RxBrain: Embodied Foundation Model with Joint Language–Visual Subgoal Planning

Anonymous authors
Paper under double-blind review

Abstract

We introduce **RxBrain**, an embodied foundation model that combines embodied understanding and reasoning with joint language–visual subgoal planning. It interleaves textual subgoals with goal images, pairing task decomposition and constraints with intended intermediate and final physical states. Built on vision-language model, RxBrain adopts a Mixture-of-Transformers architecture, with general image generation pretraining followed by interleaved text–image planning training. We introduce an inference-aware visual state construction strategy to enhance joint planning. Our automatic pipeline converts over $50{,}000$ hours of embodied video into 21.51M trainable segments, aligning textual steps with visual state transitions. Experiments demonstrate competitive performance on embodied understanding and reasoning benchmarks, alongside general image generation and joint language–visual subgoal planning capabilities. Its joint subgoals substantially improve downstream policy performance on long-horizon manipulation tasks, while an action-generation model built on RxBrain enables effective continuous action generation on real robots without additional action pretraining.

1 Introduction

Vision-language models (VLMs) provide a general foundation for embodied intelligence, supporting visual perception, spatial grounding, task understanding, and high-level reasoning (Ji et al., 2025; Team, 2025; Dang et al., 2025; 2026; Lee et al., 2025; Fang et al., 2026b). However, text-only plans often leave the intended physical outcomes underspecified, describing what to do without fully specifying the resulting object configurations or spatial relations. Visual subgoals offer a complementary representation by making these outcomes explicit. Extending embodied VLMs with this capability offers a path from understanding the current scene to planning how it should change.

Recent unified multimodal models have made progress toward combining understanding and generation (Deng et al., 2025; Agarwal et al., 2026). In embodied settings, the $\pi_{0.7}$ framework (Intelligence et al., 2026) uses a VLM (Team et al., 2025b) to generate subtask instructions and a separate image generation model (Deng et al., 2025) to generate visual subgoals, providing both as context for the downstream action policy. These advances motivate studying language and vision not merely as alternative output modalities, but as complementary components of an embodied plan. The central question is how to integrate joint planning into a VLM while preserving its broad understanding and reasoning capabilities, and how to extend this planning capability into downstream robot execution.

We introduce RxBrain, an embodied foundation model that combines embodied understanding and reasoning with joint language–visual planning. Building on the HY-Embodied-0.5 VLM (Team et al., 2026), we develop a unified multimodal model with a Mixture-of-Transformers architecture that supports both text and image generation. We first pretrain it for general-purpose text-to-image generation, then train it for joint language–visual planning through interleaved text–image generation. Throughout both stages, we co-train on embodied understanding and reasoning tasks. To further enhance joint planning, we introduce an inference-aware visual state construction strategy.

To train these capabilities, we develop an automatic data construction pipeline that converts over $50{,}000$ hours of embodied video into 21.51 M trainable segments for joint language–visual planning. Through temporal segment annotation, quality verification, and segment structuring, the pipeline aligns textual planning steps with action-induced visual state transitions, providing supervision at

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

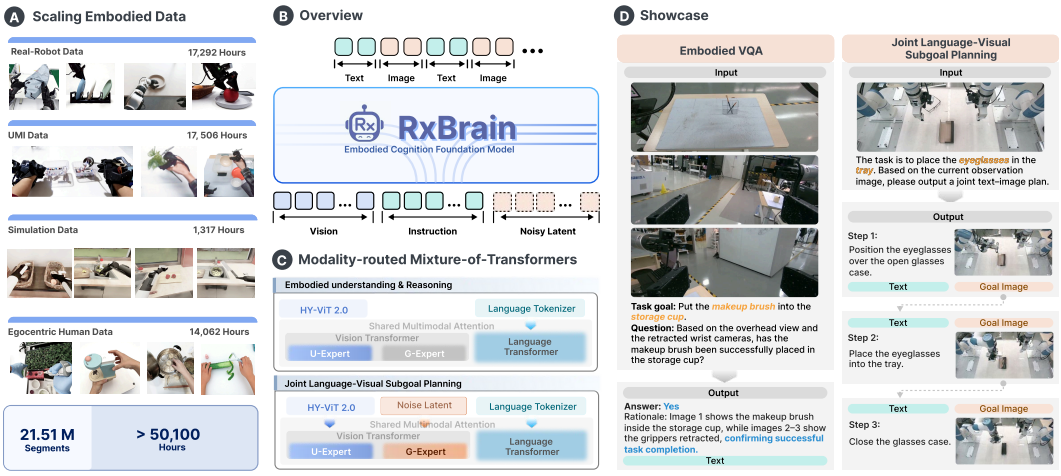


Figure 1: **Overview of RxBrain.** (A) Over 50,000 hours of embodied video spanning real-robot, UMI, simulation, and egocentric human data. (B–C) A modality-routed Mixture-of-Transformers unifies embodied understanding and reasoning with interleaved text–image generation. (D) Examples of embodied question answering and joint subgoal planning.

both step and task levels. We also introduce RxBrain-Bench-JointPlan to evaluate models' joint language–visual planning capabilities.

We evaluate RxBrain on embodied understanding and reasoning, general text-to-image generation, and joint subgoal planning benchmarks. We further assess its effectiveness for robot manipulation in two settings. As a high-level planner, RxBrain provides joint subgoals to guide a downstream policy, improving its average success rate on long-horizon tasks in RMBench (Chen et al., 2026) by 48.0 percentage points over the policy alone. As an action-generation backbone, it enables continuous robot control and achieves strong real-robot performance without additional action pretraining. Together, these results demonstrate RxBrain's strong capabilities in embodied understanding and reasoning, general image generation, joint subgoal planning, and its effectiveness in supporting robot action generation. To summarize, our main contributions are as follows:

- We introduce RxBrain, a unified embodied foundation model that combines embodied understanding and reasoning, general image generation, and joint language–visual planning. We further propose an inference-aware visual state construction strategy to enhance joint planning.
- We develop an automatic pipeline that converts over 50,000 hours of embodied video into 21.51 M trainable segments, aligning textual planning steps with visual state transitions at both step and task levels. We also introduce RxBrain-Bench-JointPlan to evaluate models' joint language–visual planning capabilities.
- We demonstrate competitive performance across embodied understanding and reasoning, general text-to-image generation, and joint subgoal planning benchmarks. We further validate RxBrain in two robot manipulation settings: its joint subgoals substantially improve downstream policy performance on long-horizon tasks, while its use as an action-generation backbone enables effective continuous control on real robots experiments.

2 Related Work

Vision-Language Models for Embodied Understanding and Reasoning. Embodied VLMs ground task reasoning in visual inputs (Driess et al., 2023) and link vision-language knowledge to robot control (Brohan et al., 2023a). RoboBrain combines manipulation planning, affordance perception, and trajectory prediction (Ji et al., 2025; Team, 2025); RynnBrain emphasizes egocentric spatiotemporal grounding and physical-space reasoning (Dang et al., 2026). Other work explicates reasoning through depth-aware perception and spatial trajectories (Lee et al., 2025; Fang et al.,

2026b), or strengthens physical understanding for embodied decisions (Azzolini et al., 2025). These models primarily output language rather than generated future visual states.

Generative World Models. World models predict environmental changes visually or latently. Large-scale video generators learn rich dynamics (Wan et al., 2025; Parker-Holder et al., 2025); V-JEPA 2 predicts spatiotemporal representations without pixel reconstruction (Assran et al., 2025). In robotics, VPP uses video-diffusion predictive features for policy learning (Hu et al., 2024); Ctrl-World enables policy evaluation through action-conditioned imagined rollouts (Guo et al., 2025b); newer models jointly generate future visual states and robot actions (Li et al., 2026; Ye et al., 2026). We focus on joint planning that pairs explicit textual subgoals with corresponding visual goal states.

Unified Models for Embodied Planning and Control. Unified models integrate understanding and generation via unified discrete tokenization (Chameleon Team, 2024) or hybrid language-modeling/diffusion objectives (Zhou et al., 2025a). BAGEL models interleaved text–image sequences for image editing and generation (Deng et al., 2025); Cosmos-3 supports embodied understanding and visual generation through distinct task modes within a unified MoT (Agarwal et al., 2026). LatentUM enables cross-modal reasoning in a shared semantic latent space (Jin et al., 2026). Predictive objectives support unified robot policy learning (Zhang et al., 2025; Bi et al., 2025); InternVLA-A1 coordinates scene understanding, visual foresight, and action execution through specialized experts (Cai et al., 2026). BagelVLA interleaves textual planning and visual forecasting within the action loop for manipulation (Hu et al., 2026). We extend an embodied VLM with general image generation and multi-step language–visual planning while retaining its embodied understanding and reasoning capabilities. We further validate its utility for both subgoal-conditioned control and direct action generation.

3 METHOD

3.1 MODEL

3.1.1 MODEL ARCHITECTURE

Visual representations. Following the vision encoder design of HY-Embodied-0.5 (Team et al., 2026), RxBrain adopts HY-ViT 2.0 (Team et al., 2026) as its single visual encoder for both semantic and VAE-aligned fine-grained feature extraction. VAE-latent supervision during encoder pretraining introduces reconstruction ability, preserving fine-grained image details alongside semantic features. For visual generation, we use the encoder and decoder of a pretrained FLUX VAE (Black Forest Labs, 2025) to improve image quality.

Modality-Aware Mixture-of-Transformers. RxBrain uses a modality-aware Mixture-of-Transformers (MoT) backbone with two attention branches (language and vision) and three Feed-Forward Network (FFN) experts (text, visual understanding, and visual generation). Modality-routed tokens interact through global self-attention. Visual understanding and generation share the vision Transformer and attention projections; generation adds a dedicated FFN with intermediate width expanded from $6{,}144$ to $12{,}288$. The base model contains **6.2 B** parameters, 1.5 B each in the text and visual-understanding branches and 2.4 B in the generation FFN. The remaining parameters reside in shared and peripheral modules, including the visual encoder (≈ 0.45 B), tied input/output embeddings (≈ 0.25 B), and image flow-matching head (≈ 7 M). A frozen VAE tokenizer (≈ 84 M), external to the backbone, produces image latents.

3.1.2 MULTIMODAL GENERATION FORMULATION

RxBrain supports autoregressive text generation, future multi-frame prediction, and interleaved text–vision generation for embodied planning.

Visual Prefix Representation. After computing causal attention over the full multimodal sequence, RxBrain applies additional bidirectional self-attention within each image and replaces the causal-attention outputs at the corresponding visual token positions with the resulting representa-

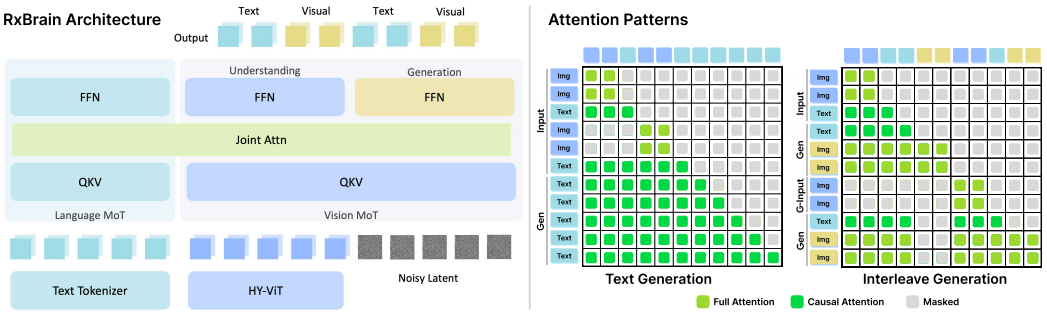


Figure 2: **Architecture and attention patterns of RxBrain.** A modality-routed MoT jointly processes text, visual inputs, and image latents through language and vision branches. Visual understanding and generation share attention projections but use separate FFN experts. Hybrid masks combine causal text decoding, bidirectional within-image attention, and causal dependencies across frames. Interleaved planning alternates autoregressive text with image generation; generated latents are VAE-decoded into images and re-encoded as visual context for subsequent steps.

tions. This enables global information exchange within each image while preserving the sequence's causal structure and independence between visual observations.

Text Generation. Conditioned on a multimodal prefix of language instructions and visual observations, RxBrain generates text autoregressively with causal attention and next-token prediction supervision, supporting question answering and language-based reasoning about embodied observations.

Interleaved Generation. For joint subgoal planning, RxBrain alternates reasoning text and visual state prediction, decoding the trajectory from left to right under causal attention:

$$\texttt{<assistant>}\ r_1\ \texttt{<Image>}\,\mathbf{z}_{1,1:N}\,\texttt{</Image>}\ r_2\ \texttt{<Image>}\,\mathbf{z}_{2,1:N}\,\texttt{</Image>}\ \cdots\ \texttt{<eos>}.$$

Here, r_i is the reasoning text at step i, and $\mathbf{z}_{i,1:N}$ is the flow-matched VAE latent span of the imagined frame. Text is generated by autoregressive next-token prediction; the learned `<Image>` transition token determines when to initiate visual generation.

After emitting `<Image>`, the model synthesizes the latent span through flow matching, conditioned on the preceding multimodal context through global attention. The generated latent is VAE-decoded into an image, re-encoded by the ViT visual encoder, and appended to the context as visual features. The `</Image>` token is then inserted automatically, and autoregressive text decoding resumes.

3.2 Training

RxBrain jointly learns language understanding, visual generation, and interleaved embodied generation through a unified multimodal objective, with generation modes sharing the MoT backbone and using modality-specific supervision.

3.2.1 Training Objective

RxBrain combines autoregressive text generation with flow-matching visual generation. Let E_{vis} denote the visual encoder, and let E_{VAE} and D_{VAE} denote the encoder and decoder of the pretrained FLUX VAE, respectively. For a target text sequence $y = (y_1, \ldots, y_T)$ conditioned on multimodal context c, the next-token prediction loss is $\ell_{\text{text}}(y; c) = -\sum_{j=1}^{T} \log p_\theta(y_j \mid y_{<j}, c)$.

Visual generation. For a target image I, we obtain its clean latent $\mathbf{z} = E_{\text{VAE}}(I)$ and construct the interpolation $\mathbf{z}_t = (1-t)\mathbf{z} + t\epsilon$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and $t \in (0, 1)$ is sampled from a logit-normal distribution. The flow-matching loss is

$$\ell_{\text{flow}}(I; c) = \mathbb{E}_{t,\epsilon}\left[\|v_\theta(\mathbf{z}_t, t, c) - (\epsilon - \mathbf{z})\|_2^2\right]. \tag{1}$$

This objective is used for both image generation and future-frame prediction. At inference, images are synthesized by integrating the learned velocity field from Gaussian noise at $t = 1$ to an image latent at $t = 0$, followed by VAE decoding.

Interleaved generation. For a trajectory of K planning steps, let h_i denote the multimodal context containing the initial instruction and observations, preceding reasoning segments $r_{<i}$, and visual representations $\tilde{I}_{<i}$. At each step, the model generates reasoning text r_i conditioned on h_i, followed by a visual state I_i additionally conditioned on r_i. The joint objective is

$$\mathcal{L}_{\text{interleave}} = \sum_{i=1}^{K} \left[\lambda_t \ell_{\text{text}}(r_i; h_i) + \lambda_v \ell_{\text{flow}}(I_i; h_i, r_i) \right], \tag{2}$$

where λ_t and λ_v weight the text and visual losses.

Inference-aware visual state construction. To align visual context construction between training and inference, we reconstruct preceding ground-truth frames through the VAE before visual encoding during interleaved training. At inference, we instead decode the generated latent span $\mathbf{z}_{i,1:N}$ defined above:

$$\begin{aligned} \tilde{I}_i^{\text{train}} &= E_{\text{vis}}(D_{\text{VAE}}(E_{\text{VAE}}(I_i))) \,, \\ \tilde{I}_i^{\text{infer}} &= E_{\text{vis}}(D_{\text{VAE}}(\mathbf{z}_{i,1:N})) \,. \end{aligned} \tag{3}$$

Thus, training exposes the model to VAE-reconstructed visual histories, matching the decoding-and-encoding pathway used for generated frames at inference.

3.2.2 Training Stages

RxBrain follows a two-stage curriculum, first acquiring general vision-language understanding and generation capabilities, then specializing in embodied world modeling and long-horizon planning.

Stage 1: Joint Vision-Language Pretraining. We jointly train all model components on large-scale text–image pairs and multimodal instruction-following data to establish a unified representation space for language understanding, visual perception, and visual generation. text–image alignment and instruction supervision develop semantic grounding and visual reasoning, while flow matching enables visual-state synthesis. Joint understanding and generation training preserves the base model's language capability.

Stage 2: Embodied Supervised Fine-tuning. We adapt RxBrain through joint training on multimodal understanding and generation tasks, combining general multimodal understanding data, embodied instruction-following examples, and joint text–image subgoal planning trajectories. Embodied instruction-following examples train the model to understand scenes, reason about tasks, and produce textual plans. Joint planning trajectories further train it to interleave textual subgoals with images depicting their intended physical outcomes.

We split each interleaved trajectory into training samples pairing target visual states with their corresponding language context. We jointly train language generation, image generation, and embodied understanding and reasoning tasks. Hyperparameters, data mixtures, and optimization settings are provided in Appendix E.

4 Joint Planning Data Construction

4.1 Data Pipeline

We convert embodied videos into joint text–visual planning supervision by aligning textual planning steps with action-induced visual state transitions, linking task semantics to physical changes. The verified corpus spans four categories—real robots, wrist-mounted human demonstrations (UMI), simulation, and egocentric human activity—with 46 processed source splits totaling $50{,}177.2$ hours (Figure 3). Our three-stage pipeline—Temporal Segment Annotation, Quality Verification, and Segment Structuring—localizes, filters, and organizes segments into four post-training tasks (L0–L3),

from intra-step prediction to final-state imagination. After quality control, 21.51M training segments are retained from the original 28.61M candidates. Pipeline details, dataset descriptions, and the scene taxonomy are provided in Appendix A, Appendix B, and Appendix D.

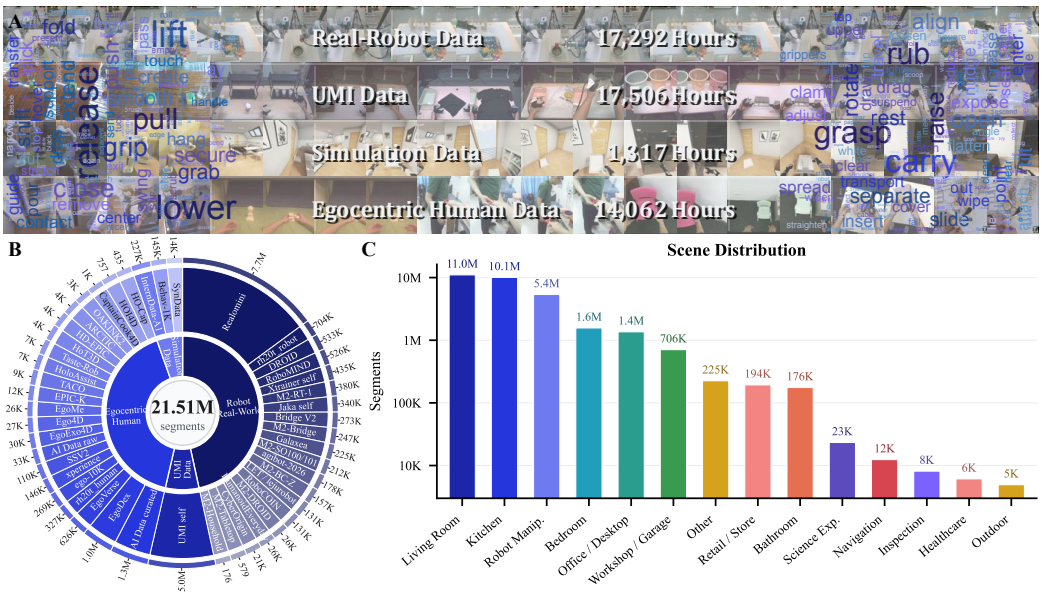


Figure 3: **Data construction overview and statistics.** (A) Video hours by source category. (B) Source/dataset composition. (C) Segments across the fourteen scene categories. Full detail in Appendix A and Appendix B.

4.2 RxBrain-Bench-JointPlan

To fill an evaluation gap in embodied joint text–image subgoal planning, we introduce RxBrain-Bench-JointPlan. Language specifies action logic, constraints, and decisions; images depict world states that subgoal finished. The Free-running evaluation feeds generated text and images into subsequent steps, capturing long-horizon error accumulation. We evaluate Observation Understanding (OU), Subtask Planning (SP), Goal-Image Correctness (GI), Subtask–Goal Consistency (SG), Chain Completion (CC), and Image Similarity (IS), with the weighted planning score

$$S_{\text{plan}} = 0.10\,\text{OU} + 0.25\,\text{SP} + 0.25\,\text{GI} + 0.20\,\text{SG} + 0.10\,\text{CC} + 0.10\,\text{IS}. \tag{4}$$

The first five are MLLM-judged; IS measures the cosine similarity of DINO (Oquab et al., 2023) representation to the corresponding ground-truth states. SP, GI, and SG are averaged over each trajectory. Overall and per-criterion scores distinguish textual-reasoning and visual-prediction failures. Full inputs, held-out sizes, judge settings, scoring details, and baseline configurations are in Appendix F. We further validate the MLLM judge against human assessments on a sampled subset, observing over 95% agreement for each of the five judged criteria in Appendix F.

5 Experiments

We evaluate RxBrain along four axes: (i) its embodied understanding and reasoning abilities across a broad suite of benchmarks; (ii) its general text-to-image generation ability; (iii) its joint subgoal planning ability; and (iv) its effectiveness in robot action generation, both as a high-level planner and as an action-generation backbone.

5.1 Evaluation Tasks

Embodied understanding and reasoning. We evaluate embodied understanding and reasoning into three benchmark groups: *embodied and spatial reasoning* (ERQA (Team et al., 2025a), EmbSpatial (Du et al., 2024), CV-Bench (Tong et al., 2024), SAT (Ray et al., 2024), DA-2k (Yang et al.,

2024b)); *robot affordance, trajectory and placement* (Share-Robot-Affordance and Share-Robot-Trajectory (Ji et al., 2025), RefSpatial (Zhou et al., 2025b), Where2Place (Yuan et al., 2024), and RoboSpatial-Home's (Song et al., 2025) *context*/*compatibility*/*configuration* splits); and *multi-view and 3D understanding* (MindCube (Yin et al., 2025), MMSI-Bench (Yang et al., 2025), ViewSpatial (Li et al., 2025), 3DSRBench (Ma et al., 2024), SITE-Bench-Image (Wang et al., 2025b), VSI-bench (Yang et al., 2024a)). RxBrain uses 4B active parameters during text generation. We compare it with similarly sized general-purpose VLMs (Qwen3.5-2B and Qwen3.5-4B (Team, 2026)), models with enhanced embodied capabilities (RoboBrain2.5-4B (Tan et al., 2026) and Cosmos-3-Edge (Agarwal et al., 2026)), and the unified multimodal model BAGEL (Deng et al., 2025).

Text-to-image generation. We use GenEval (Ghosh et al., 2023), a standard benchmark of prompt-faithful, *open-domain* text-to-image synthesis to verify that RxBrain preserves general-purpose, world-knowledge image generation. We compare against the general unified model BAGEL (Deng et al., 2025), the embodied omni model Cosmos-3-Edge and Cosmos-3-Nano (Agarwal et al., 2026).

Table 1: **Embodied understanding and reasoning benchmarks.** We compare RxBrain (4B active parameters during text generation) with similarly sized VLMs and the unified model BAGEL. Higher is better; best and second-best results per row are **bold** and underlined.

Benchmark	Qwen3.5-4B	Qwen3.5-2B	Bagel	RoboBrain2.5-4B	Cosmos-3-Edge	**RxBrain**
Embodied & spatial reasoning						
ERQA	**45.8**	39.8	39.5	35.8	38.8	42.8
EmbSpatial	74.2	64.3	67.0	80.2	73.2	**82.3**
CV-Bench	85.9	79.0	77.4	87.4	80.8	**88.6**
SAT	**78.3**	63.7	74.3	66.0	62.0	74.3
DA-2k	68.5	63.9	62.9	82.8	74.5	**83.4**
Robot affordance, trajectory & placement						
Share-Robot-Affordance	24.8	18.1	3.2	**50.3**	23.6	11.7
Share-Robot-Trajectory	29.4	28.5	11.8	27.7	33.1	**51.1**
RefSpatial	34.3	24.2	8.6	**62.5**	37.8	34.1
Where2Place	44.3	15.1	8.8	**52.5**	51.1	39.8
RoboSpatial-Home (context)	7.2	6.7	6.2	**56.6**	22.1	28.3
RoboSpatial-Home (compatibility)	50.5	32.4	42.9	36.2	52.4	**67.6**
RoboSpatial-Home (configuration)	82.1	71.5	74.0	83.7	81.3	**86.3**
Multi-view & 3D understanding						
MindCube	38.5	31.9	37.6	43.7	36.7	**47.5**
MMSI-Bench	31.0	26.6	29.8	28.3	31.9	**35.8**
ViewSpatial	42.3	36.6	38.6	41.5	45.7	**47.3**
3DSRBench	37.2	34.2	45.8	52.1	48.4	**54.1**
SITE-Bench-Image	50.7	47.6	51.4	52.1	49.3	**54.9**
VSI-bench	40.6	38.3	34.5	41.0	37.4	**43.5**

Joint subgoal planning. We evaluate joint subgoal planning on RxBrain-Bench-JointPlan, which requires interleaved text-and-image planning under a autoregressive closed-loop rollout. Since no open-source unified model supports this setting, we build baselines from released components: a modular *Qwen-Agent* (Qwen3-VL-2B (Bai et al., 2025) as the reasoner with Qwen-Image(Wu et al., 2025a) as the image generator), an agent built upon the embodied omni model Cosmos-3-Nano (Agarwal et al., 2026), and the unified BAGEL (Deng et al., 2025) run as a planning agent. More details of baselines' construction can be found at Appendix F.

Robot Action Generation We evaluate RxBrain in two settings to assess its effectiveness in robot manipulation. First, RxBrain generates joint text–image subgoal plans as context for a downstream policy. We assess the benefit of this planning on RMBench (Chen et al., 2026), comprising nine long-horizon bimanual manipulation tasks, against X-VLA (Zheng et al., 2026), Mem-0 (Chen et al., 2026), and $\pi_{0.5}$ (Physical Intelligence et al., 2025). Second, we use RxBrain as the backbone to construct an action generation model and evaluate it on three real-world manipulation tasks across two robot embodiments, comparing against π_0 (Black et al., 2024) and $\pi_{0.5}$ (Physical Intelligence et al., 2025).

5.2 EVALUATION RESULTS

Embodied understanding and reasoning. As shown in Table 1, RxBrain leads 12 of 18 entries: all six multi-view and 3D benchmarks, three of five embodied/spatial reasoning benchmarks, trajectory prediction, and RoboSpatial-Home compatibility and configuration. It ranks second on ERQA, SAT (tied), and RoboSpatial-Home context, placing among the top two on 15 of 18 entries. These results demonstrate competitive embodied understanding and reasoning abilities, complementing its general image generation and joint text–image subgoal planning capabilities.

Text-to-image generation. On GenEval, a general text-to-image generation benchmark (Table 2), RxBrain matches BAGEL's overall score (0.82), outperforming Cosmos-3-Nano (0.72) and Cosmos-3-Edge (0.15). It leads on counting (0.87), position (0.74), and color attribution (0.69). These results suggest that RxBrain can leverage general world knowledge, including object attributes and spatial relations, for text-to-image generation.

Table 2: **GenEval result**. Higher is better for all metrics. The best and second-best results are highlighted in **bold** and underline.

Model	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attri.	Overall
Bagel	0.99	0.94	0.81	0.88	0.64	0.63	**0.82**
Cosmos-3-Nano	0.98	0.85	0.79	0.60	0.54	0.54	<u>0.72</u>
Cosmos-3-Edge	0.42	0.06	0.31	0.02	0.01	0.05	0.15
RxBrain	0.99	0.89	0.87	0.76	0.74	0.69	**0.82**

Joint subgoal planning. On RxBrain-Bench-JointPlan (Table 3), RxBrain achieves the best overall performance among the compared models ($S_{\text{plan}} = 0.68$), outperforming Cosmos3-Nano agent, BAGEL, and Qwen-Agent. These results highlight its ability to generate joint text and image subgoal planning. Qualitative examples of the generated text–image plans are provided in Appendix H.

Table 3: **Comparison on RxBrain-Bench-JointPlan.** Obs: Observation Understanding (OU); Plan: Subtask Planning (SP); GoalImg: Goal-Image Correctness (GI); Consist: Subtask–Goal Consistency (SG); Chain: Chain Completion (CC); DINO: Image Similarity (IS), measured by DINO cosine similarity. The first five criteria are MLLM-judged. S_{plan} is their weighted aggregate with IS. Higher scores are better; best scores are in **bold**. n denotes the number of evaluated samples.

Method	S_{plan}	Obs	Plan	GoalImg	Consist	Chain	DINO	n
RxBrain	**0.68**	**0.83**	**0.78**	**0.52**	**0.67**	**0.69**	0.72	540
Cosmos3-Nano (agent)	0.52	0.67	0.62	0.26	0.47	0.63	**0.75**	540
BAGEL	0.50	0.58	0.55	0.25	0.60	0.56	0.67	540
Qwen-Agent	0.43	0.57	0.53	0.20	0.44	0.53	0.52	540

Robot manipulation For joint subgoal conditioned control, RxBrain identifies the current stage of a long-horizon task and guides the downstream policy with subtask instructions and predicted images of subtask completion. On RMBench (Table 4), we conduct a controlled policy-conditioning ablation in which all $\pi_{0.5}$ variants are fine-tuned with the same data and training settings, differing only in whether they receive subgoal from RxBrain. Under this matched setup, the average success rate increases from 12.6% to 23.6% with text-only subgoals and 60.6% with joint text–image subgoals. Appendix G.1 details the integration of planning–policy.

For direct action generation, we evaluate each policy with 100 real-robot trials per task across two embodiments (Figure 4). Both RxBrain and the baselines are fine-tuned on the same real world robot data. Without additional action pretraining, RxBrain achieves 97%, 95%, and 68% success on *Set the Table*, *Fold and Store Glasses*, and *Pick Trash*, respectively, averaging 87% versus π_0's 68% and $\pi_{0.5}$'s 82%. Architecture details appear in Appendix G.2.

Table 4: **Results on RMBench.** Success rates (%); best per row in **bold**. Blue parentheses report nine-task average gains over $\pi_{0.5}$ in percentage points. RxBrain improves $\pi_{0.5}$'s average success rate with text-only subgoals, while joint text–image subgoal planning yields substantially larger gains.

Task	X-VLA	Mem-0	$\pi_{0.5}$	RxBrain + $\pi_{0.5}$	
				Text goal only	**Full**
Observe and Pick Up	9.0	4.0	24.0	20.0	**60.0**
Rearrange Blocks	13.0	89.0	12.0	13.0	**96.0**
Put Back Block	18.0	90.0	12.0	30.0	**91.0**
Swap Blocks	16.0	**67.0**	18.0	58.0	57.0
Swap T	3.0	14.0	18.0	9.0	**99.0**
Battery Try	26.0	28.0	29.0	**43.0**	36.0
Blocks Ranking Try	1.0	18.0	0.0	12.0	**27.0**
Cover Blocks	2.0	68.0	0.0	26.0	**79.0**
Press Button	0.0	0.0	0.0	**1.0**	0.0
Average	9.8	42.0	12.6	23.6 **(+11.0)**	**60.6 (+48.0)**

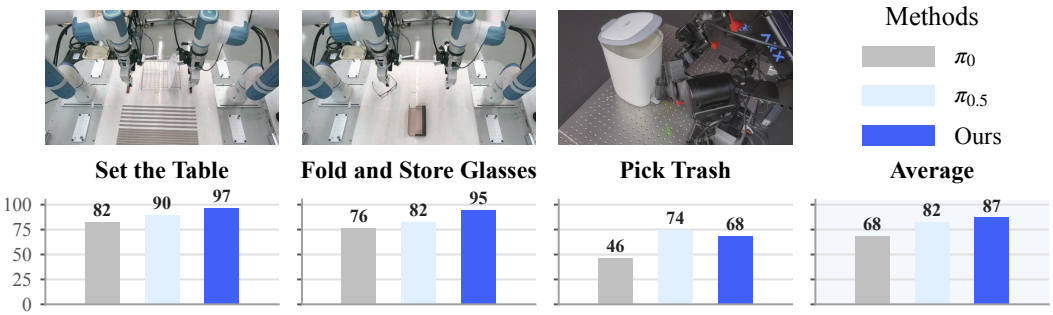


Figure 4: **Real-world robot evaluation.** We compare our method with π_0 and $\pi_{0.5}$ on three manipulation tasks. Our method achieves the highest success rate on average.

5.3 Ablation Study

Table 5: **Inference-aware visual state construction ablation on RxBrain-Bench-JointPlan.** Red parentheses indicate score decreases relative to the full model.

Method	S_{plan}	Obs	Plan	GoalImg	Consist	Chain	DINO
RxBrain	0.68	0.83	0.78	0.52	0.67	0.69	0.72
w/o recon.	0.64 (-0.04)	0.78 (-0.05)	0.71 (-0.07)	0.48 (-0.04)	0.65 (-0.02)	0.63 (-0.06)	0.72

Inference-aware visual state construction. We ablate the strategy described in Section 3.2.1 by removing VAE reconstruction of preceding ground-truth frames, while keeping the training data and all other training settings unchanged. Table 5 shows that this reduces S_{plan} from 0.68 to 0.64. All five MLLM-judged criteria decline, with the largest drops in subtask planning (-0.07) and chain completion (-0.06), while DINO similarity remains unchanged. These results support the benefit of inference-aware visual state construction for joint planning, particularly in selecting subtasks and completing the overall task.

6 Conclusion

We presented RxBrain, a unified embodied foundation model that combines understanding and reasoning with general image generation and joint language–visual planning. Our automatic data pipeline provides planning supervision from over 50,000 hours of embodied video, while inference-aware visual state construction further improves joint planning. Experiments demonstrate competitive performance across understanding, generation, and planning benchmarks. Beyond these capabilities, RxBrain improves long-horizon manipulation through joint subgoal guidance and serves as a backbone for effective real-robot action generation without additional action pretraining.

AI USE STATEMENT

In this work, we used generative AI tools to assist with searching for information, identifying relevant literature, and editing author-written text to improve its language and readability. We also used multimodal large language models (MLLMs) for automatic data annotation, refinement, and quality verification. We used MLLM judge for evaluation. To validate the judge's assessments, we conducted human evaluation experiments on a sampled subset and measured human–judge agreement, observing close agreement across the five evaluated criteria.

We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses publicly available resources and newly collected robotic task data. Activities involving recruited contributors for data collection or annotation were covered by ethics approval. Real-world robotic experiments were conducted in controlled environments with human supervision, operational safety constraints, and separation between personnel and robot operating areas. These precautions mitigate experimental risks but do not establish safety for unsupervised deployment. Learning-based robotic systems may generate unsafe actions under distribution shift and may reflect biases in their training data. Accordingly, deployment beyond the evaluated settings requires additional safety assessment and appropriate human oversight.

REPRODUCIBILITY STATEMENT

To support reproducibility, we document the proposed method, model architecture, data sources and preprocessing procedures, training configurations, and evaluation protocols in the paper and its appendix. The source code is provided in the supplementary materials to facilitate independent reproduction of the reported results.

REFERENCES

Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, et al. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025.

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

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.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Shangchen Han, Fan Zhang, Linguang Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, Jakob Julian Engel, and Tomas Hodan. HOT3D: Hand and object tracking in 3d from egocentric multi-view videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model, 2025. URL https://arxiv.org/abs/2512.13030.

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.

Black Forest Labs. FLUX.1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023a.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems (RSS)*, 2023b.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xu Huang, Shu Jiang, et al. AgiBot World Colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

Build AI. Egocentric-10K. https://huggingface.co/datasets/builddotai/Egocentric-10K, 2025. Hugging Face dataset, approximately 10,000 hours of egocentric factory video.

Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. https://github.com/kakaobrain/coyo-dataset, 2022.

Junhao Cai, Zetao Cai, Jiafei Cao, Yilun Chen, Zeyu He, Lei Jiang, Hang Li, Hengjie Li, Yang Li, Yufei Liu, et al. Internvla-a1: Unifying understanding, generation and action for robotic manipulation. *arXiv preprint arXiv:2601.02456*, 2026.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025.

Tianxing Chen, Yuran Wang, Mingleyang Li, Yan Qin, Hao Shi, Zixuan Li, Yifan Hu, Yingsheng Zhang, Kaixuan Wang, Yue Chen, et al. Rmbench: Memory-dependent robotic manipulation benchmark with insights into policy design. *arXiv preprint arXiv:2603.01229*, 2026.

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. In *Robotics: Science and Systems (RSS)*, 2024.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision (IJCV)*, 130(1):33–55, 2022.

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.

Ronghao Dang, Jiayan Guo, Bohan Hou, Sicong Leng, Kehan Li, Xin Li, Jiangpin Liu, Yunxuan Mao, Zhikai Wang, Yuqian Yuan, et al. Rynnbrain: Open embodied foundation models. *arXiv preprint arXiv:2602.14979*, 2026.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. PaLM-E: An embodied multimodal language model. In *International Conference on Machine Learning (ICML)*, 2023.

Mengfei Du, Binhao Wu, Zejun Li, Xuanjing Huang, and Zhongyu Wei. Embspatial-bench: Benchmarking spatial understanding for embodied tasks with large vision-language models, 2024. URL https://arxiv.org/abs/2406.05756.

Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J. Black, and Otmar Hilliges. ARCTIC: A dataset for dexterous bimanual hand-object manipulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. RH20T: A comprehensive robotic dataset for learning diverse skills in one-shot. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Haoquan Fang, Jiafei Duan, Donovan Clay, Sam Wang, Shuo Liu, Weikai Huang, Xiang Fan, Wei-Chuan Tsai, Shirui Chen, Yi Ru Wang, Shanli Xing, Jaemin Cho, Jae Sung Park, Ainaz Eftekhar, Peter Sushko, Karen Farley, Angad Wadhwa, Cole Harrison, Winson Han, Ying-Chun Lee, Eli VanderBilt, Rose Hendrix, Suveen Ellawela, Lucas Ngoo, Joyce Chai, Zhongzheng Ren, Ali Farhadi, Dieter Fox, and Ranjay Krishna. Molmoact2: Action reasoning models for real-world deployment, 2026a. URL https://arxiv.org/abs/2605.02881.

Haoquan Fang, Jiafei Duan, Donovan Clay, Sam Wang, Shuo Liu, Weikai Huang, Xiang Fan, Wei-Chuan Tsai, Shirui Chen, Yi Ru Wang, et al. Molmoact2: Action reasoning models for real-world deployment. *arXiv preprint arXiv:2605.02881*, 2026b.

Galaxea Team. Galaxea open-world dataset and G0 dual-system VLA model. *arXiv preprint arXiv:2509.00576*, 2025.

GenRobot. RealOmni open dataset. https://www.genrobot.ai/data/open-dataset, 2025. Open embodied-AI dataset, dual-hand household manipulation collected with the GenDAS handheld gripper.

Dhruba Ghosh, Hanna Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment, 2023. URL https://arxiv.org/abs/2310.11513.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "something something" video database for learning and evaluating visual common sense. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, et al. Ego-Exo4D: Understanding skilled human activity from first- and third-person perspectives. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Jiawei Guo, Tianyu Zheng, Yizhi Li, Yuelin Bai, Bo Li, Yubo Wang, King Zhu, Graham Neubig, Wenhu Chen, and Xiang Yue. Mammoth-vl: Eliciting multimodal reasoning with instruction tuning at scale. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13869–13920, 2025a.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. *arXiv preprint arXiv:2510.10125*, 2025b.

Ryan Hoque, Peide Huang, David J. Yoon, Mouli Sivapurapu, and Jian Zhang. EgoDex: Learning dexterous manipulation from large-scale egocentric video. *arXiv preprint arXiv:2505.11709*, 2025.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *arXiv preprint arXiv:2412.14803*, 2024.

Yucheng Hu, Jianke Zhang, Yuanfei Luo, Yanjiang Guo, Xiaoyu Chen, Xinshu Sun, Kun Feng, Qingzhou Lu, Sheng Chen, Yangang Zhang, Wei Li, and Jianyu Chen. Bagelvla: Enhancing long-horizon manipulation via interleaved vision-language-action generation, 2026. URL https://arxiv.org/abs/2602.09849.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachlan Groom, Haroun Habeeb, Hunter Hancock, Karol Hausman, Gashon Hussein, Victor Hwang, Brian Ichter, Connor Jacobsen, Szymon Jakubczak, Rowan Jen, Tim Jones, Gregg Kammerer, Ben Katz, Liyiming Ke, Mairbek Khadikov, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Brendon LeCount, Sergey Levine, Xinyu Li, Adrian Li-Bell, Vladislav Lialin, Zhonglin Liang, Wallace Lim, Yao Lu, Enyu Luo, Vishnu Mano, Nandan Marwaha, Aikys Mongush, Liam Murphy, Suraj Nair, Tyler Patterson, Karl Pertsch, Allen Z. Ren, Gavin Schelske, Charvi Sharma, Baifeng Shi, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Jiaming Tang, Jimmy Tanner, Shalom Tekeste, Marcel Torne, Kyle Vedder, Quan Vuong, Anna Walling, Haohuan Wang, Jason Wang, XuDong Wang, Chris Whalen, Samuel Whitmore, Blake Williams, Charles Xu, Sukwon Yoo, Lili Yu, Wuming Zhang, Zhuoyang Zhang, and Ury Zhilinsky. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities, 2026. URL https://arxiv.org/abs/2604.15483.

InternRobotics. InternData-A1. https://huggingface.co/datasets/InternRobotics/InternData-A1, 2025. Hugging Face dataset, hybrid synthetic-real manipulation across four embodiments.

Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. BC-Z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning (CoRL)*, 2022.

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

Jiachun Jin, Zetong Zhou, Xiao Yang, Hao Zhang, Pengfei Liu, Jun Zhu, and Zhijie Deng. Latentum: Unleashing the potential of interleaved cross-modal reasoning via a latent-space unified model. *arXiv preprint arXiv:2604.02097*, 2026.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems (RSS)*, 2024.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, and Ranjay Krishna. MolmoAct: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025.

LejuRobotics. LET: Full-size humanoid robot real-world dataset. https://huggingface.co/datasets/LejuRobotics/LET-Base-Dataset, 2025. Hugging Face dataset, Kuavo humanoid teleoperation.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. BEHAVIOR-1K: A

benchmark for embodied AI with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning (CoRL)*, 2022.

Dingming Li, Hongxing Li, Zixuan Wang, Yuchen Yan, Hang Zhang, Siqi Chen, Guiyang Hou, Shengpei Jiang, Wenqi Zhang, Yongliang Shen, et al. Viewspatial-bench: Evaluating multi-perspective spatial localization in vision-language models. *arXiv preprint arXiv:2505.21500*, 2025.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

Yun Liu, Haolin Yang, Xu Si, Ling Liu, Zipeng Li, Yuxiang Zhang, Yebin Liu, and Li Yi. TACO: Benchmarking generalizable bimanual tool-action-object understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. HOI4D: A 4d egocentric dataset for category-level human-object interaction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Wufei Ma, Haoyu Chen, Guofeng Zhang, Celso M de Melo, Alan Yuille, and Jieneng Chen. 3dsr-bench: A comprehensive 3d spatial reasoning benchmark. *arXiv preprint arXiv:2412.07825*, 2024.

OpenAI. GPT-5.5 system card, April 2026. URL `https://openai.com/index/gpt-5-5-system-card/`.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Jack Parker-Holder, Shlomi Fruchter, et al. Genie 3: A new frontier for world models. *Google DeepMind Blog*, 4, 2025.

Rohith Peddi, Shivvrat Arya, Bharath Challa, Likhitha Pallapothula, Akshay Vyas, Bhavya Gouripeddi, Qifan Zhang, Jikai Wang, Vasundhara Komaragiri, Eric Ragan, Nicholas Ruozzi, Yu Xiang, and Vibhav Gogate. CaptainCook4D: A dataset for understanding errors in procedural activities. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024.

Toby Perrett, Ahmad Darkhalil, Saptarshi Sinha, Omar Emara, Sam Pollard, Kranti Kumar Parida, Kaiting Liu, Prajwal Gatti, Siddhant Bansal, Kevin Flanagan, Jacob Chalk, Zhifan Zhu, Rhodri Guerrier, Fahd Abdelazim, Bin Zhu, Davide Moltisanti, Michael Wray, Hazel Doughty, and Dima Damen. HD-EPIC: A highly-detailed egocentric video dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Chelsea Finn, Sergey Levine, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

PsiBotAI. SynData: A large-scale real-world multimodal dataset for embodied intelligence. `https://huggingface.co/datasets/PsiBotAI/SynData`, 2025. Hugging Face dataset, real-world human manipulation captured with an exoskeleton-glove system.

Heqian Qiu, Zhaofeng Shi, Lanxiao Wang, Huiyu Xiong, Xiang Li, and Hongliang Li. EgoMe: A new dataset and challenge for following me via egocentric view in real world. *arXiv preprint arXiv:2501.19061*, 2025.

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

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. Oral Presentation.

Huajie Tan, Enshen Zhou, Zhiyu Li, Yijie Xu, Yuheng Ji, Xiansheng Chen, Cheng Chi, Pengwei Wang, Huizhu Jia, Yulong Ao, Mingyu Cao, Sixiang Chen, Zhe Li, Mengzhen Liu, Zixiao Wang, Shanyu Rong, Yaoxu Lyu, Zhongxia Zhao, Peterson Co, Yibo Li, Yi Han, Shaoxuan Xie, Guocai Yao, Songjing Wang, Leiduo Zhang, Xi Yang, Yance Jiao, Donghai Shi, Kunchang Xie, Shaokai Nie, Chunlei Men, Yonghua Lin, Zhongyuan Wang, Tiejun Huang, and Shanghang Zhang. Robobrain 2.5: Depth in sight, time in mind, 2026. URL https://arxiv.org/abs/2601.14352.

BAAI RoboBrain Team. RoboBrain 2.0 technical report. *arXiv preprint*, 2025.

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*, 2025a.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025b.

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

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

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

Homer Walke, Kevin Black, Tony Z. Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, Abraham Lee, Kuan Fang, Chelsea Finn, and Sergey Levine. BridgeData V2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Jikai Wang, Qifan Zhang, Yu-Wei Chao, Bowen Wen, Xiaohu Guo, and Yu Xiang. HO-Cap: A capture system and dataset for 3d reconstruction and pose tracking of hand-object interaction. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025a.

Wenqi Wang, Reuben Tan, Pengyue Zhu, Jianwei Yang, Zhengyuan Yang, Lijuan Wang, Andrey Kolobov, Jianfeng Gao, and Boqing Gong. Site: towards spatial intelligence thorough evaluation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9058–9069, 2025b.

Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, Neel Joshi, and Marc Pollefeys. HoloAssist: An egocentric human interaction dataset for interactive AI assistants in the real world. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025a.

Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. RoboMIND: Benchmark on multi-embodiment intelligence normative data for robot manipulation. In *Robotics: Science and Systems (RSS)*, 2025b.

Shihan Wu, Xuecheng Liu, Shaoxuan Xie, Pengwei Wang, Xinghang Li, Bowen Yang, Zhe Li, Kai Zhu, Hongyu Wu, Yiheng Liu, Zhaoye Long, Runtian Xu, Yue Wang, Chong Liu, Dihan Wang, Ziqiang Ni, Xiang Yang, You Liu, Ruoxuan Feng, Lei Zhang, Denghang Huang, Chenghao Jin, Anlan Yin, Xinlong Wang, Zhenguo Sun, Junkai Zhao, Mengfei Du, Mingyu Cao, Xiansheng Chen, Hongyang Cheng, Xiaojie Zhang, Yankai Fu, Ning Chen, Cheng Chi, Sixiang Chen, Huaihai Lyu, Xiaoshuai Hao, Yequan Wang, Bo Lei, Dong Liu, Xi Yang, Yance Jiao, Tengfei Pan, Yunyan Zhang, Songjing Wang, Ziqian Zhang, Xu Liu, Ji Zhang, Caowei Meng, Zhizheng Zhang, Jiyang Gao, Song Wang, Xiaokun Leng, Zhiqiang Xie, Zhenzhen Zhou, Peng Huang, Wu Yang, Yandong Guo, Yichao Zhu, Suibing Zheng, Hao Cheng, Xinmin Ding, Yang Yue, Huanqian Wang, Chi Chen, Jingrui Pang, YuXi Qian, Haoran Geng, Lianli Gao, Haiyuan Li, Bin Fang, Gao Huang, Yaodong Yang, Hao Dong, He Wang, Hang Zhao, Yadong Mu, Di Hu, Hao Zhao, Tiejun Huang, Shanghang Zhang, Yonghua Lin, Zhongyuan Wang, and Guocai Yao. Robocoin: An open-sourced bimanual robotic data collection for integrated manipulation, 2026. URL https://arxiv.org/abs/2511.17441.

Jihan Yang, Shusheng Yang, Anjali Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in Space: How Multimodal Large Language Models See, Remember and Recall Spaces. *arXiv preprint arXiv:2412.14171*, 2024a.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024b.

Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, Dahua Lin, Tai Wang, and Jiangmiao Pang. Mmsi-bench: A benchmark for multi-image spatial intelligence. *arXiv preprint arXiv:2505.23764*, 2025.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies, 2026. URL https://arxiv.org/abs/2602.15922.

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.

Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024.

Xinyu Zhan, Lixin Yang, Yifei Zhao, Kangrui Mao, Hanlin Xu, Zenan Lin, Kailin Li, and Cewu Lu. OAKINK2: A dataset of bimanual hands-object manipulation in complex task completion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Jianke Zhang, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. Up-vla: A unified understanding and prediction model for embodied agent, 2025. URL https://arxiv.org/abs/2501.18867.

Hongxiang Zhao, Xingchen Liu, Mutian Xu, Yiming Hao, Weikai Chen, and Xiaoguang Han. TASTE-Rob: Advancing video generation of task-oriented hand-object interaction for generalizable robotic manipulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

Zhenyu Zhao, Hongyi Jing, Xiawei Liu, Jiageng Mao, Abha Jha, Hanwen Yang, Rong Xue, Sergey Zakharov, Vitor Guizilini, and Yue Wang. Humanoid everyday: A comprehensive robotic dataset for open-world humanoid manipulation. *arXiv preprint arXiv:2510.08807*, 2025b.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 60580–60606, 2026.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. In *International Conference on Learning Representations*, volume 2025, pp. 6446–6469, 2025a.

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*, 2025b.

A JOINT PLANNING DATA CONSTRUCTION: PIPELINE DETAILS

Section 4 summarizes the joint text-visual planning corpus and its three-stage construction pipeline. This appendix gives the pipeline in full: Temporal Segment Annotation localizes atomic planning steps into segments (Section A.1), Quality Verification filters these segments with an MLLM-based verifier (Section A.2), and Segment Structuring organizes the retained segments into four post-training tasks, L0–L3 (Section A.3).

A.1 TEMPORAL SEGMENT ANNOTATION

Temporal Segment Annotation converts long, untrimmed embodied task videos into localized planning steps, as shown in Figure 5. Each localized step contains a short step name, a detailed textual description, and two visual anchors: a start frame and an end frame that make the physical state change visible. We denote the multimodal large language model used for semantic decisions as Φ. The overall process first samples visually informative keyframes, then asks Φ to propose candidate planning steps, and finally refines the start and end boundaries of each step. A raw video is an ordered sequence $V = (f_1, \ldots, f_N)$ recorded at ν frames per second, with timestamp $\tau_j = j/\nu$. We define an **atomic planning step** as one localized step in a task whose effect can be observed through a clear visual state change. A localized segment is represented as

$$S_i = \big(I_i^{\text{start}},\, I_i^{\text{end}},\, a_i,\, d_i,\, m_i\big), \tag{5}$$

where I_i^{start} and I_i^{end} are the start and end frames, a_i is a short step name, d_i is a detailed description of the step, and m_i records source, viewpoint, index, and temporal metadata. The annotation of a video is a temporally ordered sequence of segments $\{S_i\}_{i=1}^{M}$. Neighboring segments may overlap in time, which allows the same video interval to be reused when multiple planning steps share visual context, while their indices still follow the chronological order of the task. Equation 5 is the unit consumed by Quality Verification and Segment Structuring.

Keyframe sampling. Long videos often contain many static intervals, so we allocate the MLLM context to frames where visible changes are more likely to occur. Instead of uniformly sampling the whole video, we first decode the clip at a reduced frame rate and score adjacent frames by visual difference. After smoothing and pruning nearby peaks, we keep a compact set of keyframes that covers both major visual changes and long temporal gaps. The first and last frames are also included, and the final anchor set is capped to fit the global keyframe budget $\mathcal{G}$. This source-agnostic sampler is applied to all data sources with the same empirical constants.

Candidate step proposal. Given the ordered global keyframes $\mathcal{G}$, Φ first judges whether the clip is suitable for annotation. Clips that are too dark, heavily cluttered, dominated by tiny objects, or otherwise difficult to interpret are marked as unannotatable and produce no steps. For annotatable clips, Φ returns a short clip summary, a high-level goal, the camera viewpoint, and an ordered list of candidate planning steps. Each candidate step $c_i = (a_i, d_i, p_i, q_i)$ contains a short step name a_i, a detailed description d_i, and coarse start and end indices (p_i, q_i) in $\mathcal{G}$. The prompt asks Φ to follow the atomic planning step definition and to keep the coarse candidates in chronological order.

Boundary refinement. The candidate ranges provide a task-level prior, but their sparse keyframes may not capture the exact start or completion of a state change. We therefore refine each coarse segment locally. Around the coarse end, we sample a small set of candidate frames and ask Φ to select the first frame where the completed state is visible. Around the coarse start, we sample another local candidate set and ask Φ to select the start frame, using the chosen end frame as reference. This bidirectional refinement gives each planning step a clear pair of visual anchors while allowing adjacent steps to overlap when they share temporal context. The final start and end frames are rendered at native resolution for downstream training. Additional implementation details are provided in Section C.1.

Output. The annotation stage outputs a temporally ordered sequence of localized planning segments $\{S_i\}_{i=1}^{M}$ for each source video. Each segment contains start and end visual anchors, a short step name, a detailed description, and metadata. Quality Verification (Section A.2) filters these segments, and Segment Structuring (Section A.3) organizes the retained segments into model-ready training samples.

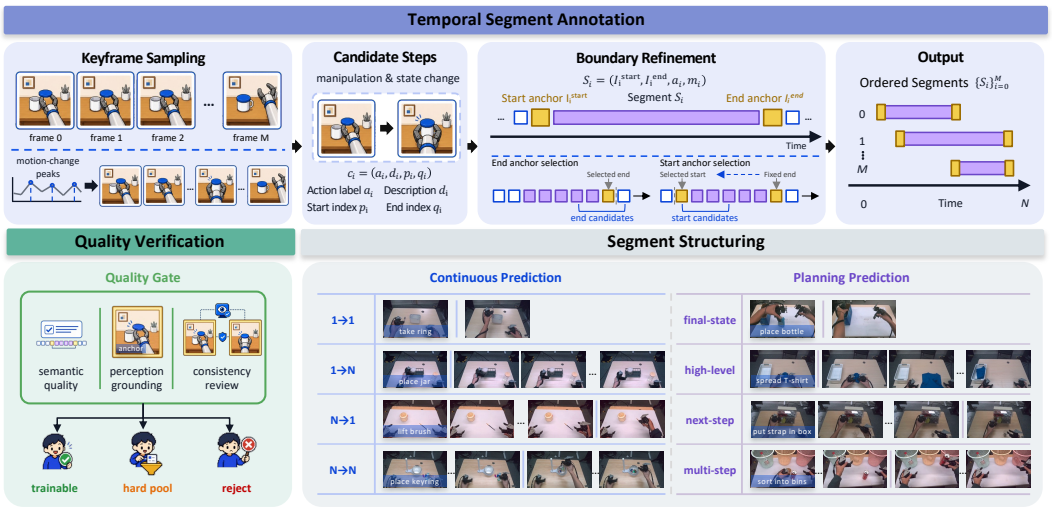


Figure 5: **Data construction pipeline.** Raw embodied task videos are converted into joint text-visual planning samples through three stages: Temporal Segment Annotation, Quality Verification, and Segment Structuring. The final samples are organized into L0–L3 tasks, covering intra-step world state prediction, step-level planning, subgoal planning, and final-state imagination.

A.2 Quality Verification

Quality Verification uses an MLLM-based verifier to decide whether each localized segment S_i from Equation 5 is suitable for training. A valid segment should contain a clear visual state transition between the start anchor I_i^{start} and the end anchor I_i^{end}, and its textual description should correctly describe the observed planning step. This verification ensures that the retained segments can supervise world state prediction and joint subgoal planning, rather than introducing noisy text-visual pairs.

The verifier checks three aspects in one MLLM call: semantic quality, visual grounding, and consistency between the text and the visual transition. Each rejected segment is associated with a specific review item, making the filtering decision interpretable instead of relying only on an opaque score. The complete review protocol and item-level rejection criteria are provided in Section C.2 and Table 6.

For semantic quality, each segment is scored on a six-level scale from 5 (clean and usable) to 0 (missing or unreadable). Segments with scores 4–5 are kept only if they also pass all consistency checks, while segments with scores 0–3 are rejected. We additionally apply a simple near-duplicate filter that removes pairs whose start and end anchors are more than 99.5% identical, since such pairs usually do not contain a meaningful state change.

When the visual transition is valid but the textual description d_i is inaccurate or incomplete, the verifier corrects d_i instead of discarding the segment. We also allow certain apparent object disappearances when they are consistent with the planning step, such as pick-up, move-away, or withdraw behaviors. After verification, 75.18% of candidate segments are retained for training.

A.3 Segment Structuring

Segment Structuring organizes verified segments into joint text-visual planning samples for RxBrain, following the packed-sequence format used in training. The goal is to construct training samples at different temporal scales, from the state change inside a single planning step to the final state of a complete task. Each sample contains a task wrapper, a goal condition, textual planning information, and visual context or target frames. We use t_i to denote the text condition associated with segment i, which can be a short step name a_i, a detailed description d_i, or an LLM-generated subgoal, depending on the task.

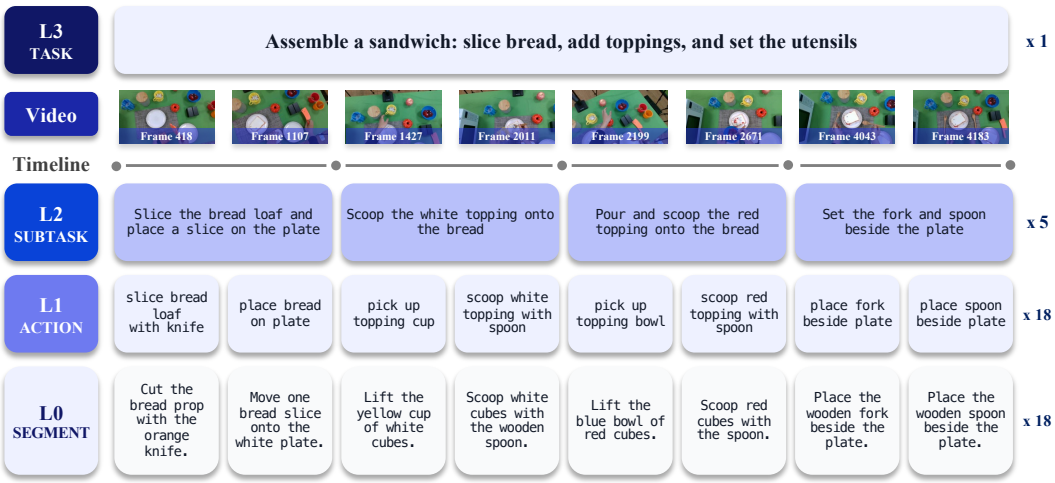


Figure 6: **Hierarchical data construction pipeline.** Verified planning segments are organized into four levels of supervision. L0 learns the visual state change inside a single planning step, while L1 links consecutive steps for step-level joint planning. L2 groups neighboring steps into subgoals, and L3 connects the initial observation with the final task state.

We organize the samples into four post-training tasks, corresponding to the L0–L3 levels in Figure 6. L0 and L1 are both built from the verified atomic planning segments. L0 uses each segment independently to learn intra-step visual state changes, while L1 links consecutive segments to learn step-level planning. Above this shared segment layer, L2 groups adjacent steps into higher-level subgoals, and L3 summarizes the whole video into a final task goal. This hierarchy allows RxBrain to learn world state prediction and joint subgoal planning at multiple levels of granularity.

L0 segment. L0 samples model the visual change inside a single atomic planning step. The basic form takes the start frame I_i^{start} and a text condition t_i as input, and predicts the end frame I_i^{end}:

$$[I_i^{\text{start}}, t_i] \rightarrow I_i^{\text{end}}.$$

We also create variants with richer visual context or denser targets. Some samples prepend pre-step history frames I_i^{pre}, while others include intermediate frames I_i^{mid} as additional targets. These variants teach the model not only the final outcome of a planning step, but also the intermediate visual dynamics leading to that outcome.

L1 step-level planning. L1 samples connect consecutive atomic planning steps. Given visual context and a goal condition, the model predicts a sequence of future planning steps and their corresponding visual outcomes. Each predicted step is represented by a text description and an end frame:

$$[C_{1:m}, g_{\text{seg}}] \rightarrow [T_1, I_1^{\text{end}}, \ldots, T_k, I_k^{\text{end}}],$$

where $C_{1:m}$ denotes visual context frames, g_{seg} denotes the goal condition for the segment window, and T_i denotes the predicted step text. This task teaches the model to jointly produce textual planning steps and visual state transitions.

We construct three layouts for L1. The first is goal-to-future planning, where the model starts from the initial visual context and predicts multiple future steps. The second is history-to-next-step planning, where the model observes completed steps and predicts the next step and its resulting state. The third is history-to-future planning, where the model conditions on several completed steps and predicts a sequence of subsequent steps. These layouts expose the model to both forward planning from a goal and continuation planning from partial progress.

L2 subgoal planning. L2 samples operate at the subgoal level. Starting from the sequence of verified planning steps, an LLM groups adjacent steps into a smaller number of high-level subgoals. Each subgoal is paired with a boundary frame that shows the state after the subgoal has been

completed. The construction mirrors L1, but replaces step-level texts and end frames with subgoal descriptions and subgoal boundary frames:

$$[B_{1:m}, g_{\text{seg}}] \rightarrow [H_1, B_1, \ldots, H_k, B_k],$$

where $B_{1:m}$ denotes visual context at the subgoal level and H_j denotes a high-level subgoal description. This task encourages RxBrain to plan over coarser task stages rather than only local step transitions.

L3 final-state imagination. L3 samples model the final outcome of the whole task. Given the initial frame I_0^{start} and a concise task goal g_{task}, the model predicts the final frame I_T^{end}:

$$[I_0^{\text{start}}, g_{\text{task}}] \rightarrow I_T^{\text{end}}.$$

The task goal is generated by an LLM from the full sequence of subgoals and summarizes the intended final objective of the video. Each video contributes at most one L3 sample. This task teaches the model to imagine long-range task outcomes without requiring every intermediate step to be specified.

B Data Sources

Section 4 summarizes the corpus by four source categories. This appendix supplements that summary with per-dataset detail omitted there for space. Each processed source split is described by its collection setup, why it was included, and the hours retained after the verification of Section A.2. The final scope contains 46 splits and 50,177 retained hours, distributed as 20 Real-Robot, 1 UMI, 3 Simulation, and 22 Egocentric Human splits.

B.1 Real-Robot Data

RealOmni-Open. RealOmni-Open contributes over 13 thousand hours of dual-hand household manipulation collected with a handheld DAS gripper, with fisheye video, trajectories, and tactile signals (GenRobot, 2025). Its scale and tactile modality extend the corpus with contact-rich household manipulation priors that complement the largely RGB sources, and we use 9,005.9 hours after verification.

JAKA. JAKA data is self-collected on single-arm JAKA platforms, recording RGB observations with synchronized robot states and actions for deployment-aligned training. We use 2,558.2 hours after verification.

Xtrainer. Xtrainer data is self-collected on dual-arm Xtrainer platforms for bimanual tasks, recording RGB observations with synchronized robot states and actions for deployment-aligned training. We use 1,515.9 hours after verification.

RoboMIND. RoboMIND provides 107 thousand real-world trajectories over 479 tasks and 96 object classes on four embodiments including the Franka Panda, UR-5e, AgileX dual-arm, and a Tien Kung humanoid, with failure demonstrations alongside successful ones (Wu et al., 2025b). Its standardized collection protocol across embodiments gives consistent cross-embodiment supervision, and we use 888.0 hours after verification.

Galaxea Open-World. The Galaxea Open-World Dataset provides over 500 hours of real-world mobile manipulation on one uniform embodiment, with subtask-level language annotations across residential, kitchen, retail, and office scenes (Galaxea Team, 2025). Its diverse scene coverage with consistent embodiment and fine-grained language annotations makes it a strong source of long-horizon mobile manipulation supervision, and we use 536.0 hours after verification.

MolmoAct2 RT-1. This processed source split uses RT-1 real-robot manipulation episodes collected on a mobile manipulator for single-arm tabletop tasks (Brohan et al., 2023b; Fang et al., 2026a). We count it as an independent MolmoAct2 split rather than inside a bundled source paragraph, and we retain 365.7 hours after verification.

LET Base / LejuRobot. The LET dataset adds full-size humanoid manipulation teleoperated on Kuavo platforms across industrial, home, and service scenes (LejuRobotics, 2025). Its humanoid embodiment complements the arm-centric sources, and we use 318.0 hours from the `let_base_dataset` production split after verification.

RoboCOIN. RoboCOIN is a multi-embodiment bimanual dataset of more than 180 thousand teleoperated demonstrations across 421 tasks and 15 robot platforms, with hierarchical trajectory, segment, and frame annotations (Wu et al., 2026). Its broad bimanual coverage across diverse platforms provides rich cross-embodiment supervision for dexterous two-arm manipulation, and we use 315.9 hours after verification.

AgiBot World. AgiBot World contributes more than 1 million trajectories across 217 tasks in five domains, namely domestic, retail, industrial, restaurant, and office, collected on mobile bimanual humanoids with multi-view RGB, depth, and language annotations for the overall task and each sub-step (Bu et al., 2025). Its standardized pipeline with human-in-the-loop verification yields high-quality long-horizon data, and we use 278.7 hours from the AgiBotWorld2026 release after verification.

DROID. DROID is an in-the-wild single-arm dataset of 76 thousand trajectories and about 350 published hours, collected on Franka Panda robots across 564 scenes and 86 tasks in 52 buildings, with three synchronized RGB views, depth, calibration, and language instructions per episode (Khazatsky et al., 2024). Its scene and task diversity makes it a strong source of real gripper-object manipulation under varied viewpoints, and we retain 255.6 hours after verification.

MolmoAct2 SO100/101. This processed source split covers the SO100/101 robot trajectories released with MolmoAct2 (Fang et al., 2026a). It is counted independently from the other MolmoAct2 robot splits, and we retain 180.3 hours after verification.

MolmoAct2 BC-Z. This processed source split uses language- and video-conditioned BC-Z manipulation trajectories collected by interactive imitation across more than 100 tasks (Jang et al., 2022; Fang et al., 2026a). It is counted as its own MolmoAct2 split, and we retain 148.9 hours after verification.

BridgeData V2. BridgeData V2 contributes 60,096 teleoperated trajectories on a low-cost WidowX arm across 24 environments and 13 skills, conditioned on language instructions or goal images and recorded with fixed and wrist RGB-D views (Walke et al., 2023). We include it because its broad task and environment variation transfers across labs, and we use 109.1 hours from 47,651 annotated trajectories after verification.

MolmoAct2 Bridge. This processed source split uses BridgeData V2 trajectories through the MolmoAct2 release (Walke et al., 2023; Fang et al., 2026a). It is counted separately from the base BridgeData V2 entry above, and we retain 105.1 hours after verification.

MolmoAct2 DROID. This processed source split uses DROID trajectories through the MolmoAct2 release (Khazatsky et al., 2024; Fang et al., 2026a). It is counted once under Real-Robot Data, separately from the base DROID entry above, and we retain 50.8 hours after verification.

HumanoidEveryday. HumanoidEveryday contributes humanoid manipulation trajectories with RGB, depth, LiDAR, tactile streams, and language annotations (Zhao et al., 2025b). Its open-world humanoid coverage adds an embodiment and sensing configuration not represented by the arm-only sources, and we use 31.8 hours after verification.

CyberOrigin. CyberOrigin is a self-collected real-robot production split used for deployment-aligned manipulation supervision. The current project bibliography has no verified external reference for this source, so we make no broader collection-scale claim and report only the 30.3 hours retained after verification.

RH20T robot. RH20T contains contact-rich robot sequences spanning many tasks and skill categories on multiple embodiments, with synchronized vision, force and torque, audio, and proprioception (Fang et al., 2024). We count its robot stream as an independent processed source split and retain 596.4 hours after verification.

MolmoAct2 Tabletop. This processed UMI-style split covers tabletop manipulation trajectories from MolmoAct2 (Fang et al., 2026a). We count it separately from the robot MolmoAct2 splits and retain 0.5 hours after verification.

MolmoAct2 Household. This processed UMI-style split covers household manipulation trajectories from MolmoAct2 (Fang et al., 2026a). We count it separately from the robot MolmoAct2 splits and retain 0.6 hours after verification.

B.2 UMI DATA

UMI (self-collected). UMI data is gathered with handheld universal manipulation grippers that capture gripper-object contact from a wrist-mounted view (Chi et al., 2024). We retain 17,505.8 hours of UMI clips after verification.

B.3 SIMULATION DATA

BEHAVIOR-1K. BEHAVIOR-1K defines 1,000 everyday household activities grounded in 50 scenes with more than 5,000 annotated objects, simulated in the OmniGibson engine with physics for rigid bodies, deformable bodies, and liquids (Li et al., 2022). We use it to add scripted and precisely labeled state transitions that balance the noisier in-the-wild sources, sampling 1,102.4 hours into the corpus.

InternData-A1. InternData-A1 is a hybrid synthetic-real manipulation dataset spanning four embodiments (InternRobotics, 2025). Its pick-and-place, articulated, and long-horizon tasks add structured manipulation trajectories at scale, and we use 197.8 hours after verification.

SynData. SynData provides multimodal manipulation clips with egocentric RGB-D, hand joint states, and fingertip keypoints (PsiBotAI, 2025). Following the production taxonomy used throughout this report, we count this processed split under Simulation Data and retain 17.2 hours after verification.

B.4 EGOCENTRIC HUMAN DATA

Ego4D. Ego4D is a massive egocentric corpus of roughly 3,670 published hours of unscripted daily-life video recorded by more than 900 camera wearers across 74 locations in 9 countries (Grauman et al., 2022). We draw on its head-mounted hand-object manipulation segments, which expose dense everyday object-state changes under natural first-person motion, and retain 3,670.0 hours after verification.

AI Data curated. The `ai_data_by_keyword_2.9m` curated library contains web-sourced RGB clips that have passed an initial in-house screening. It broadens visual and scene coverage but carries no single external reference, so we treat it as weak contextual supervision and retain 3,584.5 hours after verification.

AI Data raw. The `aidata_raw` library is the separate unfiltered web-video pool from the same internal pipeline. It remains distinct from the curated split because the two rows undergo different source-level screening, and we retain 1,918.2 hours after verification.

Egocentric-10K. Egocentric-10K is a large release of about 10,000 published hours of first-person factory video from over 2,000 workers, with unusually high rates of visible hands and active manipulation (Build AI, 2025). It adds industrial hand-object work that household sources rarely cover, and we use 1,300.0 hours after verification.

EgoVerse. EgoVerse is imported as a dedicated egocentric production split and contributes long-form first-person activity video to the open-world human-data pool. We report only its role in the pipeline and the 1,132.8 hours retained after verification.

EgoDex. EgoDex is a large-scale dexterous manipulation dataset of 829 published hours of 30 Hz egocentric video with 338 thousand demonstrations over 194 tabletop tasks, collected with Apple Vision Pro and carrying paired 3D hand and finger tracking with upper-body pose (Hoque et al., 2025). We include it because its precise per-frame hand pose gives a strong bridge from human demonstration to gripper control, retaining 829.0 hours after quality verification.

Xperience. Xperience is a head-mounted first-person video source retained as a distinct production split. We avoid attributing unverified release-scale details and report the 800.0 hours retained after verification.

Something-Something V2. Something-Something V2 provides 220,847 short clips of templated object manipulations (Goyal et al., 2017). Its label design forces a focus on temporal dynamics rather than static appearance, which suits short state-transition supervision, and we retain 240.0 hours after verification.

Ego-Exo4D. Ego-Exo4D provides about 1,286 published hours of time-synchronized first- and third-person video of skilled activities across cooking, repair, sports, and music, recorded by 740 participants across 13 cities (Grauman et al., 2024). Its paired ego and exo views give cross-view supervision for the same action, and we use 155.1 hours after filtering.

EPIC-KITCHENS-100. EPIC-KITCHENS-100 records 100 published hours of unscripted kitchen activity in 45 home kitchens, with roughly 90 thousand densely segmented action instances labeled with verbs and nouns (Damen et al., 2022). Its dense cooking, cleaning, and storage interactions add a fine-grained action vocabulary, and we retain 91.2 hours after verification.

EgoMe. EgoMe contributes 7,902 paired exocentric and egocentric videos, about 82 published hours, that record a person first observing and then imitating a demonstrated action (Qiu et al., 2025). The observe-then-imitate structure links third-person intent to first-person execution, and we use 43.9 hours after verification.

CaptainCook4D. CaptainCook4D adds egocentric cooking recordings with both correct executions and deliberately induced errors (Peddi et al., 2024). The error cases are useful for modeling deviations from a plan rather than only successful trajectories, and we use 39.4 hours after verification.

HD-EPIC. HD-EPIC adds 41 published hours of kitchen video in 9 kitchens with highly detailed annotations grounded in 3D, including recipe steps, fine-grained actions, audio events, and object movements (Perrett et al., 2025). Its dense labels give clean action and object references for state-transition supervision, and we retain 38.1 hours after verification.

HoloAssist. HoloAssist records 166 published hours of two-person assistive manipulation across 20 object-centric tasks, captured in mixed reality with seven synchronized streams including RGB, depth, hand pose, and eye gaze (Wang et al., 2023). Its mistakes and corrective interventions enrich the action distribution beyond clean demonstrations, and we use 32.2 hours after verification.

HOT3D. HOT3D offers egocentric multi-view recordings captured with Project Aria and Quest 3 and supplies 3D hand and object tracking (Banerjee et al., 2025). Its tracked hand-object interactions add contact-rich state changes, and we use 5.5 hours after verification.

TACO. TACO is a bimanual tool-action-object dataset pairing third-person and egocentric views with 3D hand and object meshes and action labels (Liu et al., 2024). Tool use broadens contact types and object functions beyond bare-hand grasping, and we use 3.0 hours after verification.

ARCTIC. ARCTIC records dexterous bimanual manipulation of articulated objects from synchronized third-person and egocentric views, with 3D hand and object meshes and dense dynamic contact (Fan et al., 2023). It is a clean source for contact-rich state changes such as opening and closing, and we retain 2.3 hours after verification.

HOI4D. HOI4D provides 2.4 million RGB-D egocentric frames over 4,000 sequences, with panoptic and motion segmentation, 3D hand pose, category-level object pose, and hand action (Liu et al., 2022). Its category-level coverage gives structured object-state supervision that raw video lacks, and we use 0.9 hours after verification.

HO-Cap. HO-Cap captures uni- and bimanual hand-object interactions with synchronized RGB-D and per-frame 3D hand and object pose (Wang et al., 2025a). Its pose-rich interactions complement the larger but weakly labeled video sources, and we use 0.6 hours after verification.

TASTE-Rob. TASTE-Rob provides 100,856 egocentric hand-object interaction videos, each aligned with a language instruction and recorded from a fixed viewpoint for interaction clarity (Zhao et al., 2025a). Its instruction-aligned clips suit clean language-to-interaction supervision for our planning formats, and we use 2.6 hours after verification.

RH20T human. RH20T pairs its robot sequences with human demonstration video for the corresponding tasks (Fang et al., 2024). We process that human-view stream separately from the robot stream and retain 98.4 hours after verification.

OakInk2. OakInk2 organizes 627 sequences of bimanual hand-object manipulation into three levels, namely affordance, primitive task, and complex task (Zhan et al., 2024). Its multi-view images with body, hand, and object pose annotations supply hierarchical task structure for long-horizon supervision, and we use 74.6 hours after verification.

C Additional Data Construction Details

C.1 Temporal Segment Annotation Method Details

This appendix expands the three-step Temporal Segment Annotation pipeline of Section A.1 with the full notation, derivations, and algorithmic detail that the main text compresses for space. The step order below follows the pipeline of Figure 5: a problem setup that fixes the segment representation, a content-adaptive keyframe sampler, a single global parse that proposes candidate steps, and a bidirectional boundary refinement that turns each coarse step into precise visual anchors. Throughout, a single multimodal large language model, written Φ, makes every semantic decision, so the module is one annotator queried with different prompts rather than a stack of specialized detectors.

Problem setup and notation. A raw video is an ordered sequence of frames $V = (f_1, \ldots, f_N)$ recorded at ν frames per second, so frame f_j carries the timestamp $\tau_j = j/\nu$ and the clip has duration $T = N/\nu$. A **keyframe** is a frame index retained for inspection, and a **candidate step** is an action phrase proposed for the whole clip. We define the annotation target from the bottom up, fixing the smallest unit first and then nesting it into the segment representation and the full per-video sequence.

- **Atomic planning step (smallest unit).** Following the annotation prompt, an **atomic planning step** is one individual execution of a single manipulation whose effect is a clearly visible change of object state. Grounding the unit on a visible state change keeps each atomic planning step aligned with the state transition that RxBrain learns to predict, so it serves as one supervision unit for the world model.
- **Segment (one localized atomic planning step).** A **segment** pins one atomic planning step to the two frames that expose its state change and attaches its supervision, giving the tuple $S_i = \big(I_i^{\text{start}},\, I_i^{\text{end}},\, a_i,\, d_i,\, m_i\big)$ of Equation 5, whose components are fixed as follows.
  - I_i^{start} and I_i^{end} are the start and end anchors, namely the two frames of V that bound the visible state change.
  - a_i is the short step name of the atomic planning step.
  - d_i is the detailed textual description of that step.
  - m_i is metadata recording the dataset identifier, the camera viewpoint, the action index, and the temporal location.
- **Annotation of a video (ordered sequence).** The annotation of V is the temporally ordered sequence of segments $\{S_i\}_{i=1}^{M}$. Repeated executions of the same manipulation appear as separate segments, and neighboring segments may overlap after independent boundary refinement while retaining their chronological indices.

Each S_i therefore corresponds to exactly one atomic planning step. Quality Verification consumes S_i, may correct its detailed description d_i, and passes retained segments to Segment Structuring (Section A.3). The rest of this appendix explains how this sequence is produced from weak video alone, by first proposing candidate steps over the whole clip and then refining each one to its anchor frames.

Content-adaptive keyframe sampling. A long clip holds far more frames than the annotator can read at once, and uniform sampling wastes this budget on static intervals while still missing brief manipulations, so we place keyframes where the visual content actually changes. We first decode the clip at a reduced rate $\nu_s = \min(4, \nu)$, downscale each decoded frame to 192×108 and convert it to the LUV color space through an operator ψ, then measure the motion between consecutive decoded frames $f_{j_{k-1}}, f_{j_k}$ as the mean absolute LUV difference

$$\delta_k = \frac{1}{|\Omega|} \sum_{u \in \Omega} \big| \psi(f_{j_k})[u] - \psi(f_{j_{k-1}})[u] \big|, \tag{6}$$

where Ω indexes the downscaled pixel-channel grid. We smooth δ with a moving average of odd width near 0.5ν to suppress per-frame flicker, and we mark a frame as an anchor when its smoothed score $\bar{\delta}$ is a local maximum that exceeds the adaptive threshold

$$\theta_\delta = \max\Big(Q_{p_{\text{lo}}}(\bar{\delta}),\; Q_{0.5}(\bar{\delta}) + \beta \left[Q_{p_{\text{hi}}}(\bar{\delta}) - Q_{0.5}(\bar{\delta}) \right] \Big), \tag{7}$$

in which Q_p denotes the p-quantile of the smoothed scores, p_{lo} a low-quantile floor, p_{hi} a high quantile, and β a margin coefficient. This makes θ_δ a per-clip robust statistic with the median $Q_{0.5}$ as a baseline, the gap between $Q_{p_{\mathrm{hi}}}$ and the median as an adaptive dispersion margin scaled by β, and the floor $Q_{p_{\mathrm{lo}}}$ guarding low-activity clips against noise-driven fragmentation. We then prune anchors greedily under a minimum spacing $\Delta_{\min}$ keeping higher $\bar{\delta}$ first, always insert the first and last frames, fill any temporal gap wider than $g_{\max}$ by interpolation so long static stretches stay represented, and cap the anchor set $\mathcal{K}$ at $A_{\max}$ frames. For the global parse we uniformly subsample $\mathcal{K}$ to n keyframes $\mathcal{G} \subseteq \mathcal{K}$, bounding the model context while preserving chronological coverage of the whole clip. In practice we set $p_{\mathrm{lo}} = 0.8$, $p_{\mathrm{hi}} = 0.9$, $\beta = 0.2$, $\Delta_{\min} = 0.2\,\mathrm{s}$, $g_{\max} = 0.5\,\mathrm{s}$, $A_{\max} = 50$, and $n = 30$, empirical constants applied uniformly to every source rather than values from an optimality criterion.

MLLM global action parsing. With the keyframe set $\mathcal{G}$ fixed, the annotator receives the n global keyframes in chronological order, numbered 0 to $n-1$, and returns in one pass a one-sentence summary, a verb-object goal, the camera viewpoint, a binary annotatability flag, and an ordered list of candidate steps. Each candidate step $c_i = (a_i, d_i, p_i, q_i)$ contains a short step name a_i, a detailed description d_i, and start and end indices p_i and q_i in $\mathcal{G}$, with $0 \le p_i \le q_i \le n-1$. The prompt holds each candidate to the atomic-planning-step definition above and enforces $q_i < p_{i+1}$, so the coarse index ranges are ordered and non-overlapping. This constraint applies only to the coarse candidates and does not prohibit overlap between the final independently refined segments. When the annotatability flag is false, for instance when the manipulated object is too small or the scene is too cluttered or dark to track a state change, the model returns an empty step list and the clip yields no segments. Otherwise each c_i is lifted to a coarse segment $\tilde{S}_i$ by mapping its keyframe indices back to the frame identifiers and timestamps carried by $\mathcal{G}$.

Local candidate-frame boundary refinement. A coarse segment $\tilde{S}_i$ inherits the timestamps of sparse keyframes, so its boundaries are biased away from the true transition, and we make this bias precise for the action onset. Let $j^\star$ be the true onset, namely the first frame at which the manipulated object has visibly changed. Because the global parse may place a boundary only on a sampled keyframe in $\mathcal{G}$, and because the annotator can report a change only once the new state is visible, the coarse onset is the first sampled keyframe at or after $j^\star$,

$$\hat{j}_0 = \min\{\, j \in \mathcal{G} \,:\, j \ge j^\star \,\} \;\ge\; j^\star, \tag{8}$$

so $\hat{j}_0$ is an upward-biased estimate of $j^\star$ whose expected error scales with the keyframe spacing rather than with the frame period. Conditioning on history alone cannot remove this bias, since a labeler that reads only frames up to a candidate boundary keeps the previously observed state until the change is unambiguous, which can only push the estimate later. The remedy is to bracket each boundary with a dense window of frames that straddle it and to relocalize it among these candidates while the opposite anchor is shown in the same prompt for reference, which replaces the keyframe spacing in Equation 8 by the much finer candidate spacing and pulls the estimate back toward $j^\star$. The offset is treated symmetrically, so refinement recovers a tight anchor pair around the genuine state change instead of the inflated coarse window.

Algorithm 1 states the procedure for one segment, with the same annotator Φ invoked under an end-selection prompt Φ_{end} and a start-selection prompt Φ_{start}. Both prompts receive the short step name a_i and the detailed description d_i. We refine the end before the start, because the action is easiest to recognize once its completed state is fixed, and the recovered end then anchors the search for the start over the $n_c = 8$ candidates of each window straddling the coarse boundary. The two selections are sequential within one segment, while different coarse segments may be refined in parallel.

Two checks keep each emitted segment well formed. The order check in Algorithm 1 places I_i^{start} before I_i^{end}, while the degenerate-expansion branch prevents the anchor pair from collapsing onto one frame. These checks act within each segment and do not impose cross-segment non-overlap. The output retains the chronological candidate order, but adjacent segments may overlap after independent local refinement. Because each step retains exactly one start anchor and one end anchor, the module emits one state-transition pair per step rather than a dense per-frame label map. The two anchors are rendered from the source video at native resolution so downstream training does not inherit the reduced resolution used during annotation.

Algorithm 1 Bidirectional boundary refinement for one segment $\tilde{S}_i$

Require: coarse bounds τ^s, τ^e; short step name a_i; detailed description d_i; video V (T s, ν fps, N frames); candidate count n_c; annotator Φ; next-segment start τ^s_{i+1} ($= T$ if none)

Ensure: anchor pair $(I_i^{\text{start}}, I_i^{\text{end}})$ with $I_i^{\text{start}} < I_i^{\text{end}}$

$\Delta \leftarrow \max(0.3,\ \tau^e - \tau^s)$

sample n_c frames E from $[\tau^s{+}0.3\Delta,\ \min(\tau^e{+}0.5\Delta,\ T)]$; $I_i^{\text{end}} \leftarrow \Phi_{\text{end}}(\tau^s,\ E,\ a_i,\ d_i)$ ▷ look-ahead past coarse end

sample n_c frames B from $[\max(0,\ \tau^s{-}0.5\Delta),\ \tau^s{+}0.7\Delta]$; $I_i^{\text{start}} \leftarrow \Phi_{\text{start}}(I_i^{\text{end}},\ \text{reverse}(B),\ a_i,\ d_i)$ ▷ look-back before coarse start

if $I_i^{\text{start}} \geq I_i^{\text{end}}$ **then**

  swap I_i^{start} and I_i^{end} ▷ enforce temporal order

end if

if $I_i^{\text{start}} = I_i^{\text{end}}$ **then** ▷ degenerate: extend end to avoid zero-length segment

  $I_i^{\text{end}} \leftarrow \min(\lfloor \tau^s_{i+1} \cdot \nu \rfloor{-}1,\ N{-}1)$ if later segment exists, else $\min(I_i^{\text{start}}{+}\text{round}(\nu),\ N{-}1)$

end if

return $(I_i^{\text{start}},\ I_i^{\text{end}})$

Table 6: **Review items in Quality Verification.** Each row pairs a check item with the failure or reject reason it captures, grouped into semantic quality, visual grounding, and consistency between the text and visual transition.

Check item	Failure or reject reason
Semantic quality	
Single atomic planning step with clean boundaries	*Clip mixes steps, has unclear boundaries, or runs too long to bound the transition*
Manipulated target clearly visible	*Target is too small, occluded, poorly lit, or lost in a cluttered background*
Frame free of overlays and edits	*Subtitle, watermark, logo, or an editing cut contaminates a frame*
Detailed description matches the anchors	*d_i disagrees with the frames or is too weak to verify against them*
Genuine action with a unique end state	*No physical action occurs, or the end state is ambiguous*
Both anchors readable	*A start or end anchor is missing or cannot be read*
Visual grounding	
Hand presence in each anchor	*Hand or gripper visibility that the consistency check compares against the text*
Robot arm presence	*Robot arm or gripper visibility recorded for downstream filtering*
Object-exit status	*Unintended loss of the target rather than an intended exit at completion*
Consistency	
Both frames valid and same scene	*A frame is blank or single-color, or the two frames share no common scene*
Stable camera viewpoint	*Camera viewing direction shifts between the two anchors*
Upright camera orientation	*Camera orientation is inverted or gravity-defying*
Description consistent with the scene	*d_i contradicts the object identity, count, position, direction, or state change*
Hand changes reflected in text	*Hand or gripper visibility changes but d_i omits it*

C.2 Quality Verification Review Items

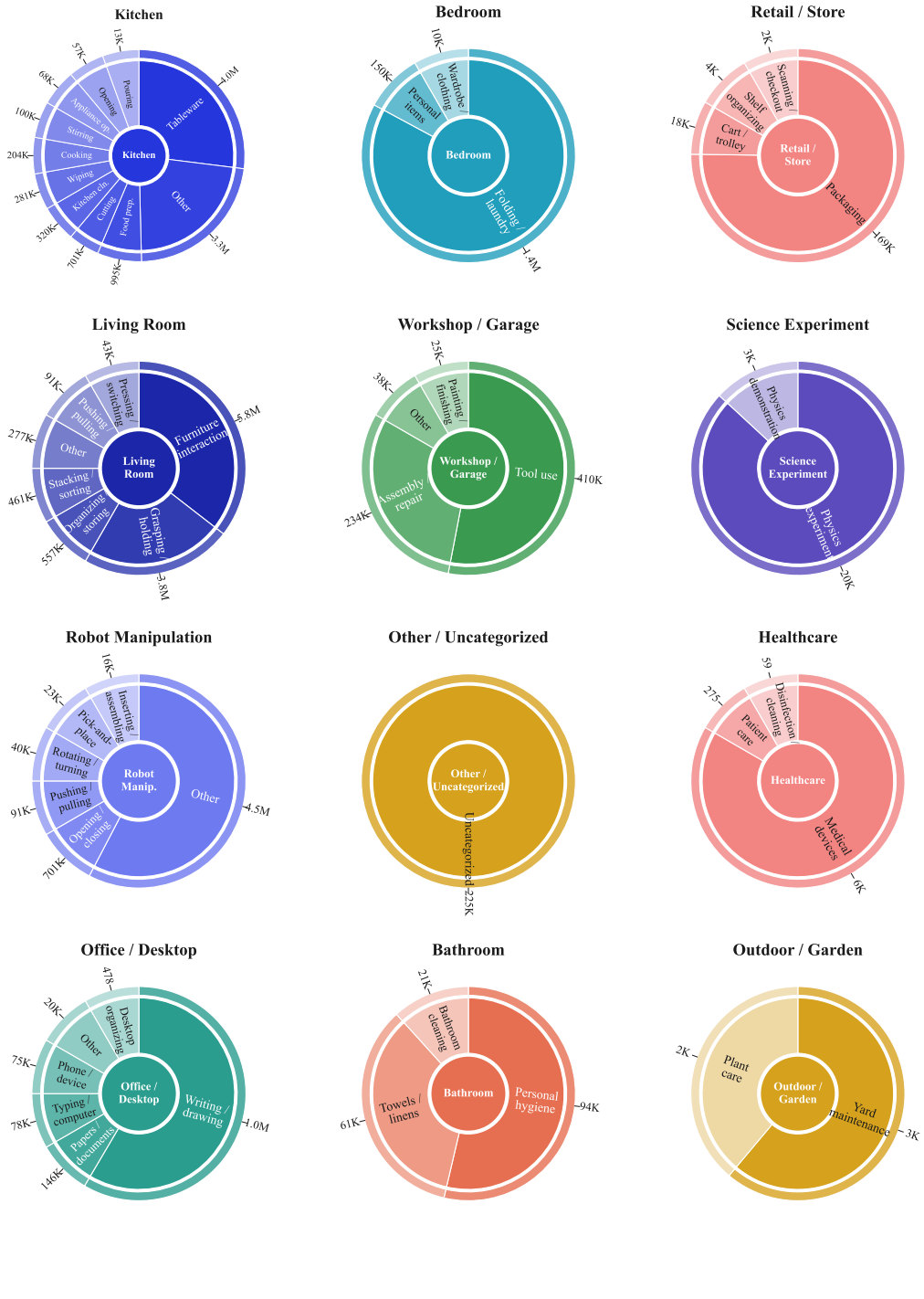
Table 6 expands the Quality Verification summary in Section A.2 into the review items used to inspect each pre-verification segment S_i. Semantic quality checks whether one observable state transition is localized and described correctly by d_i. Visual grounding records whether hands or grippers, robot arms, and manipulated objects are available for filtering, while consistency checks frame validity, viewpoint stability, and agreement between d_i and the visual transition.

Quality Verification consumes S_i and retains segments with reliable anchors and valid state changes. When the visual transition is valid but d_i is inaccurate or incomplete, the verifier corrects d_i instead

of discarding the segment. Segment Structuring consumes only the retained segments after this verification step. Pairs that are invalid, ambiguous, corrupted, or inconsistent are rejected.

D PER-SCENE TAXONOMY STATISTICS

The following figures show the per-scene segment distribution for each of the fourteen scene categories in the pretraining corpus. Each chart displays the action-subtype breakdown within that scene, with wedge area proportional to the number of verified trainable segments.



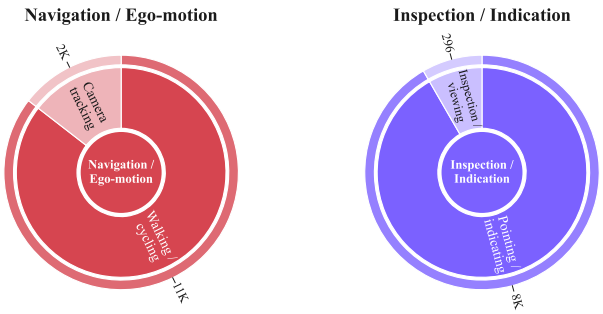


Figure 7: **Per-scene taxonomy sunburst charts (14 scene categories).** Each panel is one self-contained chart for a single scene category, flowing left-to-right and top-to-bottom across three columns (and across pages as needed). The colored center disc names the scene category; the ring around it is divided into wedges, one per action subtype observed in that scene, with wedge angle proportional to the subtype's share of verified trainable segments (very small subtypes are padded to a minimum wedge width so their labels stay legible); the thin outer ring repeats the same wedges and annotates each with its exact segment count. Reading any single chart therefore gives both the relative composition and the absolute segment count of every action subtype within that scene category.

E Training Configuration and Data Mixture

Table 8 summarizes the optimization and sequence-packing configuration for the two training stages: large-scale pretraining (Stage 1) and supervised fine-tuning (Stage 2). Stage 1 initializes all modules of HY-Embodied-0.5 as trainable and emphasizes the flow-matching generation loss (λ_{CE}=0.25, λ_{FM}=1.0); Stage 2 resumes from the Stage 1 final checkpoint and balances the two losses (λ_{CE}=λ_{FM}=1.0), with a lower peak learning rate, a longer packed sequence, and a smaller per-device batch. Both stages use AdamW (weight decay 0.01), a cosine schedule, bf16 precision with ZeRO-2, a 16×-downsampling VAE at $\leq$ 256 px generation, flow-matching timestep shift 1.0, and 0.1 classifier-free-guidance text-condition dropout.

Data mixture (Stage 1). The Stage 1 corpus is drawn from two buckets mixed at the integer ratio 6:4 (Σ=10), which sets each bucket's expected share of optimization steps (Table 7). In contrast to Stage 2, the Stage 1 mixture is deliberately generation-heavy so as to bootstrap the flow-matching path: web-scale text-to-image pairs (b0; 202.5M pairs from LAION (Schuhmann et al., 2022) and COYO (Byeon et al., 2022)) are the larger source at 60.0%, while a single large-scale vision–language understanding bucket (b1; 113.4M examples—the LLaVA-OneVision-1.5 (An et al., 2025) mid-training corpus together with the MAmmoTH VL training data (Guo et al., 2025a)) supplies the remaining 40.0% to co-train the understanding path from the outset. The text-to-image bucket is sampled length-source.

Table 7: **Stage 1 data mixture.** Two buckets mixed at integer ratio 6:4 (Σ=10); the per-step share is each bucket's ratio divided by 10.

Bucket	Content	Scale	Ratio	Per-step share
b0	Text-to-image (LAION + COYO)	202.5M	6	60.0%
b1	Understanding (LLaVA-OV-1.5 + MAmmoTH)	113.4M	4	40.0%

Table 8: **Training hyperparameters for the two stages.** Stage 1 is large-scale pretraining from HY-Embodied-0.5; Stage 2 is supervised fine-tuning resumed from the Stage 1 checkpoint. Global batch = per-device batch × gradient accumulation × GPUs.

Hyperparameter	Stage 1 (Pretraining)	Stage 2 (SFT)
Loss weights ($\lambda_{\mathrm{CE}}, \lambda_{\mathrm{FM}}$)	0.25 / 1.0	1.0 / 1.0
Optimizer	AdamW (weight decay 0.01)	AdamW (weight decay 0.01)
Peak learning rate	2×10^{-4}	6.5×10^{-5}
LR schedule	cosine, 800 warmup	cosine, 1050 warmup
Per-device batch	20	10
Gradient accumulation	1	1
GPUs (nodes × 8)	368 (46 × 8)	312 (39 × 8)
Global batch	7,360	3,120
Precision / parallelism	bf16 / ZeRO-2	bf16 / ZeRO-2
Epochs	1	1
Max sequence length	8,192	9,216
Understanding pixel budget	1,048,576	524,288
Generation resolution	≤ 256 px (VAE downsample 16)	≤ 256 px
Timestep shift (flow)	1.0	1.0
CFG text-cond. dropout	0.1	0.1

Data mixture (Stage 2). The Stage 2 corpus comprises five buckets sampled at the integer ratio 100:12:150:70:90 ($\Sigma = 422$), which determines each bucket's expected share of optimization steps (Table 9). General and embodied understanding (b2) receives the largest individual allocation at 35.5%. The two embodied generative buckets—multi-frame world-model prediction (b3) and interleaved planning (b4)—together account for 37.9%, while text-to-image data (b0, b1) is included to preserve general-purpose image-generation capabilities.

Table 9: **Stage 2 data mixture.** Per-step shares follow the sampling ratio 100:12:150:70:90 ($\Sigma = 422$) and are rounded to one decimal place.

Bucket	Content	Scale	Ratio	Per-step share
b0	T2I base (BLIP3o (Chen et al., 2025), cascade-filtered)	12.0M	100	23.7%
b1	T2I instruct (BLIP3o-60k (Chen et al., 2025) + Self-Build T2I dataset-690k)	0.75M	12	2.8%
b2	Understanding / MMU (spatial CoT + knowledge + pointing)	20.8M	150	35.5%
b3	Embodied multi-frame world model	8.5M	70	16.6%
b4	Embodied interleaved planning	10.3M	90	21.3%

F Details of RxBrain-Bench-JointPlan

Task and evaluation set. Given 1–3 initial observation frames and a task instruction, JointPlan evaluates iterative subtask-text generation, completion-state image prediction, and progress assessment. The held-out set contains 4,756 trajectories from 11 datasets: `jaka`, `egodex`, `xtrainer`, `tasterob`, `taco`, `umi`, `egodex_full`, `arctic`, `bridgev2`, `let`, and `let_base`. Planning depths of 1–5 steps contain 1,536, 749, 366, 1,366, and 739 trajectories, respectively. Cross-model comparisons use an identical 540-record subset, stratified by dataset and planning depth with seed 0.

Evaluation protocol. Evaluation is free-running: generated text and images condition subsequent predictions, exposing long-horizon error accumulation. GPT-5.5 (OpenAI, 2026) judges each trajectory from the task, initial observations, and reference and generated subtask texts and goal images. We average two independent multi-image calls returning structured JSON scores in $[0, 1]$; images are resized to a longest side of 512 pixels before judging. The five MLLM-judged criteria and DINO-based Image Similarity (IS) are defined in Table 10. DINO cosine similarity is computed locally for each generated–reference image pair before judging. Step-level criteria are averaged over the

trajectory, yielding

$$S_{\text{plan}} = 0.10\,\text{OU} + 0.25\,\text{SP} + 0.25\,\text{GI} + 0.20\,\text{SG} + 0.10\,\text{CC} + 0.10\,\text{IS}. \tag{9}$$

We report the overall score and per-criterion breakdown to distinguish textual-reasoning and visual-prediction failures.

Baselines. Qwen-Agent pairs Qwen3-VL-2B-Instruct for textual reasoning with Qwen-Image-Edit for goal-image generation. Cosmos3-Nano uses the same omni model for chat reasoning and image-to-video generation, taking the final frame of a 17-frame clip as the subgoal image. BAGEL combines its understanding path with native image editing to generate subgoal images directly. Each baseline conditions goal-image generation on the current subtask and preceding generated image, starting from the last initial observation frame. All methods are evaluated on the same subset using the above rubric.

Table 10: **JointPlan evaluation rubric.** The first five criteria are MLLM-judged; IS is computed using DINO. Step-level scores are averaged over each trajectory.

Criterion	Weight	Level	Definition
Observation Understanding (OU)	10%	Trajectory	Correct interpretation of the initial observed state.
Subtask Planning (SP)	25%	Step	Appropriateness of the proposed next action.
Goal-Image Correctness (GI)	25%	Step	Correct depiction of the subtask goal relative to the reference image.
Subtask–Goal Consistency (SG)	20%	Step	Alignment between the generated image and the model's own subtask text.
Chain Completion (CC)	10%	Trajectory	Whether the full subtask sequence would accomplish the overall task.
Image Similarity (IS)	10%	Step	DINO cosine similarity to the reference goal image.

Human validation of the MLLM judge. We validate the MLLM judge against human assessments on the five judged criteria (OU, SP, GI, SG, and CC), excluding the independently computed DINO similarity. The validation includes 42 trajectory-level items for each of OU and CC and 123 step-level items for each of SP, GI, and SG, yielding 453 criterion-level human–judge comparisons. Table 11 reports exact-bin agreement, agreement within ± 0.25, the agreement coefficient α, and mean absolute error (MAE) on the $[0, 1]$ score scale. Exact-bin agreement exceeds 95% across all five criteria, while agreement within ± 0.25 reaches 100% for four criteria and 97.6% for CC. The macro-averaged exact-bin agreement, tolerance agreement, α, and MAE are 97.1%, 99.5%, 0.943, and 0.056, respectively, with equal weight assigned to each criterion. These results demonstrate close human–judge agreement on the validation subset across both trajectory- and step-level evaluation, supporting the use of the MLLM judge for automatic joint-planning assessment.

Table 11: **Human–judge agreement on RxBrain-Bench-JointPlan.** N denotes the number of evaluated criterion-level items. Exact-bin and ± 0.25 agreement are reported as percentages; MAE is measured on the $[0, 1]$ score scale. The final row reports unweighted macro-averages across the five criteria.

Criterion	N	Exact-bin ↑	± 0.25 ↑	α ↑	MAE ↓
OU	42	95.2	100.0	0.895	0.061
SP	123	100.0	100.0	0.981	0.037
GI	123	97.6	100.0	0.958	0.060
SG	123	97.6	100.0	0.929	0.069
CC	42	95.2	97.6	0.951	0.051
Macro-average	–	97.1	99.5	0.943	0.056

G Details of Action Generation

G.1 RxBrain's joint subgoal conditioned policy

To ensure a controlled comparison, we fine-tune three matched $\pi_{0.5}$ policies on the same RMBench training data, 300 episodes for each task, using the same model initialization, random seed, op-

timization schedule, and all other training hyperparameters. The three policies differ only in the planner-conditioning inputs provided during both training and evaluation: (i) no planner-provided subgoal, (ii) a textual subgoal generated by RxBrain, or (iii) a joint textual–visual subgoal generated by RxBrain. All policies receive the same current head, left, and right camera observations, current proprioception state, and the same proprioception state history of two previous action chunks. The policy-only baseline therefore differs from the planning variants only by the absence of planner-generated subgoal conditioning, while the text-only and joint text–image variants differ only in whether the visual goal is additionally provided.

G.2 Action Model based on RxBrain

For action, we extend RxBrain with a modality-specialized branch containing dedicated attention projections $\{Q, K, V, O\}_a$, FFN, and layer norms. The branch is initialized by copying the corresponding parameters from the vision understanding branch. Meanwhile, action tokens remain in the shared global attention operation with observation, instruction, and state tokens, allowing action-specific parameterization while preserving cross-modal conditioning.

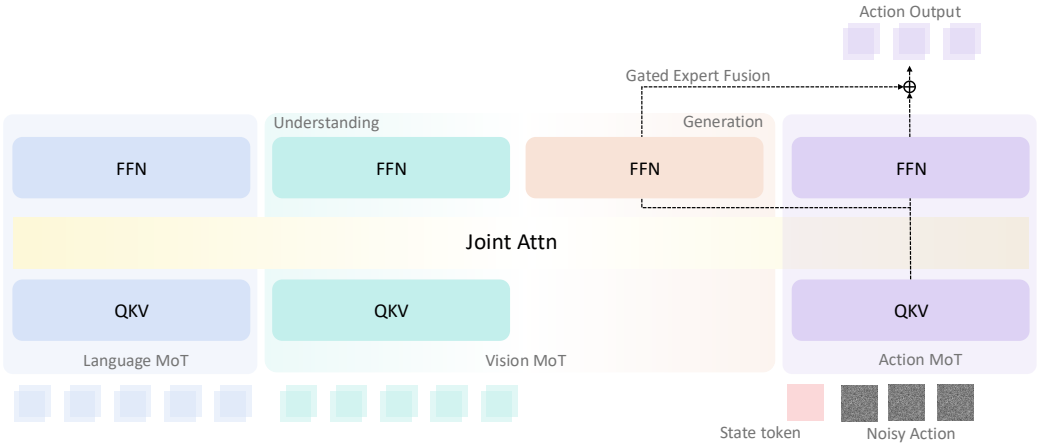


Figure 8: **Action model architecture.** A modality-specialized Action MoT branch uses dedicated attention and feed-forward parameters while sharing global self-attention with language, vision, and state tokens. A zero-initialized channel-wise gate transfers features from the pretrained generation expert to the action expert, whose outputs are decoded into action chunks.

Gated expert fusion In RxBrain, the generation branch is pretrained for general image generation, world-state prediction and joint subgoal planning, making its feed-forward expert FFN_g a natural repository of predictive and planning representations. Since action generation also depends on these capabilities, the action branch should efficiently reuse the pretrained generation expert rather than relearn the same knowledge from scratch.

Let x_a denote the hidden states of the action tokens after joint self-attention. We apply the action and generation experts to the same action-token representation, while retaining the branch-specific normalization of each expert. The fused feed-forward update is defined as

$$\mathrm{FFN}_a^{\mathrm{fused}}(x_a) = \mathrm{FFN}_a\big(\mathrm{LN}_a(x_a)\big) + g \odot \mathrm{FFN}_g\big(\mathrm{LN}_g(x_a)\big), \tag{10}$$

where LN_a and LN_g are the LayerNorm modules associated with the action and generation branches, respectively, and $\odot$ denotes element-wise multiplication. Thus, both experts process the same post-attention action representation, but through their own learned normalization parameters. This preserves the feature scaling expected by each expert when the pretrained generation expert is reused for action tokens.

The fused feed-forward update is incorporated through the standard residual connection,

$$x_a^{\mathrm{out}} = x_a + \mathrm{FFN}_a^{\mathrm{fused}}(x_a). \tag{11}$$

The per-channel gate g is initialized to zero, so the fused module initially reduces exactly to the standalone action expert. This provides a function-preserving starting point without introducing an

immediate contribution from the generation branch. During action training, the gate learns to selectively inject predictive and planning features from FFN_g, while preserving the modality-specific capacity of FFN_a. The learned magnitude of g also provides a direct measure of how strongly each channel relies on the pretrained generation expert.

Action representation and flow-matching head The proprioceptive state (bimanual end-effector pose $[\mathrm{pos}_3, \mathrm{quat}_4, \mathrm{grip}_1] \times 2$, 16-D) is mapped by a lightweight state encoder into a single proprioception token, and the next H steps form an action chunk in which each step is one action token. Actions are generated with flow-matching formulation, applied in the normalized action space: given a clean action chunk a and noise $\epsilon \sim \mathcal{N}(0, I)$, we form the interpolant $a_s = (1-s)\, a + s\, \epsilon$ and train the action head to regress the velocity field,

$$\mathcal{L}_{\text{action}} = \mathbb{E}_{s,a,\epsilon}\big[\, \|v_\phi(a_s, s, c) - (\epsilon - a)\|_2^2 \,\big].$$

Following (Black et al., 2024) and (Physical Intelligence et al., 2025), the action chunk draws a single per-chunk flow time $s \sim \mathrm{Beta}(1.5, 1)$ biased toward high noise. At inference, the action chunk is denoised from pure noise by a few-step Euler ODE along the predicted velocity field and de-normalized into end-effector pose commands.

Inference acceleration for real-robot deployment For asynchronous real-robot deployment, we cache prefix keys and values, decode only action tokens, and reuse modality-routing indices across layers to reduce synchronization overhead. We also use CUDA graph replay to reduce kernel-launch overhead. With two Euler steps per action chunk, these optimizations reduce inference latency from 210 to 143 ms ($1.47\times$ speedup). Open-loop trajectory checks show end-effector deviations from the baseline below 0.65 mm in position and 0.73° in orientation.

H Demonstration of Joint Planning

Figures 9 shows representative joint subgoal planning rollouts on a held-out task: for the same initial observation and goal, RxBrain produces sub-step texts and predicted goal frames that stay aligned with the ground-truth plan, whereas the modular Qwen-Agent tends to repeat a generic instruction and drift visually, and the Cosmos3-Nano agent hallucinates object appearances and scene changes over successive steps.

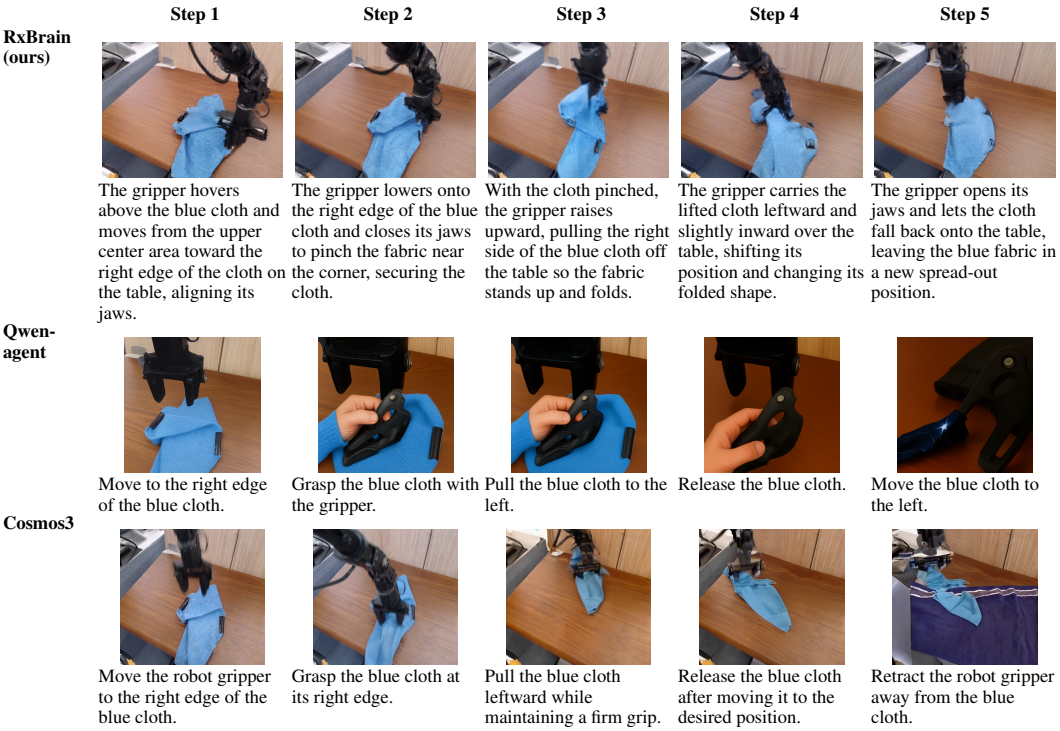


Figure 9: **RxBrain-Bench-JointPlan, task "move to the right edge of the blue cloth, grasp it, pull it leftward, and release it."** RxBrain reproduces the grasp–pull–release sequence and the cloth's changing shape.