	RynnBrain-Object	73.3	26.2	42.2	53.1	44.6	25.1	42.6
	RynnBrain-Spatial	59.3	11.6	32.2	33.7	29.0	34.2	30.7
Embodied Location	RefSpatial-Bench	59.2	54.0	49.5	26.4*	65.5	15.1	53.1
	ShareRobot-Affordance	43.2	35.3	10.4*	17.5*	26.9*	13.9	47.2 *
	ShareRobot-Trajectory ↓	0.31	0.24	0.36*	0.35*	0.29*	0.57	0.36*
	Cornell-Grasp	33.6	0.3*	0.0*	14.5*	33.2*	0.0*	29.9*
	VMRD-Grasp	14.5	0.7*	0.0*	6.2*	10.9*	4.8*	8.0*
	RynnBrain-Grounding	83.9	0.0	10.8	11.2	59.2	0.0	76.4
	RynnBrain-Area	59.4	45.3	53.2	35.8	61.5	10.1	30.9
	RynnBrain-Affordance	90.5	76.1	87.3	83.3	86.0	48.7	86.2
	RynnBrain-Trajectory	66.8	60.3	64.1	70.5	72.0	54.6	61.2
General Visual Understand	AI2D	87.0	67.3*	86.7	97.1	98.7	91.5	85.0
	ChartQA	88.3	82.4*	90.4*	89.6	93.7	88.1	83.7*
	DocVQA	96.3	90.2*	95.2*	94.2	87.1	91.7	95.0
	MVBench	70.8	57.1*	69.7	67.1*	71.5*	55.1*	72.3
	RealworldQA	69.7	67.5*	67.3*	82.5*	73.6*	68.1	73.7
	InfoVQA _{test}	83.1	75.5*	89.1*	66.8*	83.1*	62.2*	82.0
	EgoSchema	66.8	61.3*	79.3	81.2*	72.2*	67.2*	70.7
	VideoMME _{w/o sub}	71.9	55.6*	73.7*	84.7	88.6	68.6*	74.5

the strongest proprietary competitor, MiMo-Embodied-7B (65.8) and Gemini-3-Pro (65.1), by a substantial margin. Notably, it outperforms the much larger RoboBrain2.0-32B (57.7) by 16.1%, demonstrating that our reasoning architecture is more effective than simple parameter scaling for spatial tasks.

The task-specific results further highlight the model’s precision:

- **Affordance Prediction:** RynnBrain-CoP-8B achieves a peak accuracy of 90.3, being the only model to break the 90 threshold. This suggests that the physically grounded CoT effectively narrows down actionable zones.
- **Area Prediction:** While this remains the most challenging task for all baselines (with many scoring below 40), our model reaches 59.6, outperforming Gemini-3-Pro (50.7) and nearly doubling the performance of Qwen3-VL-30B (33.0).
- **Trajectory Prediction:** Our model leads with 71.2, showcasing a superior understanding of temporal-spatial sequences compared to GPT-5.2 (70.5) and InternVL3.5 (47.8).

These results validate that despite its compact 8B parameter size, RynnBrain-CoP-8B delivers consistently more accurate spatial grounding. The significant gains, particularly in complex area and trajectory tasks,

Table 5 Comparison RynnBrain-CoP with state-of-the-art thinking models on embodied reasoning tasks. All compared models are evaluated with the thinking mode enabled. We evaluate models on affordance prediction, area prediction, and trajectory prediction. Average denotes the mean across the three tasks. Best results are highlighted.

Model	Task			
	Affordance	Area	Trajectory	Average
InternVL3.5-8B [97]	63.1	9.2	47.8	40.0
MiMo-Embodied-7B [37]	85.3	47.1	64.9	65.8
RoboBrain2.0-7B [92]	65.3	38.0	58.5	53.9
RoboBrain2.0-32B [92]	73.2	39.5	60.5	57.7
Qwen3-VL-8B-Thinking [6]	56.7	20.4	46.9	41.3
Qwen3-VL-30B-A3B-Thinking [6]	62.2	33.0	54.8	50.0
GPT-5.2 [89]	83.3	35.8	70.5	63.2
Gemini-3-Pro	83.9	50.7	60.6	65.1
RynnBrain-CoP-8B	90.3	59.6	71.2	73.8

Table 6 Comparison RynnBrain-Nav with state-of-the-art navigation models. The best results are highlighted.

Method	Observation Encoder				R2R Val-Unseen				RxR Val-Unseen			
	Pano.	Odo.	Depth	S.RGB	NE↓	OS↑	SR↑	SPL↑	NE↓	SR↑	SPL↑	nDTW↑
VLN○BERT* [40]	✓	✓	✓		5.74	53.0	44.0	39.0	8.98	27.0	22.6	46.7
ETPNav* [2]	✓	✓	✓		4.71	65.0	57.0	49.0	5.64	54.7	44.8	61.9
ScaleVLN* [99]	✓	✓	✓		4.80	—	55.0	51.0	—	—	—	—
R2R-CMTP [14]	✓	✓	✓		7.90	38.0	26.4	22.7	—	—	—	—
LAW [80]		✓	✓	✓	6.83	44.0	35.0	31.0	10.90	8.0	8.0	38.0
ETPNav + FF [98]		✓	✓	✓	5.95	55.8	44.9	30.4	8.79	25.5	18.1	—
Seq2Seq [47]			✓	✓	7.77	37.0	25.0	22.0	12.10	13.9	11.9	30.8
CMA [47]			✓	✓	7.37	40.0	32.0	30.0	—	—	—	—
VLN-R1 [74]				✓	5.47	49.1	37.4	35.9	—	—	—	—
NaVid [17]				✓	5.47	49.1	37.4	35.9	—	—	—	—
NaVILA [17]				✓	5.22	62.5	54.0	49.0	6.77	49.3	44.0	58.5
UniNaVid [116]				✓	5.58	53.3	47.0	42.7	6.24	48.7	40.9	—
StreamVLN [100]				✓	4.98	64.2	56.9	51.9	6.22	52.9	46.0	61.9
RynnBrain-Nav-8B				✓	4.92	71.6	58.6	49.6	6.20	56.1	49.6	59.6

prove that interleaving multi-step reasoning thoughts with visual coordinates is a more data-efficient and hallucination-resistant paradigm for embodied agents than traditional purely text-based reasoning paradigm.

6.6 Vision-Language Navigation

Benchmarks and Metrics. We evaluate our finetuned model on two public VLN-CE benchmarks [47]: R2R-CE [4] and RxR-CE [50], which simulate continuous navigation in photorealistic Matterport3D scenes using the Habitat simulator. To assess generalization to novel environments, all experiments are conducted on the validation unseen splits. Following standard protocols, we report performance using metrics for task completion (Success Rate, SR), path efficiency (Success-weighted by Path Length, SPL), and path fidelity (normalized Dynamic Time Warping, nDTW). The nDTW metric, specifically, leverages the ground-truth trajectories to evaluate how closely the agent’s path follows the reference instruction. We also include Navigation Error (NE) and Oracle Success Rate (OS) for a comprehensive analysis.

Comparison wth State-of-The-Art. Table 6 summarizes the performance of our method on the VLN-CE R2R and RxR benchmarks under the Val-Unseen setting, compared with existing SOTA methods. For the R2R-CE benchmark, the RynnBrain-Nav-8B model demonstrates highly competitive performance even compared to methods utilizing multiple input types like panoramic views and odometry. Achieving a top-ranked SR of 58.6% and the second-best SPL of 49.6% and the lowest NE of 4.92. Noticing that our model’s OS reaching 71.6%

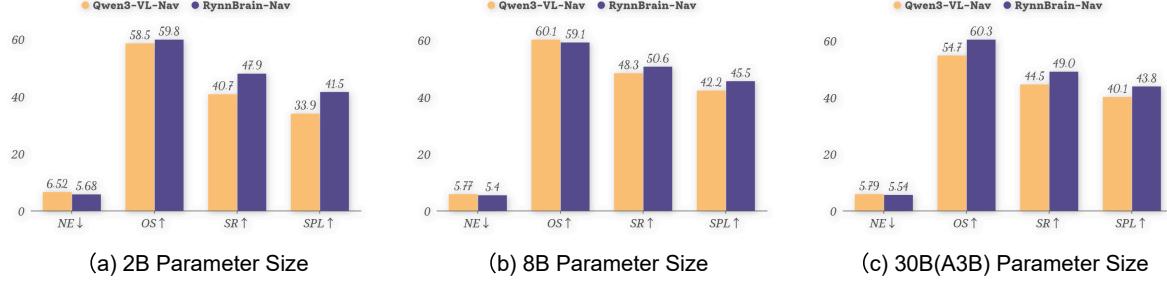


Figure 5 Compare the differences in the ability of Qwen3-VL and RynnBrain as the base model to finetune navigation models under multiple model scales. All results are reported without performing multiple rounds of DAgger.

exceeds all competitors, including topological prediction methods that utilize panoramic observations. This contrast between our high OS and lower SR indicates that our model is proficient at coarse-level navigation but lacks the precision for the terminal stopping maneuver, thereby failing the overall task.

The model’s navigation capability is further validated on the more demanding RxR benchmark. Here, RynnBrain-Nav-8B again secures a top-ranked SR of 56.1% and the lowest NE of 4.92, highlighting its superior capability in complex, long-horizon navigation tasks.

Ablation Study on Pre-training Efficacy. To isolate and evaluate the contribution of our pretraining, we conduct a comparative analysis between RynnBrain and its Qwen3-VL [6] baseline. Models of varying scales from both families are fine-tuned on the same sample of datasets (R2R, RxR, EnvDrop, ScaleVLN), and their performance is evaluated on the R2R-CE benchmark. The results, shown in Figure 5, demonstrate the benefit of the RynnBrain pretraining. RynnBrain-Nav demonstrates clear performance superiority over the Qwen3-VL counterpart, achieving consistently higher SR and SPL scores across all evaluated scales. Notably, our 2B RynnBrain-Nav model surpasses its 2B Qwen3-VL counterpart by a substantial 7.2% in SR and 7.6% in SPL, affirming the clear efficacy of our pretraining approach.

Impact of Model Scale and Architecture. Our analysis reveals a clear scaling trend for dense architectures. As shown in Figure 5, both RynnBrain-Nav and Qwen3-VL demonstrate improved SR and SPL when scaling from 2B to 8B parameters. However, this positive scaling did not extend to the Mixture-of-Experts (MoE) architecture. Despite its larger total parameter count, the 30B MoE model (3B active) failed to outperform the 8B dense models during initial training phases. This suggests that the sparse activation mechanism of MoE may not be fully leveraged by the Visual Language Navigation (VLN) task, or that alternative training strategies are required to unlock its scaling potential.

Multi-Turn DAgger Training. To further enhance navigation performance, we employ multi-turn DAgger [82] training. After initial SFT, the agent collects new trajectories from the R2R, RxR, and EnvDrop environments. This data is then combined with the original datasets to retrain the model. This iterative process proved highly effective, particularly in the initial rounds: the Success Rate (SR) increased from a 50.6% baseline to 56.4% after the first iteration and further to 58.5% after the second. However, the third DAgger iteration yielded only a marginal improvement, indicating a clear trend of diminishing returns as the agent’s policy converges.

6.7 Planning and Manipulation

We develop a three-stage evaluation system to rigorously assess the hierarchical manipulation system based on RynnBrain. In the first setting, we evaluate the planning logic in isolation: our model serves as the high-level planner, while a human operator equipped with a Universal Manipulation Interface (UMI) [18] acts as a fully reliable low-level controller. In the second setting, we design three real-robot experiments in complex, multi-objective scenarios to validate the precise manipulation capabilities of RynnBrain-VLA. In the third setting, we assess end-to-end autonomy by deploying the integrated system on a Franka robot. Throughout the entire system, **RynnBrain-Plan** is responsible for comprehending the scene and high-level tasks and then generating sub-tasks with precise coordinates. **RynnBrain-VLA** accepts sub-tasks and controls the robot arm

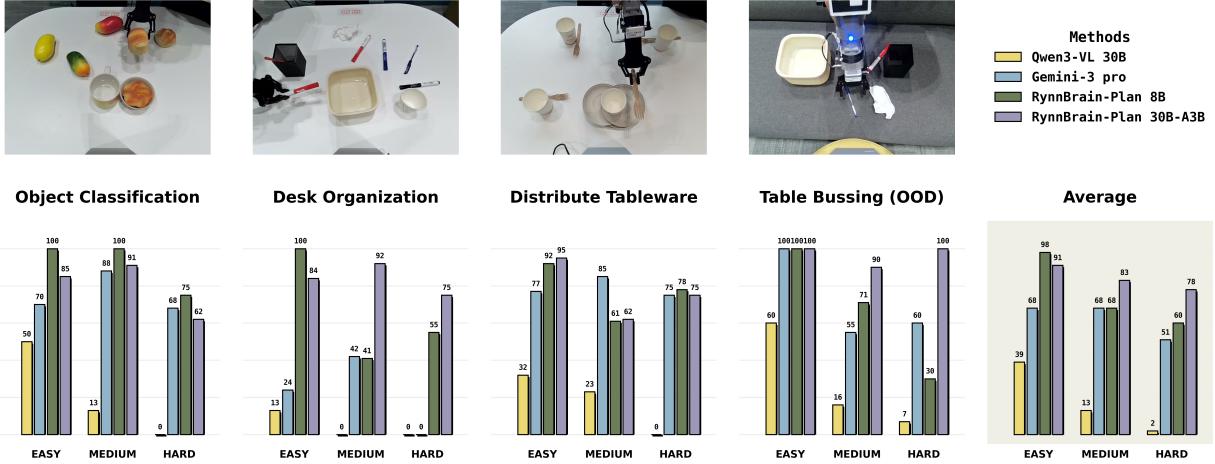


Figure 6 Comparison to other VLMs. RynnBrain-Plan-30B outperforms other methods on almost all the settings, except for the medium difficulty of Distribute Tableware. The metric is Task Process (TP ↑).

to perform the low-level tasks.

6.7.1 RynnBrain and UMI Hierarchical Evaluation

Experimental settings. We designed four long-horizon planning tasks: Object Classification, Desk Organization, Distribute Tableware, and Table Bussing. Among them, the first three are in-distribution tasks, whereas the last one is an out-of-distribution task. To assess performance across varying degrees of complexity, all tasks are stratified into three difficulty levels: Easy, Medium, and Hard. As the difficulty level increases, both the scene complexity and the instruction complexity rise accordingly. For the fine-tuning phase, we collected 100 expert demonstrations for each of the in-distribution tasks. Detailed descriptions and specifications for each task are provided in the Appendix. We benchmark our method, RynnBrain-Plan, against two state-of-the-art baselines: Gemini-3 Pro, and Qwen3-VL 30B. To mitigate randomness, each task–model evaluation is repeated five times and we report the average results. Following the protocol established in [87], we adopt Task Progress—defined as the percentage of subtasks successfully completed by the end of the episode—as our primary evaluation metric. To guarantee reliability and consistency, all assessments are conducted by trained human annotators.

Comparison on ID Tasks. As quantitatively illustrated in Figure 6, our Rynnbrain-Plan demonstrates a significant performance advantage over state-of-the-art baselines across varying difficulty levels. In the in-distribution tasks (*Object Classification*, *Desk Organization*, and *Distribute Tableware*), our Rynnbrain-Plan-30B-A3B consistently achieves superior task progress. This advantage is particularly pronounced in the “Hard” difficulty settings, which require complex long-horizon reasoning. For instance, in the *Desk Organization* task (Hard), while Qwen3-VL and Gemini-3 Pro fail to make meaningful progress (near 0% completion), our 30B model maintains a robust completion rate of over 75%. While Gemini 3 Pro shows competitive performance in simpler scenarios (e.g., *Distribute Tableware* - Medium), it suffers from severe performance degradation as task complexity increases. The Rynnbrain-Plan 8B model also delivers strong results in “Easy” and “Medium” settings, often surpassing the significantly larger Qwen3-VL 30B, highlighting the efficiency of our data construction strategy.

Generalization Analysis on OOD Tasks. The results on the out-of-distribution (OOD) task, *Table Bussing*, highlight the exceptional generalization capabilities of our approach. Despite not being exposed to this specific task during fine-tuning, Rynnbrain-Plan 30B (A3B) achieves remarkable success, reaching near 100% task progress across all difficulty levels. This stands in stark contrast to the baselines; for example, in the “Hard” setting of *Table Bussing*, Qwen3-VL completely fails (< 10%), and Gemini-3 Pro achieves only moderate success (~ 60%). Crucially, comparing our two model variants reveals that while the 8B model generalizes well in simple OOD scenarios, the larger 30B model possesses the emergent capacity to handle complex, unseen constraints, effectively bridging the gap between in-domain planning and open-world adaptability.

Ablation Study. To rigorously validate the effectiveness of fine-tuning with multi-turn dialogue data, we

Table 7 Ablation on multi-turn dialogue data. Training with single-turn dialogue and multi-turn dialogue data is short for ST and MT. The metric is Task Process (TP \uparrow).

Method	Object Classification			Desk Organization			Distribute Tableware			Table Bussing		
	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
RynnBrain-Plan-ST 8B	72	20	0	60	0	0	34	0	0	90	0	0
RynnBrain-Plan-ST 30B	75	30	0	58	10	0	28	0	0	95	0	0
RynnBrain-Plan-MT 8B	100	100	75	100	41	55	92	61	78	100	71	30
RynnBrain-Plan-MT 30B	85	91	62	84	92	75	95	62	75	100	90	100



Figure 7 Tasks for VLA evaluation.

conducted an ablation experiment by training a variant of RynnBrain-Plan exclusively on single-turn dialogue samples. As presented in the [Table 7](#), the performance of this single-turn baseline degrades significantly. It only manages to complete tasks in the “Easy” difficulty setting, yet even in these simple scenarios, the success rate remains prohibitively low. This sharp decline underscores the necessity of temporal context: without the multi-turn interaction history, the model struggles to maintain state consistency over time. In contrast, the model fine-tuned on multi-turn data effectively leverages historical actions to ground its reasoning, leading to substantially more accurate and coherent action predictions.

6.7.2 VLA Evaluation.

We benchmark RynnBrain-VLA in three multi-object scenarios to evaluate its object manipulation capabilities. As illustrated in [Figure 7](#), the experimental setup includes two fundamental tasks featuring four manipulable objects from two categories (with one category dominant), and a more challenging task involving six objects with a balanced category distribution. Each model was tested over ten trials per task, with object arrangements and target selections randomized for each run. To provide a comprehensive analysis, we employed three evaluation metrics: (1) Pickup Success Rate (PSR): The percentage of trials where any object was successfully grasped, regardless of its identity. (2) Recognition Success Rate (RSR): The accuracy in identifying the target object, defined by whether the gripper makes initial contact with the correct item. (3) Success Rate (SR): The overall rate of successfully picking up the designated target object. For comparative analysis, we fine-tuned two baseline models: (1) π -0.5 [9]: In order to enable it to manipulate specific objects, we adapted its input by appending the initial task frame and employing a consistent text format as RynnBrain-VLA. (2) Qwen3-VL [6]: This model was fine-tuned using the same architectural configuration and data format as RynnBrain-VLA to ensure a fair comparison.

As indicated in [Table 8](#), the general VLA π -0.5 struggles to identify target objects, resulting in a low RSR. This performance bottleneck stems from the limited capacity for fine-grained image-text alignment. In contrast, while Qwen3-VL-Finetuned is derived directly from a general VLM, RynnBrain-VLA achieves superior localization accuracy and higher grasping success rates. We attribute this advantage to our extensive pretraining on embodied pointing tasks. Overall, RynnBrain-VLA demonstrates significantly improved success rates, notably without necessitating extensive pretraining on specific action modalities.

6.7.3 RynnBrain and VLA Hierarchical Evaluation

To validate the long-horizon planning and manipulation capabilities of our proposed framework, we integrate RynnBrain-Plan with RynnBrain-VLA to construct a hierarchical manipulation system. In this architecture, RynnBrain-Plan functions as the high-level planner, decomposing complex instructions into executable

Table 8 VLA evaluation results.

Method	Pick up bread			Pick up bowl			Mixed			Overall		
	PSR	RSR	SR	PSR	RSR	SR	PSR	RSR	SR	PSR	RSR	SR
$\pi_{0.5}$ -Finetuned	0.7	0.6	0.5	0.8	0.5	0.5	0.5	0.6	0.4	0.67	0.57	0.47
Qwen3-VL-Finetuned	0.7	1.0	0.7	0.5	1.0	0.5	0.6	1.0	0.6	0.60	1.00	0.60
RynnBrain-VLA	0.8	1.0	0.8	0.7	1.0	0.7	0.9	0.9	0.8	0.8	0.97	0.77

subtasks, while RynnBrain-VLA acts as the low-level controller, generating precise robot action commands. As illustrated in the qualitative results provided in the Figure 8, these two modules are effectively integrated, demonstrating robust performance in completing long-horizon manipulation tasks. We also evaluate a comparative setup in which Gemini generates purely textual plans that are subsequently executed by $\pi_{0.5}$. This paradigm often leads to grasping and placement mismatches in tasks involving multiple identical objects or requiring precise placement. This clearly demonstrates the significance of our physical-aware planning mode in complex scenarios and intricate operational tasks.

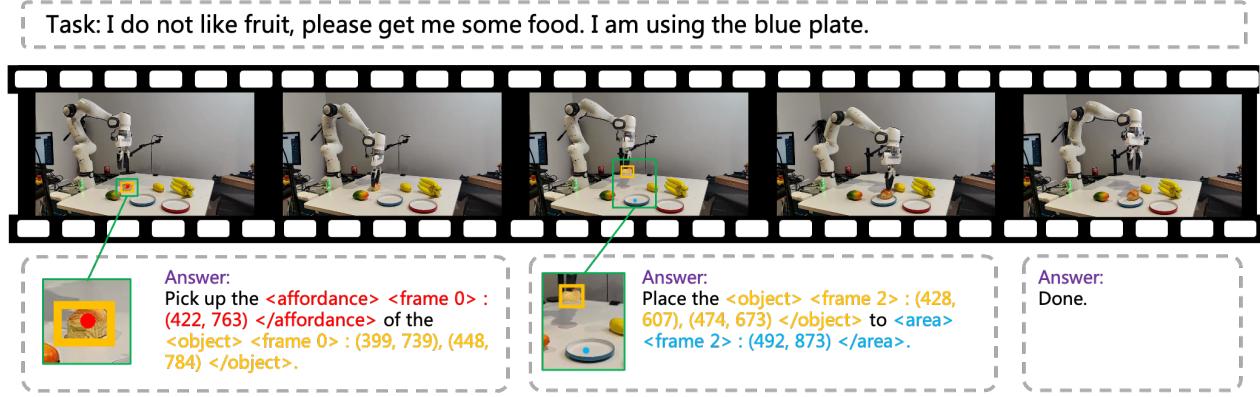


Figure 8 Planning Manipulation Video Examples of RynnBrain-Plan. An example of the RynnBrain-Plan model on a multi-step online planning task. The executor is RynnBrain-VLA.

7 Conclusion and Future Works

In this study, we introduce **RynnBrain**, a suite of advanced embodied foundation models. RynnBrain expands the capability frontier of embodied foundation models along four axes: egocentric cognition, spatio-temporal localization, physically grounded reasoning, and physics-aware planning. Across a comprehensive evaluation on 28 benchmarks, RynnBrain consistently emerges—at all model scales—as a highly capable and well-rounded open embodied foundation model. Building on the RynnBrain foundation models, we further post-train four specialized variants—RynnBrain-CoP, RynnBrain-Nav, RynnBrain-Plan, and RynnBrain-VLA—each achieving state-of-the-art performance in its respective domain and collectively demonstrating the substantial value of RynnBrain pretraining for a wide range of embodied tasks. Beyond model development, we introduce **RynnBrain-Bench**, a high-dimensional evaluation suite designed to rigorously assess fine-grained spatio-temporal cognition and localization in embodied settings. RynnBrain-Bench advances existing benchmarks by emphasizing video understanding across episodes, precise spatio-temporal grounding, and physically meaningful pointing behaviors, providing a more faithful measure of embodied reasoning capabilities in real-world environments.

Looking forward, we view RynnBrain as a key engine for advancing multimodal foundation models into the physical world. Future embodied intelligence systems will likely comprise a holistic agent stack, including components such as a brain, cerebellum, memory modules, and a sensorimotor interface. RynnBrain is positioned to serve as a core foundation of this agent system, enabling efficient exploration, autonomous decision-making, and dynamic interaction in complex physical environments. By openly releasing the full

model family under the Apache 2.0 license, we hope to empower the community to address broader embodied scenarios with RynnBrain and accelerate progress toward general embodied intelligence.

References

- [1] Env-qa: A video question answering benchmark for comprehensive understanding of dynamic environments, author=Gao, Difei and Wang, Ruiping and Bai, Ziyi and Chen, Xilin, journal=Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), month=October, year=2021, pages = 1675-1685 .
- [2] Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. Etpnav: Evolving topological planning for vision-language navigation in continuous environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [3] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018.
- [4] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018.
- [5] 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.
- [6] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. <https://arxiv.org/abs/2511.21631>.
- [7] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- [8] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0 : A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.
- [9] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y Galliker, et al. $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization. In *9th Annual Conference on Robot Learning*, 2025.
- [10] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015.
- [11] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018.
- [12] Zhongang Cai, Ruiqi Wang, Chenyang Gu, Fanyi Pu, Junxiang Xu, Yubo Wang, Wanqi Yin, Zhitao Yang, Chen Wei, Qingping Sun, et al. Scaling spatial intelligence with multimodal foundation models. *arXiv preprint arXiv:2511.13719*, 2025.
- [13] 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.

[14] Kevin Chen, Junshan K Chen, Jo Chuang, Marynel Vázquez, and Silvio Savarese. Topological planning with transformers for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11276–11286, 2021.

[15] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Zhenyu Tang, Li Yuan, et al. Sharegpt4video: Improving video understanding and generation with better captions. *Advances in Neural Information Processing Systems*, 37:19472–19495, 2024.

[16] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1971–1978, 2014.

[17] An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. *arXiv preprint arXiv:2412.04453*, 2024.

[18] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

[19] F. Chu, R. Xu, and P. A. Vela. Real-world multiobject, multigrasp detection. volume 3, pages 3355–3362, Oct 2018. doi: 10.1109/LRA.2018.2852777.

[20] Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Mohammadreza Salehi, Rohun Tripathi, Sangho Lee, Zhongzheng Ren, Chris Dongjoo Kim, Yinuo Yang, et al. Molmo2: Open weights and data for vision-language models with video understanding and grounding. *arXiv preprint arXiv:2601.10611*, 2026.

[21] Open X-Embodiment Collaboration, Abby O'Neill, and et.al. Open X-Embodiment: Robotic learning datasets and RT-X models. <https://arxiv.org/abs/2310.08864>, 2023.

[22] AgiBot World Colosseum contributors. Agibot world colosseum. <https://github.com/OpenDriveLab/AgiBot-World>, 2024.

[23] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European conference on computer vision (ECCV)*, pages 720–736, 2018.

[24] Ronghao Dang, Yuqian Yuan, Yunxuan Mao, Kehan Li, Jiangpin Liu, Zhikai Wang, Xin Li, Fan Wang, and Deli Zhao. Rynnec: Bringing mllms into embodied world. *arXiv preprint arXiv:2508.14160*, 2025.

[25] Ronghao Dang, Yuqian Yuan, Wenqi Zhang, Yifei Xin, Boqiang Zhang, Long Li, Liuyi Wang, Qinyang Zeng, Xin Li, and Lidong Bing. Ecbench: Can multi-modal foundation models understand the egocentric world? a holistic embodied cognition benchmark. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

[26] Chengqi Duan, Rongyao Fang, Yuqing Wang, Kun Wang, Linjiang Huang, Xingyu Zeng, Hongsheng Li, and Xihui Liu. Got-r1: Unleashing reasoning capability of mllm for visual generation with reinforcement learning. *arXiv preprint arXiv:2505.17022*, 2025.

[27] Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Dilin Wang, Zhicheng Yan, et al. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction. *arXiv preprint arXiv:2505.20279*, 2025.

[28] Huang Fang, Mengxi Zhang, Heng Dong, Wei Li, Zixuan Wang, Qifeng Zhang, Xueyun Tian, Yucheng Hu, and Hang Li. Robix: A unified model for robot interaction, reasoning and planning. *arXiv preprint arXiv:2509.01106*, 2025.

[29] Miquel Farré, Andi Marafioti, Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. Finevideo. <https://huggingface.co/datasets/HuggingFaceFV/finevideo>, 2024.

[30] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025.

[31] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[32] Chaoyou Fu, Yuhang Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24108–24118, 2025.

[33] Xingyu Fu, Minqian Liu, Zhengyuan Yang, John Corring, Yijuan Lu, Jianwei Yang, Dan Roth, Dinei Florencio, and Cha Zhang. Refocus: Visual editing as a chain of thought for structured image understanding. *arXiv preprint arXiv:2501.05452*, 2025.

[34] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18995–19012, 2022.

[35] Ziyu Guo, Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Rui Huang, Haoquan Zhang, Manyuan Zhang, Jiaming Liu, Shanghang Zhang, Peng Gao, et al. Can we generate images with cot? let's verify and reinforce image generation step by step. *arXiv preprint arXiv:2501.13926*, 2025.

[36] 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.

[37] 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.

[38] Haibin He, Maoyuan Ye, Jing Zhang, Juhua Liu, Bo Du, and Dacheng Tao. Gomatching: A simple baseline for video text spotting via long and short term matching. *Advances in Neural Information Processing Systems*, 37: 25663–25686, 2024.

[39] Haibin He, Jing Zhang, Maoyuan Ye, Juhua Liu, Bo Du, and Dacheng Tao. Gomatching++: Parameter-and data-efficient arbitrary-shaped video text spotting and benchmarking. *arXiv preprint arXiv:2505.22228*, 2025.

[40] Yicong Hong, Zun Wang, Qi Wu, and Stephen Gould. Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15439–15449, 2022.

[41] Jie Huang, Xuejing Liu, Sibo Song, Ruibing Hou, Hong Chang, Junyang Lin, and Shuai Bai. Revisiting multimodal positional encoding in vision-language models. *arXiv preprint arXiv:2510.23095*, 2025.

[42] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[43] Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. *arXiv preprint arXiv:2502.21257*, 2025.

[44] Baoxiong Jia, Ting Lei, Song-Chun Zhu, and Siyuan Huang. Egotaskqa: Understanding human tasks in egocentric videos. *Advances in Neural Information Processing Systems*, 35:3343–3360, 2022.

[45] 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)*, 2014.

[46] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pages 235–251. Springer, 2016.

[47] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pages 104–120. Springer, 2020.

[48] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

[49] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. *arXiv preprint arXiv:2010.07954*, 2020.

[50] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. *arXiv preprint arXiv:2010.07954*, 2020.

[51] Sicong Leng, Jing Wang, Jiaxi Li, Hao Zhang, Zhiqiang Hu, Boqiang Zhang, Yuming Jiang, Hang Zhang, Xin Li, Lidong Bing, et al. Mmr1: Enhancing multimodal reasoning with variance-aware sampling and open resources. *arXiv preprint arXiv:2509.21268*, 2025.

[52] 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.

[53] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024.

[54] Yitang Li, Yuanhang Zhang, Wenli Xiao, Chaoyi Pan, Haoyang Weng, Guanqi He, Tairan He, and Guanya Shi. Hold my beer: Learning gentle humanoid locomotion and end-effector stabilization control. In *RSS 2025 Workshop on Whole-body Control and Bimanual Manipulation: Applications in Humanoids and Beyond*.

[55] Long Lian, Yifan Ding, Yunhao Ge, Sifei Liu, Hanzi Mao, Boyi Li, Marco Pavone, Ming-Yu Liu, Trevor Darrell, Adam Yala, et al. Describe anything: Detailed localized image and video captioning. *arXiv preprint arXiv:2504.16072*, 2025.

[56] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[57] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[58] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[59] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, et al. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024.

[60] Zixian Ma, Jianguo Zhang, Zhiwei Liu, Jieyu Zhang, Juntao Tan, Manli Shu, Juan Carlos Niebles, Shelby Heinecke, Huan Wang, Caiming Xiong, et al. Taco: Learning multi-modal models to reason and act with synthetic chains-of-thought-and-action. In *Workshop on Reasoning and Planning for Large Language Models*.

[61] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Videogpt+: Integrating image and video encoders for enhanced video understanding. *arXiv preprint arXiv:2406.09418*, 2024.

[62] 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*, 2024.

[63] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

[64] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.

[65] Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pages 2263–2279, 2022.

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

[67] Minesh Mathew, Viraj Bagal, Rubén Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022.

[68] Lingchen Meng, Jianwei Yang, Rui Tian, Xiyang Dai, Zuxuan Wu, Jianfeng Gao, and Yu-Gang Jiang. Deepstack: Deeply stacking visual tokens is surprisingly simple and effective for lmms. *Advances in Neural Information Processing Systems*, 37:23464–23487, 2024.

[69] Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16695–16705, 2025.

[70] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017.

[71] OpenAI. Introducing GPT-5.2, December 2025. <https://openai.com/index/introducing-gpt-5-2/>.

[72] Alkesh Patel, Vibhav Chitalia, and Yinfei Yang. Advancing egocentric video question answering with multimodal large language models. *arXiv preprint arXiv:2504.04550*, 2025.

[73] Baoqi Pei, Yifei Huang, Jilan Xu, Yuping He, Guo Chen, Fei Wu, Yu Qiao, and Jiangmiao Pang. Egothinker: Unveiling egocentric reasoning with spatio-temporal cot. *arXiv preprint arXiv:2510.23569*, 2025.

[74] Zhangyang Qi, Zhixiong Zhang, Yizhou Yu, Jiaqi Wang, and Hengshuang Zhao. Vln-r1: Vision-language navigation via reinforcement fine-tuning. *arXiv preprint arXiv:2506.17221*, 2025.

[75] 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.

[76] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, et al. Paco: Parts and attributes of common objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7141–7151, 2023.

[77] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3505–3506, 2020.

[78] 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. *arXiv preprint arXiv:2408.00714*, 2024.

[79] Ruchit Rawal, Khalid Saifullah, Miquel Farré, Ronen Basri, David Jacobs, Gowthami Somepalli, and Tom Goldstein. Cinepile: A long video question answering dataset and benchmark. *arXiv preprint arXiv:2405.08813*, 2024.

[80] Sonia Raychaudhuri, Saim Wani, Shivansh Patel, Unnat Jain, and Angel Chang. Language-aligned waypoint (law) supervision for vision-and-language navigation in continuous environments. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 4018–4028, 2021.

[81] Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, et al. Grounding dino 1.5: Advance the "edge" of open-set object detection. *arXiv preprint arXiv:2405.10300*, 2024.

[82] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011.

[83] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019.

[84] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[85] Pierre Sermanet, Tianli Ding, Jeffrey Zhao, Fei Xia, Debidatta Dwibedi, Keerthana Gopalakrishnan, Christine Chan, Gabriel Dulac-Arnold, Sharath Maddineni, Nikhil J Joshi, et al. Robovqa: Multimodal long-horizon reasoning for robotics. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 645–652. IEEE, 2024.

[86] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[87] Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolò Fusai, Adrian Li-Bell, Danny Driess, Lachy Groom, Sergey Levine, and Chelsea Finn. Hi robot: Open-ended instruction following with hierarchical vision-language-action models, 2025. <https://arxiv.org/abs/2502.19417>.

[88] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Charades-ego: A large-scale dataset of paired third and first person videos. *arXiv preprint arXiv:1804.09626*, 2018.

[89] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

[90] 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. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15768–15780, 2025.

[91] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. *arXiv preprint arXiv:1904.04195*, 2019.

[92] BAAI RoboBrain Team, Mingyu Cao, Huajie Tan, Yuheng Ji, Minglan Lin, Zhiyu Li, Zhou Cao, Pengwei Wang, Enshen Zhou, Yi Han, et al. Robobrain 2.0 technical report. *arXiv preprint arXiv:2507.02029*, 2025.

[93] Gemini Team, Rohan Anil, Sébastien Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[94] 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.

[95] Yuchuang Tong, Haotian Liu, and Zhengtao Zhang. Advancements in humanoid robots: A comprehensive review and future prospects. *IEEE/CAA Journal of Automatica Sinica*, 11(2):301–328, 2024.

[96] An Dinh Vuong, Minh Nhat Vu, Hieu Le, Baoru Huang, Huynh Thi Thanh Binh, Thieu Vo, Andreas Kugi, and Anh Nguyen. Grasp-anything: Large-scale grasp dataset from foundation models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14030–14037. IEEE, 2024.

[97] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[98] Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Sim-to-real transfer via 3d feature fields for vision-and-language navigation. *arXiv preprint arXiv:2406.09798*, 2024.

[99] Zun Wang, Jialu Li, Yicong Hong, Yi Wang, Qi Wu, Mohit Bansal, Stephen Gould, Hao Tan, and Yu Qiao. Scaling data generation in vision-and-language navigation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12009–12020, 2023.

[100] Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, et al. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025.

[101] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. <https://www.aclweb.org/anthology/2020.emnlp-demos.6>.

[102] xai. Realworldqa benchmark. <https://huggingface.co/datasets/xai-org/RealworldQA>, 2024.

[103] Yi Xu, Chengzu Li, Han Zhou, Xingchen Wan, Caiqi Zhang, Anna Korhonen, and Ivan Vulić. Visual planning: Let's think only with images. *arXiv preprint arXiv:2505.11409*, 2025.

[104] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[105] 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. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10632–10643, 2025.

[106] Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis Brown, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, et al. Cambrian-s: Towards spatial supersensing in video. *arXiv preprint arXiv:2511.04670*, 2025.

[107] Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, et al. Mmsi-bench: A benchmark for multi-image spatial intelligence. *arXiv preprint arXiv:2505.23764*, 2025.

[108] Zuhao Yang, Sudong Wang, Kaichen Zhang, Keming Wu, Sicong Leng, Yifan Zhang, Bo Li, Chengwei Qin, Shijian Lu, Xingxuan Li, and Lidong Bing. Longvt: Incentivizing "thinking with long videos" via native tool calling. *arXiv preprint arXiv:2511.20785*, 2025.

[109] Baiqiao Yin, Qineng Wang, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, et al. Spatial mental modeling from limited views. In *Structural Priors for Vision Workshop at ICCV'25*, 2025.

[110] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European conference on computer vision*, pages 69–85. Springer, 2016.

[111] 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, 2025. <https://arxiv.org/abs/2505.08548>.

[112] Yuqian Yuan, Wentong Li, Jian Liu, Dongqi Tang, Xinjie Luo, Chi Qin, Lei Zhang, and Jianke Zhu. Osprey: Pixel understanding with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28202–28211, 2024.

[113] Yuqian Yuan, Hang Zhang, Wentong Li, Zesen Cheng, Boqiang Zhang, Long Li, Xin Li, Deli Zhao, Wenqiao Zhang, Yuetong Zhuang, et al. Videorefer suite: Advancing spatial-temporal object understanding with video llm. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18970–18980, 2025.

[114] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025.

[115] Hanbo Zhang, Xuguang Lan, Site Bai, Xinwen Zhou, Zhiqiang Tian, and Nanning Zheng. Roi-based robotic grasp detection for object overlapping scenes. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4768–4775. IEEE, 2019.

[116] Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. *arXiv preprint arXiv:2412.06224*, 2024.

[117] 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.

[118] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

[119] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Sglang: efficient execution of structured language model programs. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pages 62557–62583, 2024.

[120] Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing "thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025.

- [121] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.
- [122] Enshen Zhou, Jingkun An, Cheng Chi, Yi Han, Shanyu Rong, Chi Zhang, Pengwei Wang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, et al. Roborefer: Towards spatial referring with reasoning in vision-language models for robotics. *arXiv preprint arXiv:2506.04308*, 2025.
- [123] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- [124] Sheng Zhou, Junbin Xiao, Qingyun Li, Yicong Li, Xun Yang, Dan Guo, Meng Wang, Tat-Seng Chua, and Angela Yao. Egotextvqa: Towards egocentric scene-text aware video question answering. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3363–3373, 2025.

Appendix

Appendix Contents

A Detailed Contributions	32
B Qualitative Examples	33
B.1 Examples for Embodied Cognition	33
B.2 Examples for Embodied Location	33
B.3 Examples for General Visual Understanding	33
B.4 Examples for Physically Grounded Reasoning	33
B.5 Examples for Navigation	33
B.6 Examples for Manipulation Planning	33
C Prompts Details	45
C.1 Training QA Prompts	45
C.2 Evaluation and Inference Prompts	45
C.3 Hyper-parameters for Evaluation	48

A Detailed Contributions

- **Data:** Ronghao Dang, Zhikai Wang, Yunxuan Mao, Yuqian Yuan, Bohan Hou, Jiangpin Liu, Kehan Li, Jiayan Guo, Xin Li, Sicong Leng, Minghao Zhu, Yang Bai, Qian Jiang
- **Model Training:**
 - **Pre-training:** Jiayan Guo, Kehan Li, Ronghao Dang, Sicong Leng, Zhikai Wang, Xin Li, Jiangpin Liu, Yuqian Yuan, Yunxuan Mao
 - **Post-Training:**
 - **Chain-of-Point Reasoning:** Jiayan Guo, Zhikai Wang, Minghao Zhu, Yuqian Yuan, Ronghao Dang, Kehan Li
 - **Vision-Language Navigation:** Jiangpin Liu, Yuqian Yuan, Kehan Li, Ronghao Dang, Liuyi Wang
 - **Planning:** Yunxuan Mao, Bohan Hou, Kehan Li, Ronghao Dang, Yuqian Yuan, Xiao Lin
 - **VLA:** Kehan Li, Bohan Hou, Yunxuan Mao, Yuqian Yuan, Jiayan Guo, Ronghao Dang
- **Infrastructure:** Kehan Li, Jiayan Guo, Xin Li
- **Evaluation:** Minghao Zhu, Bohan Hou, Yuqian Yuan, Sicong Leng, Jiayan Guo, Kehan Li, Yunxuan Mao, Jiangpin Liu, Ronghao Dang, Yuming Jiang, Xin Li
- **Hardware and Robot System:** Yaxi Zhao, Minghua Zeng, Junlong Gao
- **Senior Advisory:** Jun Cen, Siteng Huang, Wenqiao Zhang (Zhejiang University), Chengju Liu (Tongji University), Jianfei Yang (Nanyang Technological University), Shijian Lu (Nanyang Technological University), Deli Zhao

B Qualitative Examples

This section presents an extensive set of visual examples to demonstrate RynnBrain’s robust capabilities across a wide range of embodied tasks.

B.1 Examples for Embodied Cognition

As shown in [Figure 9](#) and [Figure 10](#), our RynnBrain supports a wide range of embodied cognition abilities, including estimating object size and distance, reasoning about relative directions and object counts, performing fine-grained grounded object-centric understanding and OCR-based perception, and conducting higher-level spatial reasoning over egocentric views and 3D shapes.

B.2 Examples for Embodied Location

As illustrated in [Figure 11](#) and [Figure 12](#), our RynnBrain demonstrates robust embodied location understanding by accurately interpreting spatial references in egocentric views and grounding natural language instructions to precise physical coordinates. It can localize objects based on relative positions, identify graspable items via functional cues, and generate complex spatial trajectories such as moving or cleaning specified regions.

B.3 Examples for General Visual Understanding

As illustrated in [Figure 13](#), RynnBrain also demonstrates strong general visual understanding abilities, including video comprehension and image-level understanding of structured content such as charts and documents.

B.4 Examples for physically Grounded Reasoning

As illustrated in [Figure 14](#), RynnBrain-CoP demonstrates physically grounded reasoning capabilities.

B.5 Examples for Navigation

As illustrated in [Figure 15](#), [Figure 16](#) and [Figure 17](#), RynnBrain-Plan demonstrates robust long-horizon navigation planning capabilities.

B.6 Examples for Manipulation Planning

As illustrated in [Figure 18](#), [Figure 19](#) and [Figure 20](#), RynnBrain-Plan demonstrates robust long-horizon planning capabilities. Furthermore, its precise grounding ability enables our method to handle a wide range of fine-grained manipulation tasks adeptly. The details of each task are as follows:

Distribute Tableware. In this task, the planning model is required to distribute tableware for a specified number of people. The detailed task prompt is provided below.

- Easy: Distribute the tableware on the table among three people.
- Medium: Distribute the tableware on the table among three people, making sure that the cups are on the right side of each person.
- Hard: Distribute the tableware on the table among four people, making sure that the cups are on the right side of each person.

Object Classification. In this task, the planning model is required to categorize the food items on the table and arrange them in a row according to specific instructions. The detailed task prompt is provided below.

- Easy: Sort the fruits on the table and arrange each type of fruit in a row.
- Medium: Sort the fruits on the table into categories and arrange each category in a row, placing them on either side of the mug.
- Hard: Sort the fruits on the table into categories and arrange each category in a row, placing them on either side of the mug. Don’t move the fruits in the bowl.



Question: What is the length of the **longest dimension** (length, width, or height) of the door, measured in centimeters?
Answer: 192.

Object size

Question: If I am **standing by the window and facing the keyboard**, is the telephone to my front-left, front-right, back-left, or back-right?
Answer: front-right.

Relative direction

Question: How many trash bin(s) are in this room?

Answer: 2.

Object counting

Question: Measuring from the closest point of each object, what is the **distance between the door and the telephone** (in meters)?
Answer: 3.5.

Object distance



Question: Where did I put the steel container?
Answer: On the table.

Ego-centric Understanding

Question: Where did I take the steel container?
Answer: In the cabinet below.

Ego-centric Understanding



Question: What **brand of sponge** should I use to clean up?
Answer: vileda.

OCR

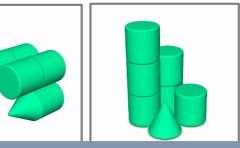
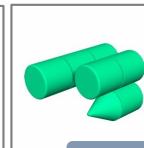
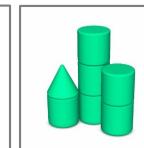
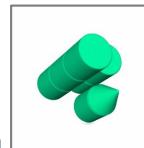
Question: Where do I place the **yellow vileda sponge**?
Answer: on the sink.

OCR



Spatial Reasoning

Question: Based on these two views showing the same scene: **in which direction did I move** from the first view to the second view?
Answer: Diagonally forward and right.



Spatial Reasoning

Question: Which of these images are different perspectives **of the same object**? A. First and second. B. First and third. C. All are the same. D. All are different.
Answer: B. First and third.

Figure 9 Embodied Cognition Examples of RynnBrain. RynnBrain supports diverse embodied cognition tasks, including spatial understanding of object size, direction, distance, and counting, OCR-based perception, and higher-level spatial reasoning over egocentric views and 3D shapes.



Question: What degree of clockwise rotation is necessary for you to face <frame 82>; (359,157),(596,361) </object>?

Answer: About 10 degrees.

Camera rotation

Question: What o' clock is <frame 5>; (459,157),(696,661) </object> located with respect to your position?

Answer: 10 o'clock position.

Direction to camera

Question: What is the function of <frame 2>; (910,480),(979,596) </object>?

Answer: The object is used to control the computer.

Object function

Question: What is <frame 5>; (459,157),(696,661) </object> made of?

Answer: The object is made of wood and fabric.

Object material



Question: What is the distance between <frame0>; (39, 310), (232, 414) </object> and <frame2>; (463, 207), (849, 1000) </object>?

Answer: 1.02 meters.

Distance estimation



Question: There are 3 objects in the video: <object> <frame0>; (855, 657), (925, 837) </object>, <object> <frame1>; (619, 710), (669, 781) </object>, <object> <frame4>; (190, 896), (284, 977) </object>. Which of the three objects <object0>, <object1>, and <object2> is tallest?

Answer: <object2> is the tallest.

Height comparison



Question: There are 2 objects in the video: <object> <frame 0>; (786,425),(904,559) </object>, <object> <frame1>; (17,614),(380,974) </object>. Does <object0> have a larger size compared to <object1>?

Answer: Actually, <object0> might be smaller than <object1>.

Size comparison

Figure 10 Embodied Cognition Examples of RynnBrain. RynnBrain also supports a diverse range of fine-grained embodied cognition tasks, including spatial understanding of camera rotation, direction, distance, and size, as well as object understanding of function and material.

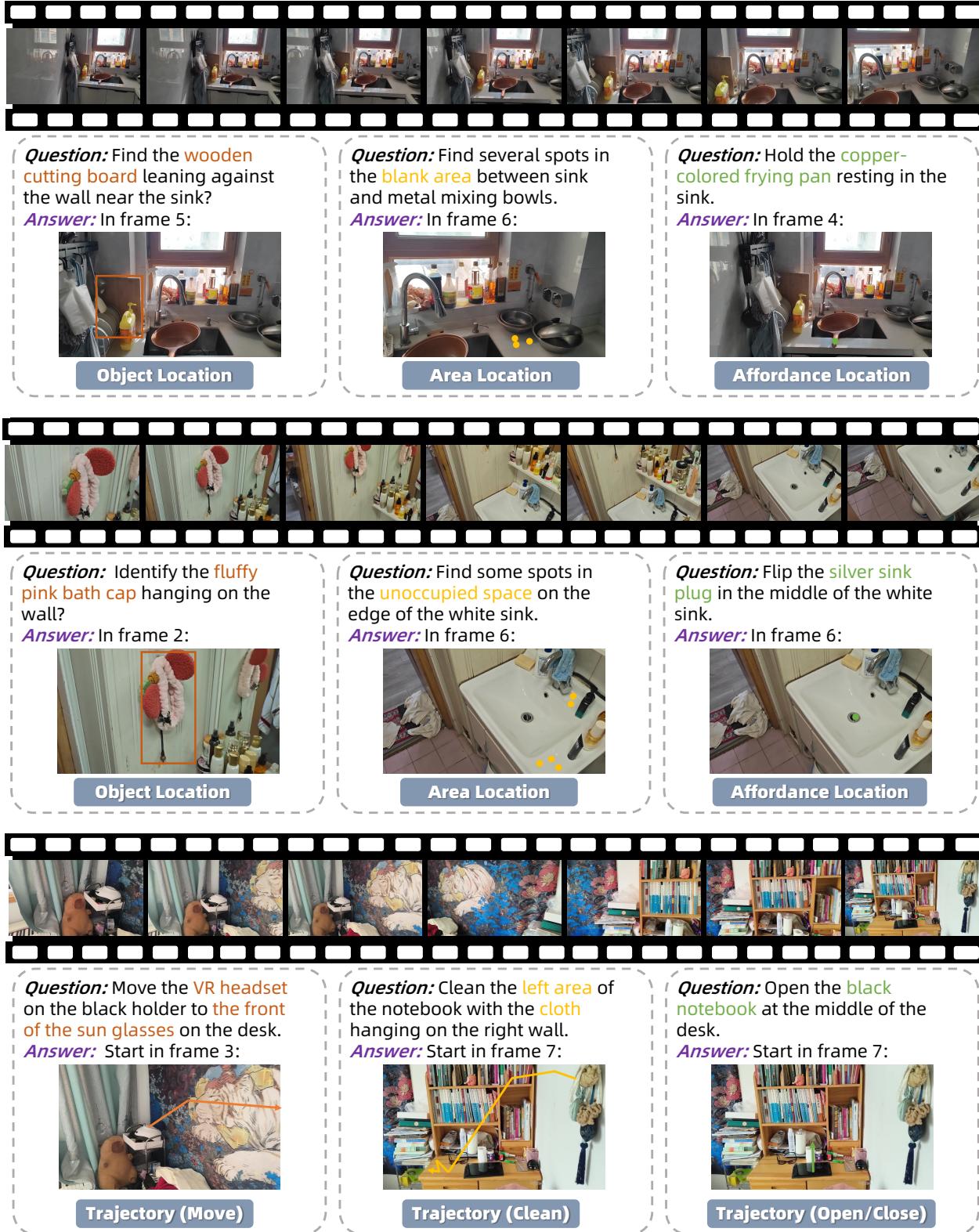


Figure 11 Embodied Location Video Examples of RynnBrain. RynnBrain excels at grounded spatial reasoning, supporting video-based location tasks for object, area, affordance, trajectory. These examples highlight its ability to map linguistic descriptions to 3D locations and actions in real-world scenes.

Question 1: Identify the grounding box for the **blue plastic water bottle** on the ground behind the door.



Object Location

Question 1: Identify **several spots on the ground** between the two sofas.



Area Location

Question 1: Click down the **C key** on the white electric keyboard.



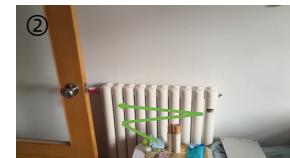
Affordance Location

Question 1: Move the **black metallic kettle** on the left-side **black heat unit**.



Trajectory (Move)

Question 1: Wipe the **black plastic cabinet door** on the left side of the table with the **green cloth** located on the table's right side.



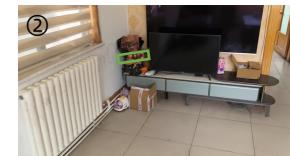
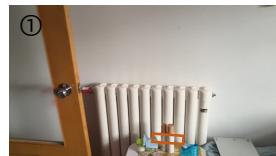
Trajectory (Clean)

Question 1: Open the second **yellow cabinet door** from the right.



Trajectory (Open/Close)

Question 1: Identify the grasping coordinates for the two **white skincare bottles**.



Grasp Pose

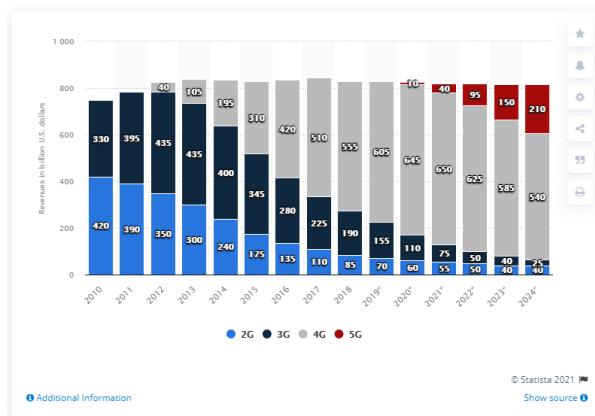
Figure 12 Embodied Location Image Examples of RynnBrain. RynnBrain excels at grounded spatial reasoning, supporting image-based location tasks for object, area, affordance, trajectory, and grasp pose. These examples highlight its ability to map linguistic descriptions to 3D locations and actions in real-world scenes.



Question: Describe the video in detail.

Video Understanding

Answer: The video takes place in a **forested area**, likely during a camping trip or outdoor event. The scene is set with **tall trees** and a **ground covered in dirt and patches of grass**. Colorful triangular flags are strung between the **trees**, adding a festive atmosphere to the setting. A pink tent is visible in the background, indicating a **campsite**. The main focus of the video is on two individuals, an **older person** and a **younger boy**, who are engaged in a playful game of **hide-and-seek**. The older person is wearing a **gray puffer vest** over a **plaid shirt** and **gray pants**, while the **younger boy** is dressed in a **yellow striped shirt** and **denim shorts**. They are seen **peeking around a large tree**, with the older person trying to hide behind it and the boy attempting to find them. The interaction between the two is light-hearted and fun, capturing a moment of joy and connection in a natural setting.



Question: What was the global 4G service revenues in dollars in 2018?

Answer: 555.

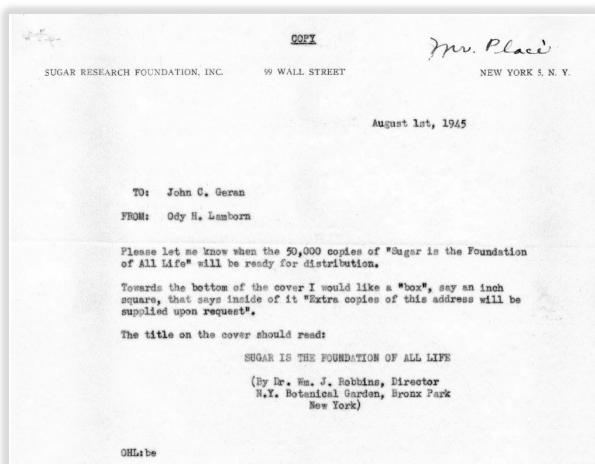
Question: What new generation of mobile technology is forecast to gain market revenue in the coming years?

Answer: 5G.

Question: What year is the first represented on the grid?

Answer: 2010.

ChartQA



Question: Please parse all the text in the document.

Answer: COPY 99 WALL STREET NEW YORK 5, N. Y.

August 1st, 1945

TO: John C. Geran FROM: Ody H. Lamborn
 Please let me know when the 50,000 copies of "Sugar is the Foundation of All Life" will be ready for distribution.

Towards the bottom of the cover I would like a "box", say an inch square, that says inside of it "Extra copies of this address will be supplied upon request".

The title on the cover should read:

SUGAR IS THE FOUNDATION OF ALL LIFE
 (By Dr. Wm. J. Robbins, Director N.Y. Botanical Garden, Bronx Park New York)

OHL:be

DocVQA

Figure 13 General Visual Understanding Examples of RynnBrain. RynnBrain also preserves strong general visual understanding capabilities, including video comprehension and image understanding for content such as charts and documents.



Question: Lift the patterned pillow on the far left of the sofa.

Thinking: I first locate the main object, the [sofa] `<area> <frame 7> (940, 939), (559, 612), (806, 808), (530, 681) </area>`, which is the large piece of furniture in the center of the image. Next, I scan the [sofa] `<area> <frame 7> (940, 939), (559, 612), (806, 808), (530, 681) </area>` to find the target object, a 'patterned pillow'. I identify two [patterned pillows] `<object> <frame 7> ((213, 185), (480, 489)) </object>` with cartoon faces on the [sofa] `<area> <frame 7> (940, 939), (559, 612), (806, 808), (530, 681) </area>`. The instruction specifies the one 'on the far left', so I select the [pillow on the far left] `<object> <frame 7> ((213, 185), (480, 489)) </object>`. The required action is 'lift'. To lift a soft object like a pillow, the most effective way is to grasp its main body. Therefore, the final affordance prediction is placed on the surface of the [patterned pillow on the far left] `<object> <frame 7> ((213, 185), (480, 489)) </object>`.

Answer: `<affordance> <frame 7>, (347, 369) </affordance>`

Affordance Prediction with Thinking



Question: Move the black chopsticks from the container on the left to the metal rack on the right.

Thinking: First, I must parse the user's instruction. The command is to 'move' the 'black chopsticks' from the 'container on the left' to the 'metal rack on the right'. This requires picking up the specified chopsticks and placing them in the rack. The starting point of the trajectory is the object being moved, which are the [black chopsticks] `<object> <frame 2> ((81, 352), (206, 470)) </object>` located inside the [container on the left] `<area> <frame 2> (129, 491), (154, 550), (154, 516) </area>`. The destination is the target location, which is the [metal rack on the right] `<area> <frame 5> (834, 695), (934, 733), (736, 572) </area>`. Therefore, the trajectory originates from the [black chopsticks] `<object> <frame 2> ((81, 352), (206, 470)) </object>`, moves upwards and to the right, and terminates on the [metal rack on the right] `<area> <frame 5> (834, 695), (934, 733), (736, 572) </area>`.

Answer: `<trajectory> <frame 2>, (258, 404), (306, 354), (413, 406), (501, 465) </trajectory>`

Trajectory Prediction with Thinking

Figure 14 Embodied Location Video Examples of RynnBrain. RynnBrain excels at grounded spatial reasoning, supporting video-based location tasks for object, area, affordance, trajectory. These examples highlight its ability to map linguistic descriptions to 3D locations and actions in real-world scenes.

Desk Organization. In this task, the planning model is required to place different types of pens and garbage into designated locations according to specific requirements. The detailed task prompt is provided below.

- Easy: Tidy up the desktop. Put the thin pens in the pen holder and arrange the thick pens from left to right in the order of red, and black. Finally, make sure there is no trash on the desktop.
- Medium: Tidy up the desktop. Put the thin pens in the pen holder and arrange the thick pens from left to right in the order of red, black and blue. Finally, make sure there is no trash or used paper cups on the table.
- Hard: Tidy up the desktop. Put the thin pens in the pen holder and arrange the thick pens from left to right in the order of red, black and blue. Finally, make sure there is no trash or used paper cups on the table.

Table Bussing (OOD). In this task, the planning model is required to generate a detailed action plan in response to the instruction “Bus the table”. The detailed task setting is provided below.

- Easy: Two forks, two pens, one plate, and one pen holder.
- Medium: Two forks, two pens, two plates, two cups and one pen holder.
- Hard: Two forks, two pens, two plates, two cups, one trash, one trash can and one pen holder.

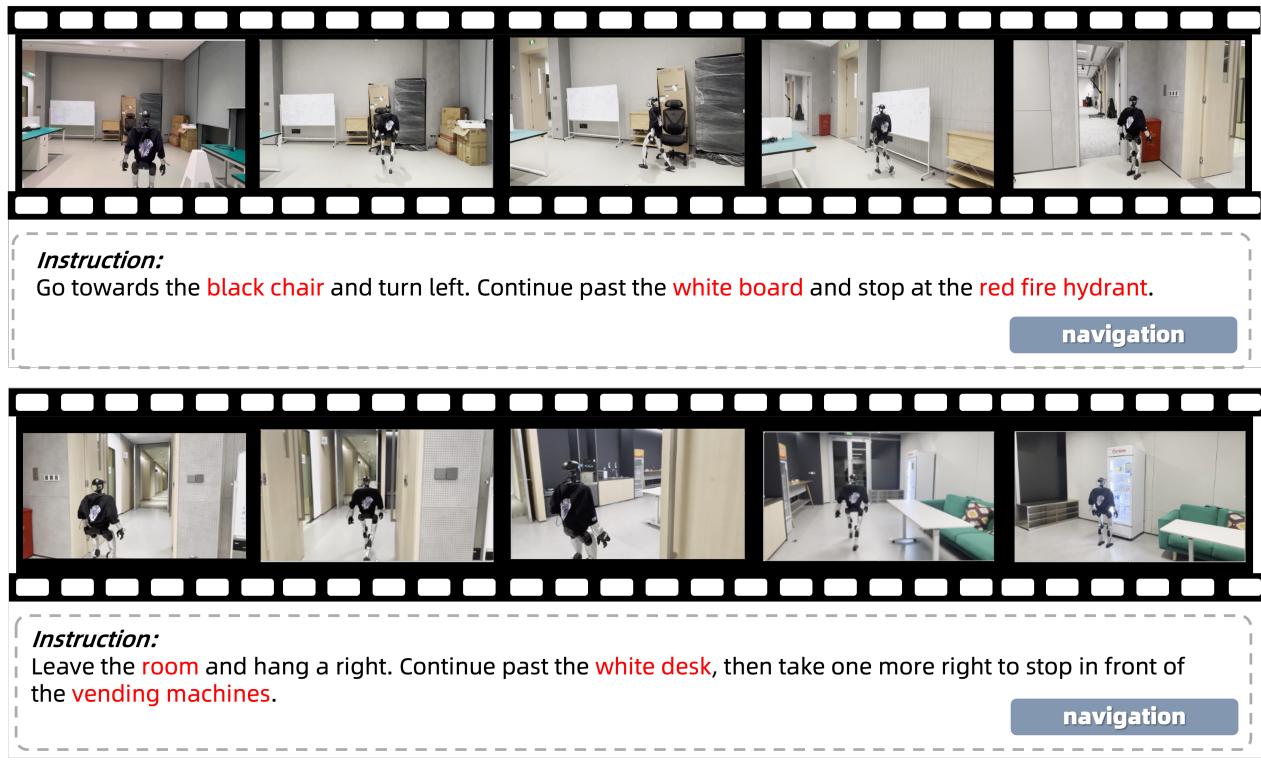
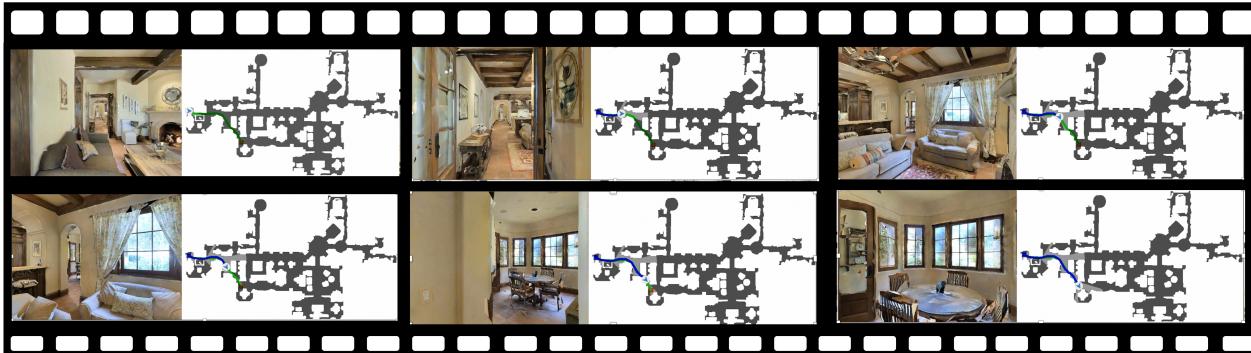


Figure 15 Visual Language Navigation Video Examples of RynnBrain in Real Environment. Examples of the RynnBrain-Nav model in a real indoor environment. The results demonstrate the strong navigation ability of the model in the real environment.



Instruction:

Go straight past the **couches** and the **fireplace** on the right. Go through the **double doors**. Turn right and pass the **couches**. Go into the **room** on the right. Wait near the **round table** and **chairs** with the white flower in the **table**.

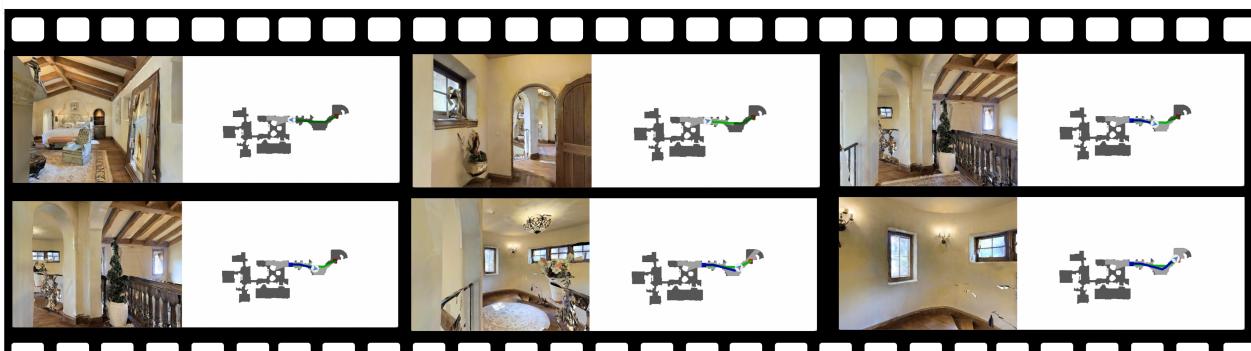
navigation



Instruction:

walk towards **fireplace**, walk on the left side of the **brown couch**, turn right after **couch**, turn left after **four person table** and stop in front of **barstools**.

navigation

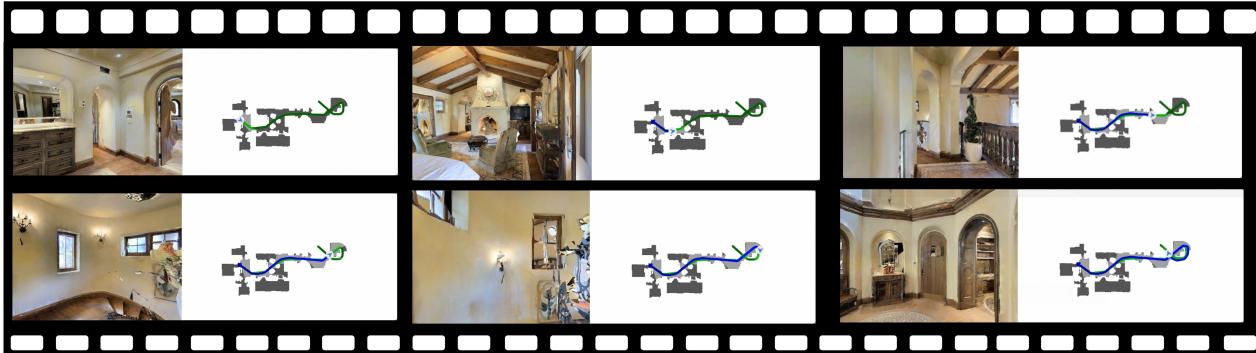


Instruction:

Walk through the **door** and over the **overpass**. Wait on the **white circular rug**.

navigation

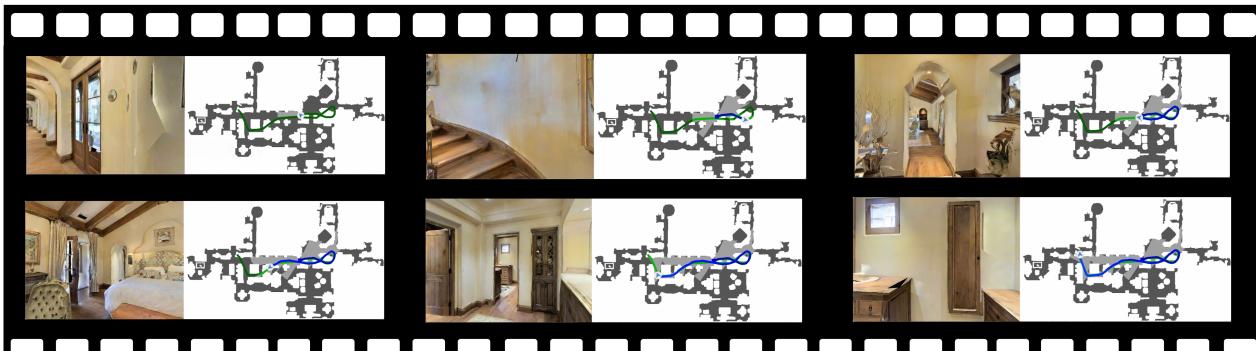
Figure 16 Visual Language Navigation Video Examples of RynnBrain on R2R-CE. Examples of the RynnBrain-Nav model on R2R-CE. The results demonstrate the strong navigation ability of the model.



Instruction:

You are inside a house. Face towards the **table** which has **12 drawers** on it and above it a **mirror**. When you are facing towards that **cabinet**, stay to the right of it and walk under the **archway and down the hallway**. Once you walk through that **small archway**, you will enter a **bedroom** where there is a large **fireplace** straight ahead, the **bed** is on your left, and two open **french doors** on your right. Keep walking straight ahead towards the **two chairs and small coffee table**, where there is a **fireplace** behind them. Walk in between the **left chair and the coffee table** and keep walking straight, past the left side of the **fireplace**. Continue walking down the hall through the **open door**. Walk onto the **rug**, which is centered between two large potted plants. Continue past the rug, under the **archway**. You are now in a **landing area**, just above a staircase, where there is a large colorful pot of flowers on your right and a circular mirror on your left. Walk all the way **down the winding staircase**. Once you've reached the **bottom of the staircase**, walk into the middle of the foyer, on to the to the part of the floor with a circular design. Once you're standing atop that **circular design on the floor**, you've reached your final destination.

[navigation](#)



Instruction:

Now you are facing to the **wall**. Turn slightly right and move forward, on your right there is a **stair case** take the stairs case and climb up and turn left, pass through the railing till you reach **red vessel**, on your right their is an **open entrance** move forward, if you look forward there is a **couch**, move towards the **couch** and turn slightly left, you can see a **open door** move towards the open door, now you are facing the **bed** in your right and the **open door** on your left, if you look forward there is an **open entrance** move towards it and turn slightly right, there is a **open door** enter into the room which you can see a **white mat** and stand in front of a **small cupboard** and stop.

[navigation](#)

Figure 17 Visual Language Navigation Video Examples of RynnBrain on RxR-CE. Examples of the RynnBrain-Nav model on RxR-CE. The results demonstrate the strong navigation ability of the model.

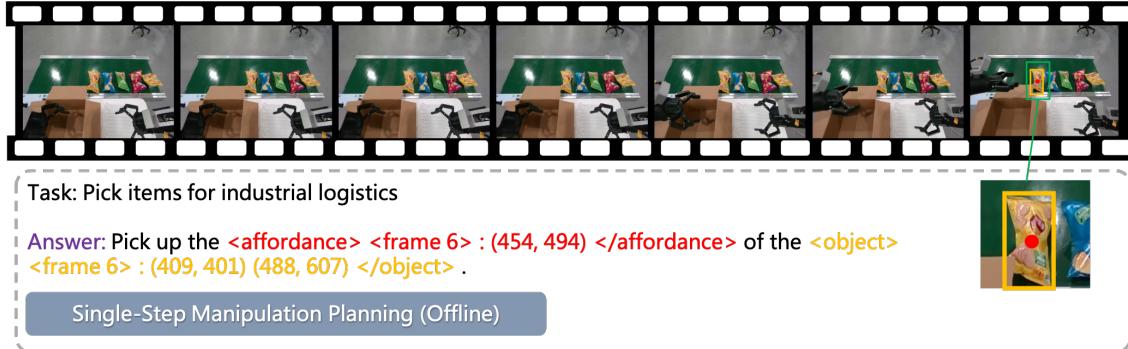


Figure 18 Planning Manipulation Video Examples of RynnBrain-Plan. An example of the RynnBrain-Plan model on a one-step offline planning task on the Agibot Dataset.

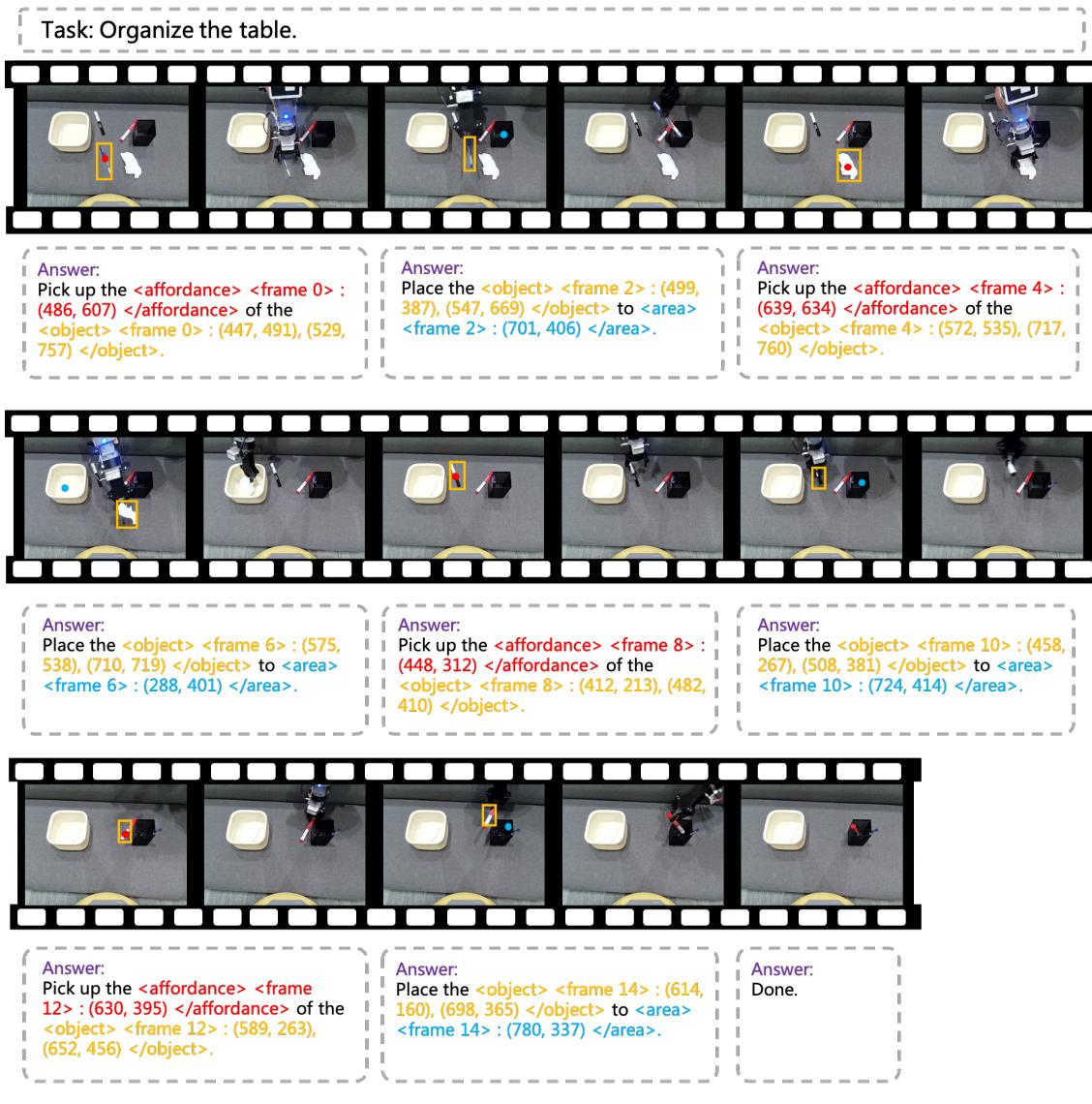
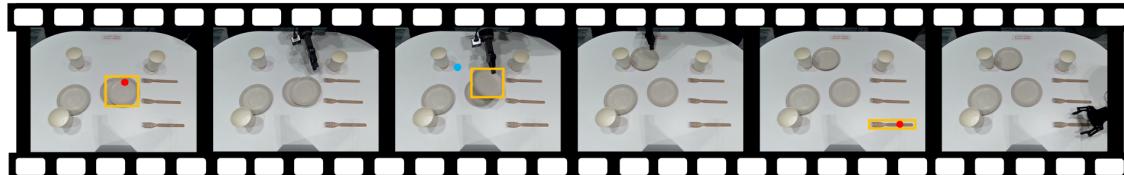


Figure 19 Planning Manipulation Video Examples of RynnBrain-Plan. An example of the RynnBrain-Plan model on a multi-step online planning task. The executor is a human expert with a UMI.

Task: Distribute the tableware for three people.



Answer:
Pick up the <affordance> <frame 0> : (578, 446) </affordance> of the <object> <frame 0> : (483, 394), (652, 612) </object>.

Answer:
Place the <object> <frame 2> : (468, 318), (650, 567) </object> to <area> <frame 2> : (380, 165) </area>.

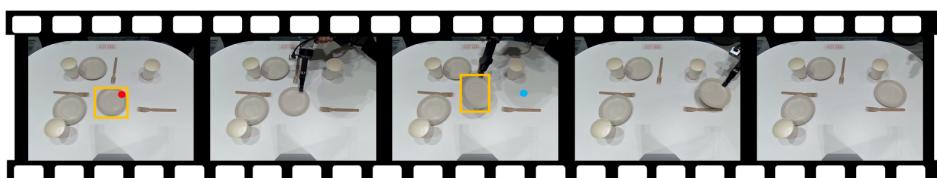
Answer:
Pick up the <affordance> <frame 4> : (891, 787) </affordance> of the <object> <frame 4> : (666, 752), (928, 799) </object>.



Answer:
Place the <object> <frame 6> : (701, 717), (847, 783) </object> to <area> <frame 6> : (268, 443) </area>.

Answer:
Pick up the <affordance> <frame 8> : (792, 430) </affordance> of the <object> <frame 8> : (661, 394), (886, 456) </object>.

Answer:
Place the <object> <frame 10> : (684, 291), (934, 338) </object> to <area> <frame 10> : (491, 255) </area>.



Answer:
Pick up the <affordance> <frame 12> : (540, 453) </affordance> of the <object> <frame 12> : (412, 406), (605, 623) </object>.

Answer:
Place the <object> <frame 14> : (422, 279), (615, 569) </object> to <area> <frame 14> : (793, 456) </area>.

Answer:
Done.

Multi-Step Manipulation Planning (Online-UMI)

Figure 20 Planning Manipulation Video Examples of RynnBrain-Plan. An example of the RynnBrain-Plan model on a multi-step online planning task. The executor is a human expert with a UMI.

C Prompts Details

C.1 Training QA Prompts

To ensure reproducibility, we list the detailed prompt templates used for each training task in [Table 9](#). All prompts are presented in a unified Python f-string format.

C.2 Evaluation and Inference Prompts

To ensure reproducibility, we list the detailed prompt templates used for each benchmark in [Table 10](#). All prompts are presented in a unified Python f-string format.

Table 9 Prompt Templates for Various Training Tasks

Training Task	Prompt Template (Unified Format)
Object Understanding	f"I'd like to know about the area labeled <object> ({x[0]}, {y[0]}), ({x[1]}, {y[1]}) </object> in the image. Can you give a short description?"
Spatial Understanding	f"You are in the last frame of the video. There are {n} objects in the video: <object{idx}> <object> <frame {frame_id}>: ({x[0]}, {y[0]}), ({x[1]}, {y[1]}) </object>, ... \n{question}"
OCR	f"Locate the text "{text}" in this video.\n\nStep 1: Predict the key frame.\nStep 2: Output a tuple series.\nOutput format: <area> <frame n>: (x1, y1), (x2, y2), </area>\nwith all coordinates normalized to 0-1000 range."
Object Location	<ol style="list-style-type: none"> 1. f"{question}\nOutput the bounding box in the format <object> <frame n>: ...; (x1, y1), (x2, y2) </object>. n is the chosen frame index." 2. f"{question}\nGenerate coordinates for one object bounding box. Constraints: x1, y1, x2, y2 ∈ [0, 1000]. Response must be in the format: <object> (x1, y1), (x2, y2) </object>"
Area Location	<ol style="list-style-type: none"> 1. f"{question}\nFirst perform key frame prediction, then generate a sequence of coordinate tuples.\nOutput format: <area> <frame n>: ...; (x1, y1), (x2, y2), </area>\nEach coordinate pair must contain normalized pixel values within the [0, 1000] range." 2. f"{question}\nExpress the coordinates as a tuple sequence in the format <area> (x1, y1), (x2, y2), ... </area> with all coordinate values normalized to the standardized pixel coordinate system spanning 0 to 1000."
Affordance Location	<ol style="list-style-type: none"> 1. f"{question}\n1. First identify the key frame\n2. Then predict one affordance point.\nOutput format: <affordance> <frame n>: ...; (x, y) </affordance>\nCoordinates normalized to 0-1000 pixel space." 2. f"{question}\nTask: Affordance point prediction\n- Identify one possible affordance point\n- Normalize coordinates to 0-1000 range\n- Output format: <affordance> (x, y) </affordance>\n- Example: [450, 320]"
Trajectory Location	<ol style="list-style-type: none"> 1. f"{question}\nor trajectory completion:\n1. First locate the frame with the trajectory start point\n2. Then predict up to 10 key points as list.\nOutput format: <trajectory> <frame n>: ...; (x1, y1), (x2, y2), </trajectory>\nAll coordinates normalized to 0-1000 pixel space." 2. f"{question}\nPredict a trajectory comprising up to 10 key points. Return coordinates in the format <trajectory> (x1, y1), (x2, y2), ... </trajectory> with all values normalized to the [0, 1000] range."

Continued on next page...

Table 9 – continued from previous page

Training Task	Prompt Template (Continued)
Navigation	f"You are an autonomous navigation assistant. Your task is to Walk straight, and when you reach the end of the table turn right. Wait by the brass chairs. Devise an action sequence to follow the instruction using the four actions: TURN LEFT, TURN RIGHT, MOVE FORWARD, or STOP."
Planning	f"You are a sophisticated dual-arm robot planning the next action for the goal: {question}. Put the thin pens in the pen holder and arrange the thick pens from left to right in the order of red, black and blue. Finally, make sure there is no trash on the desktop..\n\n Adhere to the following output rules:\n - Rule 1: The response must be a single, complete sentence.\n - Rule 2: The sentence must embed data by selecting a frame 'n' and predicting integer coordinates within the [0, 1000] range.\n - Rule 3: Data format within tags must be '<tag> <frame n>: (data) </tag>', where 'data' is a single point for both affordance and area, and two points '(min_coord), (max_coord)' for object."

Table 10 Prompt Templates for Various Benchmarks

Benchmark	Prompt Template (Unified Format)
VSL-Bench	1. f"{question}\nAnswer with the option's letter from the given choices directly." 2. f"{question}\n\nAnswer the question with an exact number, which should be accurate to at most two decimal places."
MMSI	f"{question}"
ERQA	f"{question}\nAnswer with the option letter from the given choices directly."
RoboSpatial	1. f"{question}. Pinpoint several points within the vacant space situated to the left of the vacuum. Your answer should be formatted as a list of tuples, i.e. [(x1, y1), ...], where each tuple contains the x and y coordinates... indicating the normalized pixel locations of the points." 2. f"{question} Answer yes or no."
EgoTaskQA	f"Select the best answer to the following multiple-choice question based on the video.\n{question}\nOptions:\n(A) {options[0]}\n(B) {options[1]}\n(C) {options[2]}\n(D) {options[3]}\n(E) {options[4]}\nAnswer with the option's letter from the given choices directly and only give the best option. The best answer is: "
EgoTextVQA_indoor	f"You are a person in the situation shown in the following consecutive images... Answer the question as detailed as possible, covering all relevant aspects and providing comprehensive context.\n\nQuestion: {question}"
Open-X VQA	f"Select the best answer to the following multiple-choice question based on the image.\n{question}\nOptions:\n{opts_text}\nAnswer with the option's letter from the given choices directly and only give the best option. The best answer is: "
QAEgo4D	f"You are a helpful assistant. Please evaluate the predicted answer based on the given question. A score of 0 means the answer is completely incorrect... Output only a single score from the following set: [0, 1, 2, 3, 4, 5]. {question}"

Continued on next page...

Table 10 – continued from previous page

Benchmark	Prompt Template (Continued)
MindCube	f"Select the best answer to the following multiple-choice question based on the image.\n{question}\nAnswer with the option's letter from the given choices directly and only give the best option. The best answer is: "
RynnBrain-Object	f"{question} Your current position is at the last frame of the video."
RynnBrain-Spatial	f"{question}"
RefSpatial-Bench	f"Locate {object_name} in this image. Output the point coordinates in JSON format."
ShareRobot-Affordance	f"{question}The coordinates should be between 0 and 1000, indicating the normalized pixel locations of the point."
ShareRobot-Trajectory	f"{question}Your answer should be formatted as a list of tuples, i.e. [(x1, y1), (x2, y2), ...], where each tuple contains the x and y coordinates of a point. The coordinates should be between 0 and 1000, indicating the normalized pixel locations of the point."
Cornell-Grasp VMRD-Grasp	f"How should the robot grasp the object? Output the grasping pose as 4 corner points of the gripper rectangle.\n- Format: <grasp pose> (x1, y1), (x2, y2), (x3, y3), (x4, y4) </grasp pose>\n- All coordinates in range [0, 1000] (normalized)\n- The 4 corners define the gripper's position, orientation, and width"
RynnBrain-Grounding	f"{question}. Output the bounding box in the format <object> <frame n>: ...; (x1,y1), (x2,y2) </object>. n is the chosen frame index."
RynnBrain-Area	f"{question}. First predict the key frame, then output coordinates as a series of tuples. \nOutput format: <area> <frame n>: ...; (x1, y1), (x2, y2), </area>\n All coordinates must be normalized between 0 and 1000."
RynnBrain-Affordance	f"{question}. First predict the key frame, then output a single affordance point as coordinates (x, y).\nOutput format: <affordance> <frame n>: ...; (x, y) </affordance>\n Both x and y values must be normalized between 0 and 1000."
RynnBrain-Trajectory	f"{question}. First predict the frame containing the trajectory start point, then output up to 10 key trajectory points as a list of tuples in the format: <trajectory> <frame n>: ...; (x1, y1), (x2, y2), </trajectory>\n All coordinates must be normalized between 0 and 1000."
AI2D	f"{question}"
ChartQA	f"{question}\nAnswer the question using a single word or phrase."
DocVQA	f"{question}\nAnswer the question with a single word or phrase."
MVBench	f"Question: {question}\nOptions: \n{option_string}Answer with the option's letter from the given choices directly and only give the best option."
RealworldQA	f"{question}"
InfoVQA_test	f"{question}\nAnswer the question with a single word or phrase."

Continued on next page...

Table 10 – continued from previous page

Benchmark	Prompt Template (Continued)
EgoSchema	f"Select the best answer to the following multiple-choice question based on the video.\n{question}\nOptions:\n(A) {options[0]}\n(B) {options[1]}\n(C) {options[2]}\n(D) {options[3]}\n(E) {options[4]}\nAnswer with the option's letter from the given choices directly and only give the best option. The best answer is: "
VideoMME w/o sub	f"Select the best answer to the following multiple-choice question based on the video. Respond with only the letter (A, B, C, or D) of the correct option.\n {question} "

C.3 Hyper-parameters for Evaluation

To ensure reproducibility, we disabled sampling during autoregressive text generation unless otherwise specified. For images processing, we constrained the resolution by setting min_pixels to $16 \times 32 \times 32$ and max_pixels to $16384 \times 32 \times 32$. For video-related benchmarks, frames are sampled at 2 FPS; if the total exceeds 512 frames, we apply uniform sampling to maintain a maximum of 512. The min_pixels for each frame and max_pixels for the whole video are set to $16 \times 32 \times 32$ and $24576 \times 32 \times 32$, respectively. On certain pointing-related benchmarks (e.g., ERQA, RoboSpatial, and ShareRobot), we observed that sampling can further enhance model performance. In these instances, we set the temperature to 0.2, top_p to 0.95, and top_k to 50. Additionally, for RefSpatial and ShareRobot—which involve numerous precise positioning tasks—we employed a higher resolution by increasing min_pixels to $1024 \times 32 \times 32$.