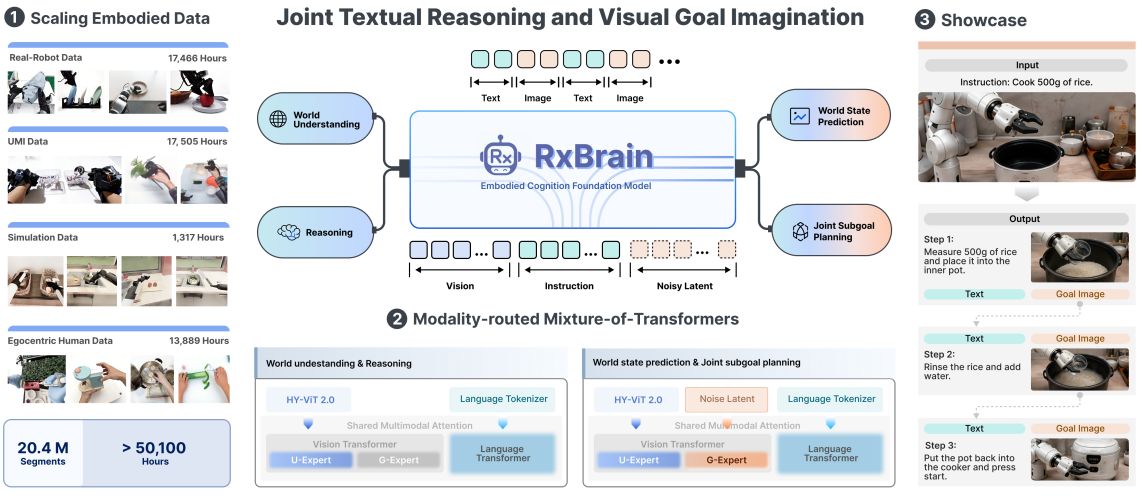
RxBrain: Embodied Cognition Foundation Model with Joint Language-Visual Reasoning and Imagination

Tencent Robotics X Team × Futian Laboratory × Tencent Hy Team

https://huggingface.co/tencent/Hy-Embodied-RxBrain-1.0
https://github.com/Tencent-Hunyuan/Hy-Embodied-RxBrain-1.0

Abstract

Embodied cognition requires agents to connect high-level task reasoning with the physical states to be achieved. We introduce **Hy-Embodied-RxBrain**, an embodied cognition foundation model with joint language-visual reasoning and imagination. Unlike vision-language models that emphasize scene understanding and textual decision making, or generative world models that mainly predict future visual states, RxBrain represents embodied plans in a single planning sequence where language and visual imagination play complementary roles. Language provides the abstract structure of a plan, including task decomposition, planning primitives, constraints, temporal order, and decision logic, while visual imagination grounds this structure through world state prediction and joint subgoal planning, associating each planning step with intermediate and final physical states. RxBrain adopts a unified multimodal Mixture-of-Transformers architecture that supports language, image, and video understanding and generation within one model. To train this capability, we build an automatic pipeline that converts embodied videos into joint text-visual planning supervision by decomposing videos into planning steps and aligning them with visual state transitions. We further introduce RxBrain-Bench to evaluate whether models can represent embodied plans through joint textual and visual components rather than separate understanding or generation. Experiments show that RxBrain maintains embodied understanding and generation abilities, and produces plans with coupled textual reasoning, world state prediction, and joint subgoal planning. We also extend RxBrain to continuous robot action generation, where it shows promising real-robot performance without large-scale action-data pretraining. These results provide an initial step toward foundation models for embodied cognition.



Contents

1 Introduction

Embodied cognition requires an agent to reason not only about what to do, but also about what the world should look like after its actions. Textual reasoning provides a natural interface for abstract task understanding, causal inference, constraint satisfaction, and long-horizon decomposition, enabling an embodied agent to formulate plans beyond immediate perception. Meanwhile, visual goal imagination offers a complementary form of reasoning by representing desired physical states, spatial configurations, and object interactions in the modality where embodied actions ultimately take effect. However, in complex tasks such as planning, neither textual reasoning nor visual imagination alone can fully specify an executable intention: language may describe high-level procedures while omitting critical spatial or state details, whereas visual goals may depict desired outcomes without exposing the underlying causal structure or intermediate decisions required to achieve them. This motivates a joint formulation of embodied cognition, where textual reasoning and visual goal imagination are integrated to represent, reason about, and plan toward goals in the physical world.

Recent advances in embodied AI have largely developed along two complementary directions. Vision-language models (VLMs) (Ji et al., 2025; Team, 2025; Dang et al., 2025; 2026; Lee et al., 2025; Fang et al., 2026) provide a general interface for connecting perception, language, and action: they can interpret visual observations, understand instructions, ground objects and spatial relations, decompose tasks, and support high-level decision-making through textual reasoning. However, their cognitive process is typically expressed in language, leaving the desired physical outcome only implicitly specified and often lacking an explicit imagination of the visual goal state. In parallel, world models and generative models (Guo et al., 2025; Team et al., 2026b; Li et al., 2026; Ye et al., 2026b) focus on predicting future observations, simulating state transitions, or synthesizing goal-conditioned visual outcomes, making them particularly suitable for visual planning. Yet these models are usually less capable of explicit causal reasoning, abstract task understanding, and constraint-aware decision-making. As a result, existing approaches often separate the two sides of embodied cognition: VLMs reason over what should be done, while world models imagine what might happen, but neither fully integrates textual reasoning with visual goal imagination.

More recently, embodied foundation models have begun to move beyond the separation between VLMs and world models. A representative example is Cosmos 3 (Agarwal et al., 2026), which connects understanding, generation, simulation, and action within an omnimodal framework for physical AI. Such models mark an important step toward unified embodied intelligence. However, despite integrating reasoning and generation within a unified framework, they still tend to treat them as separate capabilities rather than organizing them into a coupled cognitive process. This suggests that joint embodied reasoning requires more than architectural unification: the model must be endowed with the ability to coordinate textual reasoning and visual imagination toward the same planning objective. $\pi_{0.7}$ (Intelligence et al., 2026) explores combining a vision-language model with an image generation model to generate joint language-visual subgoals, which are used as conditioning signals for VLA policies to guide action generation. In the broader multimodal domain, models such as BAGEL (Deng et al., 2025) further show that text and images can be generated in a mixed or interleaved form, where language can describe, refine, or guide visual generation. This form of jointness is often centered on representing the same semantic content across modalities. Embodied planning requires a more complementary relation between language and visual, where language specifies actions, constraints, and decisions, while visual imagination specifies the goal and world states those actions should realize.

To address this challenge, we introduce RxBrain, an embodied cognition foundation model with joint language-visual reasoning and imagination for embodied planning. Rather than treating language reasoning and visual generation as separate capabilities, RxBrain organizes them within the same planning sequence: language expresses task decomposition, planning primitives, constraints, temporal order, and decision logic, while visual imagination grounds the plan through world state prediction and joint subgoal planning, depicting the intermediate and final physical states that each planning stage is intended to realize, as shown in Figure 1. This allows RxBrain to represent an embodied plan through complementary textual and visual components, specifying both how a task should progress and what world states should be achieved along the way. Built upon HY-Embodied-0.5(Team et al., 2026a), RxBrain adopts a unified multimodal architecture based on Mixture-of-Transformers (Liang et al., 2024), supporting understanding and generation across language, images, and videos within a single model. Its modality-specific text and vision transformers enable language understanding/generation and visual understanding/generation to be learned in a shared framework. In the vision transformer, visual understanding tokens and generation tokens are processed by separate FFNs while sharing QKV projections, allowing perception and imagination to interact within the same visual tower. On the generation path, RxBrain removes the need for additional VAE-encoded visual input tokens while keeping visual generation in the VAE latent space, simplifying the multimodal input pipeline and encouraging visual understanding and generation to share a more aligned semantic representation.

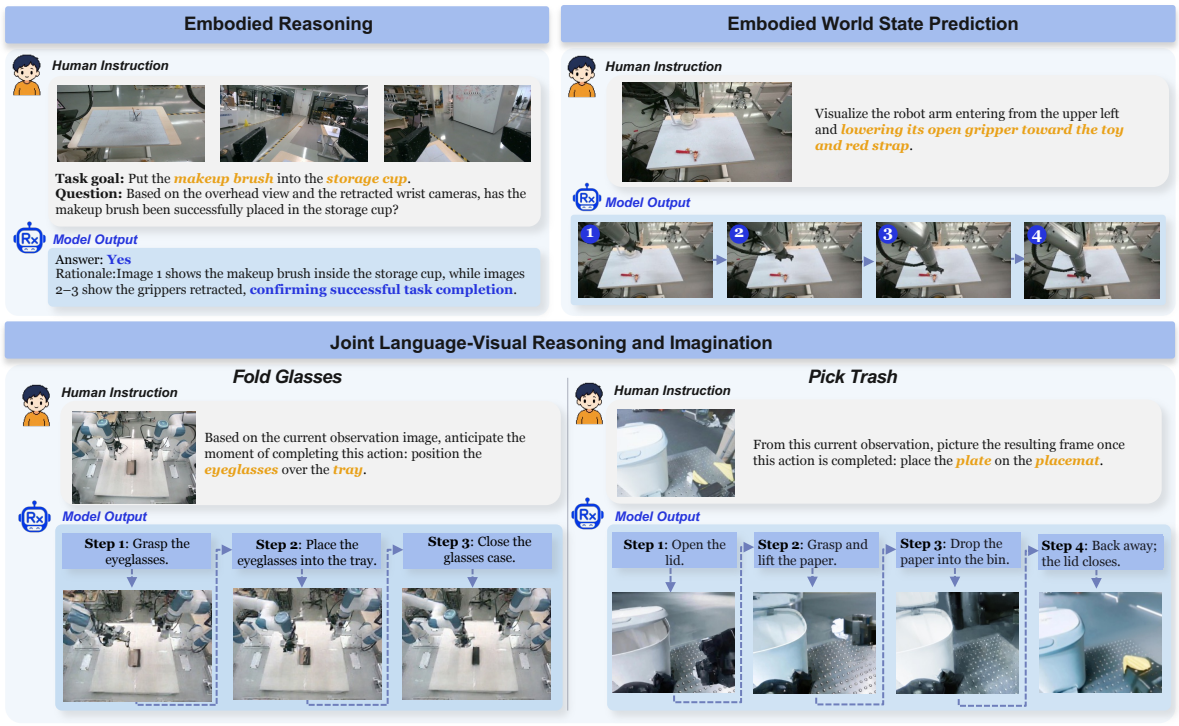


Figure 1: **Capability overview of RxBrain.** RxBrain supports multiple embodied tasks within a unified model, including embodied visual question answering, multi-frame visual generation, and joint textual reasoning with visual goal imagination. Given human instructions and visual observations, the model can reason about the current scene, imagine future or goal states, and generate joint language and visual planning step by step.

Training such a model requires supervision in which language-visual reasoning and imagination are coupled around the same embodied task. However, data that jointly specifies what actions should be taken and what physical states those actions should produce is scarce and expensive to annotate manually. To address this, we propose an automatic data construction pipeline that converts embodied videos into joint text-visual planning supervision. Given an embodied video, the pipeline decomposes it into action-centric segments, associates each action with its corresponding visual state transition, filters out ambiguous or unsuitable sequences, and organizes the remaining data into multi-granularity samples ranging from step-level subgoals to full-task plans. This supervision teaches RxBrain to align textual action descriptions with the visual state transitions they produce, enabling each planning step to be represented by both what action should be taken and what physical change it should achieve. We further introduce RxBrain-Bench, a benchmark for embodied cognition foundation models. Unlike existing evaluations that separately measure embodied understanding, or world states prediction, RxBrain-Bench evaluates whether a model can jointly use language and visual imagination to understand tasks, predict goal states, and construct coherent embodied plans across diverse scenarios.

We evaluate RxBrain on a broad set of embodied and multimodal benchmarks to assess its general image generation embodied understanding and planning capabilities. Beyond these standard evaluations, we use RxBrain-Bench to examine whether a model can represent embodied plans through complementary language and visual components. On RxBrain-Bench, RxBrain produces planning outputs in which language specifies actions, constraints, and decisions, while images depict the goal and intermediate world states associated with them. We further extend RxBrain to continuous robot action generation, where it shows encouraging real-robot performance without relying on large-scale action-data pretraining. These results provide an initial step toward embodied cognition models that reason through coupled language and visual planning representations.

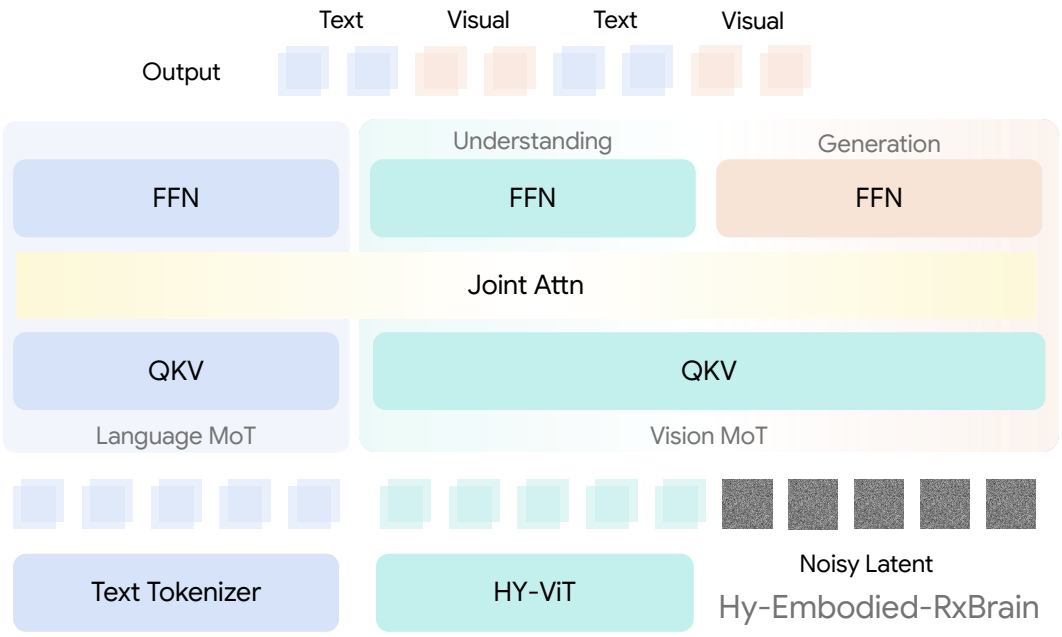


Figure 2: **Architecture overview of RxBrain.** The model adopts a modality-aware MoT architecture to jointly process visual input tokens, text tokens, and generated vision tokens. Visual tokens are handled by a shared vision backbone with full attention, where understanding and generation tokens share attention projections but are routed to separate FFN experts. Text tokens are processed by the Language Transformer with global causal attention for autoregressive generation. Generated vision tokens are decoded through a VAE decoder. Different token colors indicate visual inputs, text inputs/outputs, and generated visual outputs.

2 Model

RxBrain is a unified embodied foundation model designed to support three core capabilities: embodied understanding and reasoning, world state prediction and joint language-visual embodied subgoal planning. To enable these capabilities, RxBrain is built on a modality-aware Mixture-of-Transformers (MoT) architecture. Tokens from different modalities are routed to specialized Transformer pathways and communicate through cross-modal attention, allowing language, understanding visual tokens and generated visual tokens to be modeled within a unified framework. For visual modeling, visual-understanding tokens and visual-generation tokens share a common vision Transformer, including shared attention projection layers, while modality-aware expert routing provides specialized computation for perception and generation. In addition, RxBrain adopts a hybrid attention design: causal attention is used for autoregressive text generation, whereas frame-wise bidirectional attention is applied to visual tokens for both visual understanding and flow-matching-based generation.

2.1 Model Architecture

Visual representations. A key challenge in building a unified model is how to align visual representations for both understanding and generation within a single architecture. RxBrain inherits the Vision Encoder design from HY-Embodied-0.5, where reconstruction ability is introduced during its pre-training (Team et al., 2026a). Specifically, the Vision Encoder is supervised in the VAE latent space, allowing it to preserve fine-grained image details while maintaining strong semantic feature extraction capability. Inspired by this design, RxBrain adopts HY-ViT 2.0 as the single visual encoder to extract both semantic features and VAE-aligned fine-grained features from visual inputs. To improve image generation quality, we employ a pretrained VAE model from FLUX (Black Forest Labs, 2025) as the encoder and decoder for visual generation tokens.

Modality-Aware Mixture-of-Transformers. RxBrain adopts a modality-aware Mixture-of-Transformers (MoT) design. Input tokens are first distinguished by their modality and routed to the corresponding Transformer branch, while a global self-attention enables cross-modal information exchange. Structurally, the backbone exposes two modality-specialized attention branches (language and vision) together with three modality-specific feed-forward experts for text, visual understanding, and visual generation.

The text pathway and the visual-understanding pathway form the two modality-specialized branches for

language and vision processing. For visual modeling, the input vision tokens (used for understanding) and the output vision tokens (used for generation) are processed by the same vision Transformer and share the same attention projection layers; instead of introducing a fully separate Transformer for generation, RxBrain adds a dedicated *visual-generation FFN expert* whose intermediate dimension is widened from 6,144 to 12,288. A modality-conditioned routing then dispatches visual-understanding and visual-generation tokens to their respective experts, so that the two visual functions share visual attention while preserving modality-specific computation for visual input understanding and visual output generation.

In terms of capacity, the base RxBrain model contains **6.2 B** parameters in total. The text and visual-understanding branches are of roughly equal size ($\sim$ 1.5 B each), the widened visual-generation expert contributes about 2.4 B, and the remaining parameters lie in the shared and peripheral modules—the SigLIP vision tower ($\sim$ 0.45 B), the tied input/output token embedding ($\sim$ 0.25 B), and the lightweight image flow-matching head ($\sim$ 7 M); image latents are produced by an external, frozen VAE tokenizer (83.8 M) outside the backbone.

2.2 Multimodal Generation Formulation

The formulation is designed to support three generation modes for embodied tasks: autoregressive text generation, future multi frames visual prediction, and interleaved text-vision generation for embodied planning.

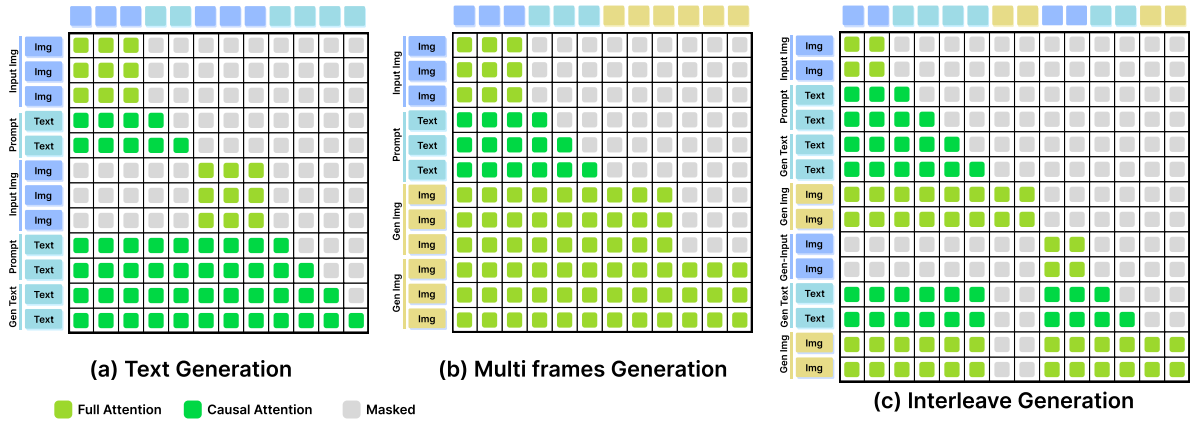


Figure 3: **Hybrid attention patterns in RxBrain.** RxBrain adopts hybrid attention masks for unified multimodal understanding and generation. For text generation (a), input images are encoded with bidirectional visual attention, while text tokens follow causal attention for autoregressive decoding. For multi frames generation (b), input images and text prompts follow the same pattern, while generated frames use bidirectional attention within each frame and causal attention across frames. For interleaved generation (c), text is generated autoregressively before image generation; generated images are then re-encoded as visual input for next steps.

Visual Prefix Representation. RxBrain processes visual tokens with a dedicated intra-image bidirectional attention mechanism in addition to the standard causal attention computation. Specifically, after computing causal attention over the full multimodal sequence, the model performs an additional bidirectional self-attention operation among the visual tokens belonging to each individual image. The resulting intra-image visual representations are then used to replace the corresponding outputs of the causal attention branch at the visual token positions. This design enables each image to achieve global information exchange among its own visual tokens while preserving the causal structure across the multimodal sequence and the independence between different visual observations.

Text Generation. For language outputs, RxBrain follows a standard autoregressive generation formulation. Given the multimodal prefix, including language instructions and visual observations, text tokens are generated sequentially with causal attention and optimized through next-token prediction. This enables the model to answer questions about embodied observations and produce language-based reasoning conditioned on both textual and visual context.

Multi Frames Generation. RxBrain formulates world state prediction tasks as the generation of four future image frames. Given historical observations and task context as the prefix, the model predicts

the future frames in the VAE latent space using a flow-matching objective. During generation, the four frames are denoised in parallel for efficiency, while the internal attention mask preserves the causal structure of the visual sequence. As a result, each generated frame can attend to its own frame tokens and all preceding prefix tokens, including historical observations, text context, and earlier generated frames. This design allows RxBrain to perform parallel visual denoising while maintaining causal temporal dependencies for future multi-frame prediction. At the interface level, RxBrain represents the four future frames as consecutive latent spans within a single <Video> $\cdots$ </Video> wrapper.

Interleave Generation. For joint subgoal planning, RxBrain adopts an interleave generation scheme that jointly produces text reasoning and visual state prediction within a autoregressive process with the trajectory decoded 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 denotes the reasoning text at planning step i, and $\mathbf{z}_{i,1:N}$ denotes the flow-matched latent representation of the imagined frame. The reasoning tokens are generated through standard autoregressive next-token prediction, while the <Image> token acts as a learned modality transition token that allows the model to decide when visual imagination should be performed.

After generating <Image>, the model synthesizes the corresponding latent span $\mathbf{z}_{i,1:N}$ through flow matching in the VAE latent space, conditioned on the preceding multimodal context within the full attention. Once the frame is synthesized, it is re-encoded by ViT visual encoder and appended to the context, after which the </Image> token is automatically inserted and autoregressive decoding continues with the next reasoning step.

3 Joint Planning Data Construction

Learning joint textual-visual planning requires supervision that captures both task semantics and the physical state changes produced during execution. Standard text-to-image pairs teach open-domain visual synthesis (Esser et al., 2024; Black Forest Labs, 2025; Rombach et al., 2022), but they do not capture how embodied tasks unfold over time, including object motion, hand or gripper interactions, and intermediate visual states. We therefore use embodied task videos collected from robots, wrist-mounted human demonstrations, simulation, and egocentric human activity. Processing these videos produces joint text-visual planning samples that align textual planning steps with observed state transitions and provide supervision for world state prediction and subgoal planning.

Our corpus comprises four source categories, namely Real-Robot Data, UMI Data, Simulation Data, and Egocentric Human Data. Real-Robot Data and UMI Data include self-collected sources, while detailed descriptions of all sources are provided in Section B. Figure 4 summarizes the verified corpus, which spans 46 processed source splits and 50,177.2 hours and contains 21,506,919 trainable segments retained from approximately 28.61 M candidates at a reported 75.18% pass rate. Category-level duration, source and dataset composition, and scene coverage are reported in Figure 4A, Figure 4B, and Figure 4C, respectively, with finer scene taxonomies provided in Section D.

To turn raw videos into usable supervision, we process all sources with a three-stage pipeline: Temporal Segment Annotation, Quality Verification, and Segment Structuring. This pipeline produces four post-training tasks: continuous prediction, planning, high-level planning, and final-state imagining. These tasks provide supervision at different levels of granularity, from local state transitions to higher-level planning structures.

3.1 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

$$S_i = \left(I_i^{\text{start}},\ I_i^{\text{end}},\ a_i,\ d_i,\ m_i\right), \tag{1}$$

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

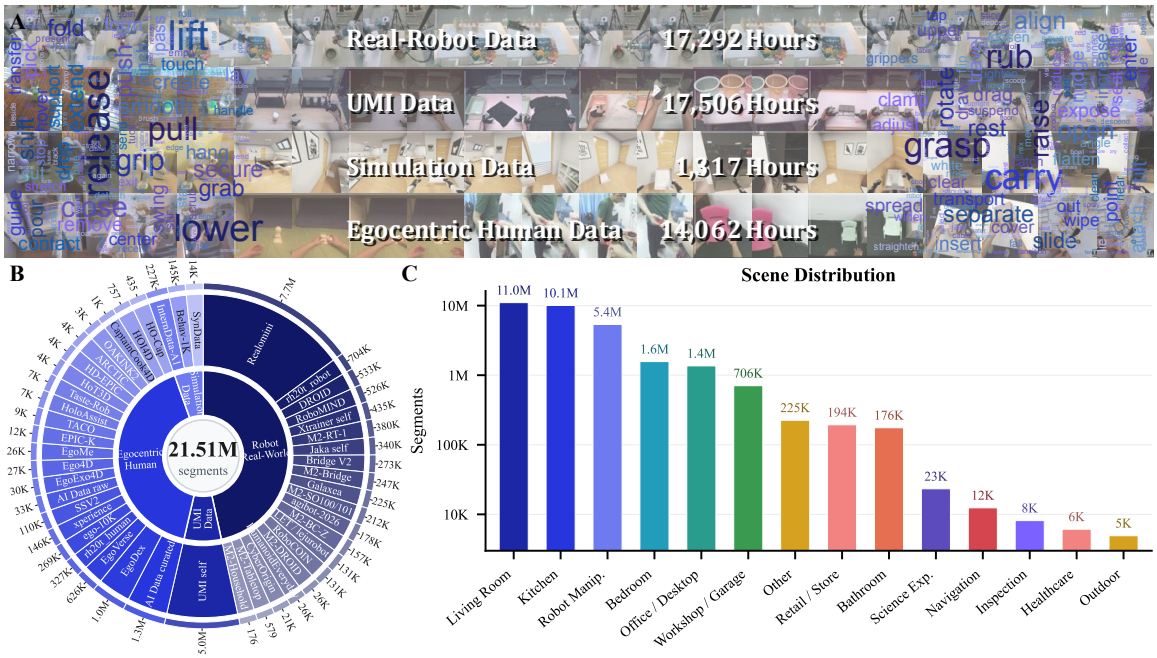


Figure 4: **Data construction overview and statistics.** This figure summarizes the verified post-training corpus by scale, source composition, and scene coverage. Panel A reports total video hours across four data categories: Real-Robot Data, UMI Data, Simulation Data, and Egocentric Human Data. Panel B shows the source composition, with source categories in the inner ring and individual datasets in the outer ring. Panel C ranks the fourteen top-level scene categories by segment count.

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 1 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

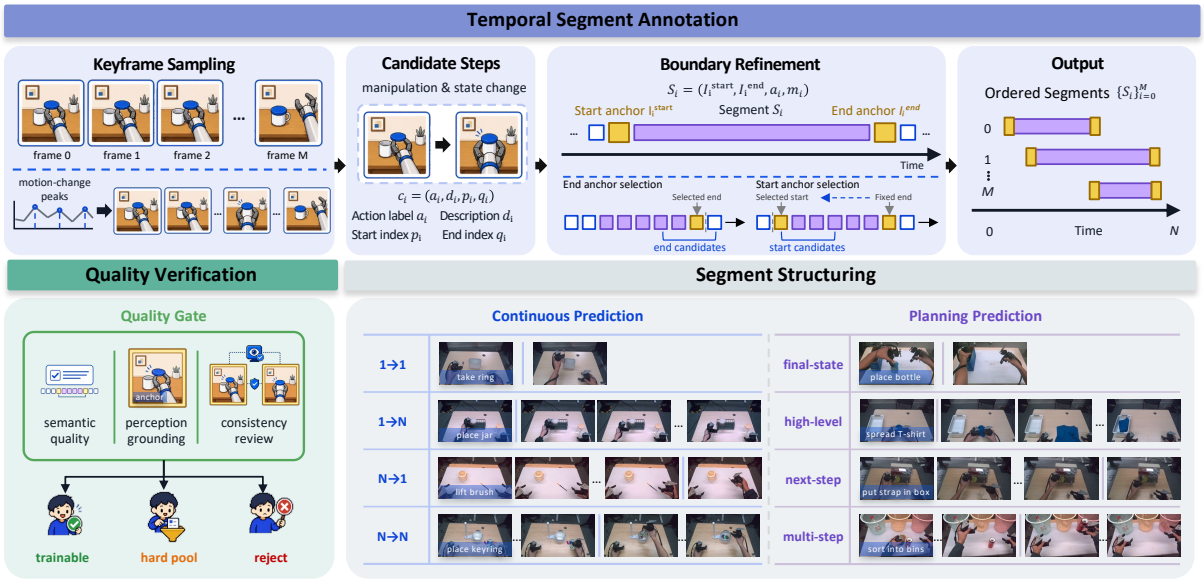


Figure 5: **Data construction pipeline.** Raw embodied task videos are converted into joint text-visual planning samples through three stages: Temporal Segment Annotation, Quality Verification, and Segment Structuring. The final samples are organized into L0–L3 tasks, covering intra-step world state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

detailed description, and metadata. Quality Verification (Section 3.2) filters these segments, and Segment Structuring (Section 3.3) organizes the retained segments into model-ready training samples.

3.2 Quality Verification

Quality Verification uses an MLLM-based verifier to decide whether each localized segment S_i from Equation 1 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 5.

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.

3.3 Segment Structuring

Segment Structuring organizes verified segments into joint text-visual planning samples for RxBrain, following the packed-sequence format in Section 2. 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.

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

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.

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.

4 Training

4.1 Training Objective

RxBrain is optimized with a unified multimodal objective that jointly trains language understanding, visual generation, and interleaved embodied generation. Different generation modes share the same MoT backbone while adopting modality-specific supervision objectives.

Text Generation Objective. For language generation, RxBrain follows the standard autoregressive formulation. Given a multimodal context c and a target text sequence $y = (y_1, \ldots, y_T)$, we optimize the next-token prediction objective:

$$\mathcal{L}_{\text{text}} = -\sum_{t=1}^{T} \log p_\theta(y_t | y_{<t}, c). \tag{2}$$

The text branch adopts causal attention, where each token attends only to previous text tokens and available multimodal context.

Image Generation Objective. RxBrain generates visual tokens in the VAE latent space using a flow-matching objective. Given a clean VAE-encoded image latent x_0 and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, the intermediate latent is constructed with a linear flow path:

$$x_t = (1 - t)x_0 + t\epsilon, \tag{3}$$

where $t \in [0, 1]$ is sampled from a logit-normal distribution. The generation branch predicts the velocity field with the flow-matching objective:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{t, x_0, \epsilon}\left[\|v_\theta(x_t, t, c) - (\epsilon - x_0)\|_2^2\right]. \tag{4}$$

The learned velocity field is used for image synthesis by solving the corresponding ODE during inference. This objective is applied to both future-frame prediction and image generation tasks.

Interleave Generation Objective. RxBrain is trained to generate interleaved language and visual states for embodied reasoning. Each training sample is represented as an ordered sequence of text segments and visual states:

$$S = \{(x_1, I_1), (x_2, I_2), \ldots, (x_N, I_N)\}, \tag{5}$$

where x_i denotes the reasoning text segment and I_i denotes the corresponding visual state. At each step k, RxBrain predicts the next text and visual state conditioned on the preceding multimodal context:

$$h_k = \{x_{<k}, \tilde{I}_{<k}\}, \tag{6}$$

where $\tilde{I}_{<k}$ denotes the visual representations of previous states. The text generation follows an autoregressive objective:

$$\mathcal{L}_{\text{text}} = -\sum_{k} \log p_\theta(x_k | h_k). \tag{7}$$

For visual generation, the model additionally conditions on the current reasoning text x_k:

$$\mathcal{L}_{\text{vision}} = \mathcal{L}_{\text{flow}}(I_k|h_k, x_k), \tag{8}$$

where $\mathcal{L}_{\text{flow}}$ is the flow-matching objective described above.

To reduce the discrepancy between training and inference contexts, we employ an **inference-aware visual state construction strategy**. During training, the previous visual states are obtained from ground-truth images through the same reconstruction and encoding pipeline used during inference:

$$\tilde{I}_i = E_{\text{vis}}\left(D_{\text{VAE}}\left(E_{\text{VAE}}(I_i)\right)\right), \tag{9}$$

where D_{VAE} and E_{VAE} denote the decoder and encoder of VAE, E_{vis} denotes the ViT encoder. During inference, the generated latent state $\hat{z}_i$ is first decoded into an image and then transformed into visual representation:

$$\tilde{I}_i = E_{\text{vis}}\left(D_{\text{VAE}}(\hat{z}_i)\right). \tag{10}$$

By aligning the visual state construction process between training and inference, the model learns to reason over visual representations consistent with those encountered during generation, while reducing the impact of error accumulation from interleave rollout.

The overall interleaved generation objective is:

$$\mathcal{L}_{\text{interleave}} = \sum_k \left(\lambda_t \mathcal{L}_{\text{text}}(x_k|h_k) + \lambda_v \mathcal{L}_{\text{vision}}(I_k|h_k, x_k)\right), \tag{11}$$

where λ_t and λ_v denote the weighting coefficients for the text generation loss and the vision generation loss respectively.

4.2 Training Stage

RxBrain is trained with a two-stage curriculum that progressively transforms the model from a general multimodal foundation model into an unified embodied model with joint language-visual reasoning and imagination abilities. The first stage focuses on establishing a unified vision-language generation capability, where the model learns to understand, reason over, and generate visual content while preserving the underlying language capability. Starting from this multimodal foundation, the second stage adapts RxBrain to embodied scenarios by introducing world modeling and long-horizon planning supervision. This progressive training strategy enables RxBrain to acquire general visual knowledge before specializing in embodied tasks.

Stage 1: Joint Vision-Language Pretraining. The first stage jointly trains all components of RxBrain using large-scale image-text pairs and multimodal instruction-following data. The objective of this stage is to establish a unified representation space that connects language understanding, visual perception, and visual generation. Image-text alignment and instruction data provide semantic grounding and visual reasoning capabilities, while visual generation supervision based on the flow-matching objective equips the model with the ability to synthesize visual states. By jointly optimizing understanding and generation objectives, this stage preserves the general language capability of the base model while extending it with multimodal reasoning and generation abilities.

Stage 2: Embodied Supervised Fine-tuning. The second stage adapts RxBrain to embodied scenarios through supervised multimodal generation learning. The training mixture combines general multimodal understanding data with embodied data, including embodied instruction-following examples, multi-frame prediction samples, and interleaved planning trajectories. The embodied instruction data further enhances the model's ability to understand embodied environments and generate language responses for reasoning and planning tasks, while the multi-frame prediction objective improves the model's ability to generate future states.

For interleaved planning data, each multimodal trajectory is transformed into multiple text-image generation samples during training, where each generated visual state is paired with its corresponding language context. This training strategy enables RxBrain to learn interleaved generation over sequential language and visual states while maintaining compatibility with the inference process.

During this stage, RxBrain is optimized with a combination of language generation, visual generation, and embodied supervision objectives. More details about training hyperparameters, data mixture, and optimization settings are provided in Appendix E.

5 RxBrain-Bench

5.1 Overview

Unified multimodal models for embodied AI are increasingly expected to do more than answer questions about an observed scene or generate a future video from a fixed instruction. To act as embodied agents, interleaving the two modalities helps a lot: generate the next action in language, imagine the visual state produced by that action, use the generated state to decide what to do next, and repeat until the goal is achieved. While existing benchmarks provide valuable tests of visual understanding, embodied reasoning, and video generation, but these abilities are typically evaluated in isolation. To our knowledge, no existing benchmark directly and comprehensively evaluates the full interleaved generation loop, evaluating this *interleaved text–visual generation* capability is the primary motivation for **RxBrain-Bench**.

For embodied reasoning, RxBrain-Bench covers a broad set of evaluation dimensions, including action and trajectory understanding, initial state judgment, visual grounding and localization, complex task planning, simulation-based planning, task completion judgment, error recognition and recovery, task state estimation, spatial relation understanding, and a small portion of multi-view and temporal reasoning. These dimensions systematically evaluate a model's perception, understanding, reasoning, planning, and state judgment abilities in embodied manipulation scenarios.

Based on these dimensions, the benchmark is organized into three tracks: RxBrain-Bench-EVQA, RxBrain-Bench-WorldPred, and RxBrain-Bench-JointPlan.

5.2 Tracks

RxBrain-Bench-EVQA. The RxBrain-Bench-EVQA track evaluates whether models can understand embodied scenes and reason about task-relevant decisions from visual observations. The scenarios are selected from domains with potential commercial value, including industrial, retail, and service environments. Different from general-purpose VQA benchmarks, this track focuses on questions grounded in real-world deployment, where models must reason not only about objects and spatial relations, but also about task efficiency, operational safety, goal satisfaction, and physical constraints. Concretely, it covers several categories of embodied reasoning: high-level planning, which asks models to decompose a goal into executable steps; simulation-based planning, which evaluates multi-round planning by requiring models to update subsequent decisions according to simulated state transitions and intermediate outcomes; fine-grained step reasoning over grasp points, motion trajectories, action selection, temporal order, and multi-view observations; error recovery, which tests whether models can identify abnormal states and choose appropriate responses; and task-completion judgment, which requires models to determine whether the current scene state satisfies the task goal. To make the evaluation discriminative, we design distractors around realistic physical conflicts, production safety rules, and trade-offs between safety and efficiency.

RxBrain-Bench-WorldPred. The RxBrain-Bench-WorldPred track evaluates visual goal imagination in embodied scenarios. Given a current observation and a natural-language action instruction, a model is required to generate a short future video depicting the physical state transition caused by the action. This tests whether the model can imagine not only what action is being executed, but also how the scene should change as a result. Unlike existing embodied video generation benchmarks (Zhou et al., 2025c; Deng et al., 2026) that emphasize long-horizon synthesis, our track focuses on short-horizon future prediction. This setting better matches the closed-loop nature of embodied control, where an agent repeatedly predicts the near future, executes an action, and re-grounds its next decision on new observations. Short-horizon prediction also reduces error accumulation in long rollouts and provides the visual look-ahead needed for step-by-step planning.

RxBrain-Bench-JointPlan. The RxBrain-Bench-JointPlan track evaluates the core capability targeted by RxBrain-Bench: representing an embodied plan through joint textual reasoning and visual goal imagination. Unlike Embodied VQA, which focuses on understanding and reasoning over the current scene, or Embodied Video Generation, which focuses on predicting the visual consequence of a given action, this track requires a model to use text and images as complementary parts of the same plan. Given a goal and an initial observation, the model must describe the next action in language, generate the corresponding visual state after that action, judge progress toward the goal, and continue this process until the task is completed. In this setting, language specifies the action logic, constraints, and decisions, while visual imagination specifies the intermediate and final world states associated with them. This track therefore evaluates whether a model can move beyond separate understanding or generation tasks and jointly represent what should be done and what physical state should be achieved.

5.3 Evaluation Metrics

We evaluate the three benchmark tracks using task-specific metrics. Embodied VQA is evaluated by exact-match accuracy. The two generative tracks (Embodied Video Generation and Multimodal Embodied Planning) are evaluated primarily by an **MLLM-as-judge** using task-specific weighted rubrics, together with auxiliary fidelity metrics computed against the ground-truth frames. We use a fixed GPT-5.5 model as the evaluator and require structured JSON outputs for all judgments (Singh et al., 2025). Each criterion is rated on a five-point scale, and the reported score is averaged over $k = 2$ independent evaluations.

EVQA. All questions are multiple-choice with a single correct answer among physically grounded distractors. We report exact-match **accuracy**, both overall and macro-averaged across the four question categories (high-level planning, fine-grained step reasoning, error recovery, and task-completion judgment).

WorldPred. Given the current observation and an action instruction, the model predicts future frames. We evaluate each generated trajectory using five criteria: *Observation Continuity (OC)*, *Action Correctness (AC)*, *Goal Completion (GC)*, *Temporal Plausibility (TP)*, and *Ground-Truth Trajectory Match (TM)*. The overall score is the weighted sum

$$S_{\text{gen}} = 0.15\,\text{OC} + 0.30\,\text{AC} + 0.20\,\text{GC} + 0.20\,\text{TP} + 0.15\,\text{TM}.$$

Action Correctness and Temporal Plausibility receive the largest weights, as they measure whether the predicted future both follows the commanded action and remains physically coherent over time. We additionally report PSNR and DINO similarity as auxiliary frame-level fidelity metrics.

JointPlan. Planning is evaluated in a fully autoregressive (free-running) setting, where the generated reasoning text and imagined visual state at each step are fed back as inputs for subsequent prediction. This protocol evaluates true closed-loop planning and naturally captures error accumulation over long horizons.

Each planning trajectory is assessed using six criteria: *Observation Understanding (OU)*, *Subtask Planning (SP)*, *Goal-Image Correctness (GI)*, *Subtask–Goal Consistency (SG)*, *Chain Completion (CC)*, and *Image Similarity (IS)*. The weighted planning score is defined as

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

The first five criteria are rated by the MLLM judge, while Image Similarity is computed as the DINO cosine similarity between each generated visual state and its corresponding ground-truth state. The three step-level criteria (SP, GI, and SG) are averaged across the entire planning trajectory. We report both the overall planning score and the per-criterion breakdown, enabling separate analysis of failures in textual reasoning and visual prediction. The full generation-side evaluation protocol is provided in Section F. It specifies the exact task inputs and held-out set sizes for both generative tracks, the GPT-5.5 VLM-as-judge configuration (multi-image prompting, structured JSON scoring, and votes=2 averaging), the complete criterion definitions and weights of the two rubrics (Table 9 for JointPlan and Table 10 for WorldPred), the local perceptual pass that supplies the DINO image-similarity term, and how each baseline is composed into a comparable interleaved planner or video generator for a like-for-like comparison.

6 Evaluation

We evaluate RxBrain along four axes: (i) its core text-to-image generation ability; (ii) its embodied and spatial *understanding* across a broad suite of benchmarks; (iii) its world state prediction ability based on multi frames generation and (iv) its joint subgoal planning ability. For the first two axes we report standard benchmark and RxBrain-EVQA comparisons against representative open and released models. For world state prediction and joint subgoal planning we report our evaluation results on RxBrain-WorldPred and RxBrain-JointPlan benchmarks.

6.1 Evaluation Tasks

We organize the evaluation along the four axes introduced above, and describe the benchmarks, baselines, and metrics used for each before presenting results in Section 6.2.

Text-to-image generation. We first assess RxBrain's core text-to-image generation ability. We use GenEval (Ghosh et al., 2023), a standard benchmark of prompt-faithful, *open-domain* text-to-image synthesis rather than any embodied skill, to verify that the unified model preserves general-purpose, world-knowledge image generation. We compare against the generation-specialist baseline Bagel (Deng et al., 2025) and the other generation-capable model Cosmos-3-Nano (Agarwal et al., 2026).

Embodied and spatial understanding. Second, we evaluate embodied and spatial *understanding* across a broad suite of public benchmarks together with our self-built RxBrain-Bench-EVQA. The public benchmarks are grouped into three families: *embodied and spatial reasoning* (ERQA (Team et al., 2025), 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 the *context/compatibility/configuration* splits of RoboSpatial-Home (Song et al., 2025)); and *multi-view and 3D understanding* (MindCube (Yin et al., 2025), MMSI-Bench (Yang et al., 2025), ViewSpatial (Li et al., 2025), 3DRSBench (Ma et al., 2024), SITE-Bench-Image (Wang et al., 2025b), VSI-bench (Yang et al., 2024a)). RxBrain is compared against representative open and released vision–language and embodied models; higher is better for all metrics.

To probe physical-world understanding beyond generic image QA, we additionally construct RxBrain-Bench-EVQA, a self-built multiple-choice VQA benchmark grounded in *real* robot-manipulation episodes. Each question is built from frames sampled from task-execution videos and posed as a multiple-choice problem—four options, or two for binary yes/no judgments—with hand-written distractors, and answered by a single option letter. Questions are grounded in one to nine images (median three, mean 4.5)—covering single observations, multiple synchronized camera views, and temporally ordered frames—so that answering requires genuine visual and physical reasoning rather than language priors. The benchmark comprises **1,123** multiple-choice questions spanning ten categories, which we group into four families: (i) *perception and state*—State Estimation and State Understanding; (ii) *spatial and grounding*—Spatial Reasoning and Pointing (localizing a queried target); (iii) *action, trajectory and temporal*—Action Reasoning, Trajectory Reasoning, and Temporal Reasoning (recovering the chronological order of shuffled frames); and (iv) *outcome and planning*—Success Detection, Error Recovery, Complex Planning, and Simulation Planning. Simulation Planning is a separate multi-round task-planning split with **258** samples, evaluated by a weighted MLLM judge. Together, the benchmark contains **1,381** evaluated items. Items come from a model-assisted, verified pool together with a human-authored cognition set, and every item is checked to have a unique, image-grounded correct answer; the temporal-ordering split is additionally curated with a strong external VLM. We report top-1 accuracy for the multiple-choice subtasks and the weighted judge score for Simulation Planning.

World state prediction. Third, we evaluate world state prediction on RxBrain-Bench-WorldPred: given the current observation and an action instruction, the model predicts a short-horizon (4-frame) future. Since no released *unified* model supports instruction-conditioned short-horizon video generation in our closed-loop setting, we construct strong baselines from released components—the general-purpose video model *Wan2.2-TI2V-5B* (Wan et al., 2025) and the embodied world model *Cosmos3-Nano* (Agarwal et al., 2026). All evaluation trajectories are held out from training, and we report the weighted MLLM-judge score S_{gen} (Section 5.3) as the primary metric.

Joint subgoal planning. Finally, we evaluate joint subgoal planning on RxBrain-Bench-JointPlan, which requires interleaved text-and-image planning under a fully autoregressive (free-running) closed-loop rollout. Since no released unified model supports this setting either, we again build strong baselines from released components: a modular *Qwen-Agent* (Qwen3-VL-2B (Bai et al., 2025) as the reasoner with Qwen-Image-Edit as the image generator), an agent built upon the omni model *Cosmos3-Nano* (Agarwal et al., 2026), and the unified *BAGEL-7B-MoT* (Deng et al., 2025) run as a planning agent. All evaluation trajectories are held out from training, and we report the weighted MLLM-judge score S_{plan} (Section 5.3) as the primary metric.

6.2 Evaluation Results

Text-to-image generation. Table 1 reports the main comparison across the four benchmark families of Section 6.1. RxBrain first preserves strong general image generation: GenEval places RxBrain (82.4) on par with the generation-specialist baseline Bagel (82) and well ahead of the other generation-capable model Cosmos-3-Nano (71.68). Crucially, this generative competence is preserved *alongside*, not traded against, broad embodied understanding, on which most generation-capable models do not operate at all.

Embodied and spatial understanding. On *embodied and spatial reasoning*, RxBrain leads CV-Bench (88.59), EmbSpatial (82.3), and DA-2k (83.4), and is second-best on SAT (74.33). Its clearest advantage is in *multi-view and 3D understanding*, where it attains the best score on 3DRSBench (54.1), MMSI-Bench (35.8), MindCube (47.5), and SITE-Bench-Image (54.9), and second-best on ViewSpatial (47.3) and VSI-bench (43.53). On *robot affordance, trajectory, and placement* it gives the best trajectory prediction (Share-Robot-Trajectory, 51.13) and leads RoboSpatial-Home configuration (86.28); fine-grained pointing and affordance (RefSpatial, Where2Place, Share-Robot-Affordance) remain the main gap, where the pointing-specialist RoboBrain2.5 (Tan et al., 2026) is stronger. Overall, this combination—general prompt-faithful generation

Table 1: **Main comparison on embodied reasoning and image generation benchmarks.** RxBrain is compared against representative open and released vision–language and embodied models across 19 benchmarks, grouped into four families: text-to-image generation, embodied and spatial reasoning, robot affordance/ trajectory/placement, and multi-view/3D understanding. GenEval measures general open-domain text-to-image generation to demonstrate model's world knowledge generation ability. Larger models, family-specific variants, sparsely evaluated models, diagnostic splits, and benchmarks without reported RxBrain results are included in the full comparison in the appendix. Benchmarks are listed as rows and models as columns; higher is better for all metrics and "–" marks entries that are not applicable or not reported. RoboSpatial-Home is split into its *context/compatibility/ configuration* sub-rows. Best result per row is in **bold**, and second-best result is underlined.

Benchmark	Qwen3.5 4B	Qwen3.5 2B	Bagel 14B(A7B)	RoboBrain2.5 4B	Cosmos-3-Nano 16B(A8B)	**RxBrain** 6B
Text-to-image generation						
GenEval	–	–	82	–	71.68	**82.4**
Embodied & spatial reasoning						
ERQA	45.8	39.8	39.5	35.8	**46**	42.8
EmbSpatial	74.2	64.3	67.0	80.2	82.1	**82.3**
CV-Bench	85.9	78.98	77.4	87.4	87.9	**88.59**
SAT	**78.33**	63.67	74.3	66	73.3	74.33
DA-2k	68.5	63.9	62.9	82.8	78.9	**83.4**
Robot affordance, trajectory & placement						
Share-Robot-Affordance	24.75	18.09	3.2	**50.31**	27.97	11.74
Share-Robot-Trajectory	29.41	28.51	11.78	27.74	39.02	**51.13**
RefSpatial	34.27	24.22	8.62	**62.5**	59.5	34.08
Where2Place	44.28	15.09	8.83	52.47	**60.64**	39.75
RoboSpatial-Home (context)	7.22	6.74	6.15	**56.56**	31.22	28.28
RoboSpatial-Home (compatibility)	50.5	32.38	42.86	36.19	**71.43**	67.62
RoboSpatial-Home (configuration)	82.1	71.54	73.98	83.74	86.18	**86.28**
Multi-view & 3D understanding						
MindCube	38.5	31.9	37.6	43.7	36.7	**47.5**
MMSI-Bench	31	26.6	29.8	28.3	34.6	**35.8**
ViewSpatial	42.3	36.6	38.6	41.5	**54.1**	47.3
3DRSBench	37.2	34.2	45.8	52.1	52.6	**54.1**
SITE-Bench-Image	50.7	47.6	51.4	52.1	54.8	**54.9**
VSI-bench	40.57	38.29	34.45	40.96	**52.97**	43.53

together with broad embodied and spatial competence in a single model—is the central capability RxBrain is designed to deliver, rather than trading one axis of ability for the other.

RxBrain-Bench-EVQA. Table 2 reports the per-subtask results on RxBrain-Bench-EVQA. RxBrain achieves an overall score of **72.7** across the 1,123 multiple-choice questions and 258 simulation-planning samples. Its clearest strengths lie in outcome-aware reasoning and multi-round planning: the model obtains the best results on Success Detection (**85.1**) and Simulation Planning (**51.6**), and achieves the second-best result on State Estimation (69.8). In particular, the Simulation Planning result indicates that Hy-Embodied-RxBrain can update its decisions over multiple rounds according to intermediate simulated states and execution outcomes, rather than producing only a one-shot task decomposition. Overall performance remains below Qwen3.5-4B (78.1) and Cosmos3-Nano (76.7), with the main gaps appearing in fine-grained spatial grounding, trajectory and temporal reasoning, and error recovery. These results suggest that Hy-Embodied-RxBrain is particularly effective at judging task outcomes and maintaining goal-directed plans, while more precise perception and physical-state reasoning remain important directions for improvement.

World state prediction and joint subgoal planning. Tables 4 and 3 summarize RxBrain on the two *generative* tracks of RxBrain-Bench, including both the per-subset/per-criterion breakdowns and comparisons with the baselines described in Section 6.1. We report the weighted MLLM-judge scores S_{plan} and S_{gen} (§5.3) as the primary metrics.

On RxBrain-Bench-WorldPred, RxBrain achieves $S_{\text{gen}} = 0.62$. Observation continuity (0.67) and action correctness (0.65) receive the highest scores, while temporal and physical plausibility (0.53) remains the weakest aspect, suggesting that motion consistency is the main limitation of the current model. Performance ranges from 0.73 on *umi* to 0.35 on *arctic*, with two-frame conditioning providing the best overall results. Compared with the baselines (Table 3), RxBrain substantially outperforms the general-purpose video model Wan2.2-TI2V-5B (0.429) and edges out Cosmos3-Nano (0.591), despite the latter

Table 2: **RxBrain-Bench-EVQA: per-subtask accuracy breakdown (%).** Ten multiple-choice subtasks covering 1,114 of 1,123 questions; three rare categories ($n \leq 4$) are omitted from the per-row listing due to insufficient sample size. Best result per row is **bold**; second-best is underlined. †RxBrain Temporal Reasoning evaluated on 100 of 102 questions (two items filtered). ‡Simulation Planning is a separate multi-round task-planning split (n=258) scored by a weighted MLLM judge (rescaled to %). The **Overall** row is the sample-weighted average over all evaluated items (1,123 multiple-choice questions plus the 258 simulation-planning samples, n=1,381).

Subtask	n	Qwen3.5 2B	Qwen3.5 4B	Bagel 14B(A7B)	RoboBrain2.5 4B	Cosmos-3-Nano 16B(A8B)	**RxBrain** 6B
Perception & state							
State Understanding	171	75.4	**95.3**	81.3	87.7	90.6	86.0
State Estimation	86	64.0	**84.9**	58.1	65.1	65.1	69.8
Spatial & grounding							
Spatial Reasoning	88	44.3	**89.8**	80.7	85.2	86.4	73.9
Pointing	133	62.4	72.2	69.9	64.7	**79.7**	69.9
Action, trajectory & temporal							
Action Reasoning	96	49.0	**89.6**	75.0	69.8	88.5	84.4
Trajectory Reasoning	119	42.9	78.2	66.4	72.3	**84.0**	69.7
Temporal Reasoning†	102	51.1	**81.1**	57.8	68.3	79.3	69.1
Outcome & planning							
Complex Planning	119	58.8	**92.4**	79.8	89.1	91.6	88.2
Success Detection	101	47.5	76.2	69.3	60.4	80.2	**85.1**
Error Recovery	99	65.7	96.0	**97.0**	**97.0**	**97.0**	72.7
Simulation Planning‡	258	26.8	45.3	27.4	41.2	42.0	**51.6**
Overall	1,381	51.6	**78.1**	65.1	69.7	76.7	72.7

Table 3: **RxBrain-Bench-WorldPred: method comparison.** Held-out, short-horizon (4 frames). S_{gen} is the weighted MLLM-judge score; the five columns after it are its judge criteria. Baselines: *Wan2.2-TI2V-5B* = a general-purpose I2V diffusion model; *Cosmos3-Nano* = an embodied omni world model (I2V). Both baselines and RxBrain are scored on the same held-out balanced subset with identical 4-frame alignment; n differs due to judge content-filtering.

Method	S_{gen}	ObsCont	ActCorr	GoalComp	TempPlaus	GTMatch	n
RxBrain	**0.62**	0.67	**0.65**	**0.62**	**0.53**	**0.59**	420
Cosmos3-Nano	0.591	**0.76**	0.62	0.58	0.48	0.53	420
Wan2.2-TI2V-5B	0.429	0.75	0.42	0.37	0.32	0.34	420

being a specialized embodied world model pretrained on substantially larger-scale video data. Overall, these results indicate that a single unified model can match dedicated world models on future prediction while significantly outperforming modular pipelines on interleaved multimodal planning. Across both tasks, language reasoning consistently exceeds visual generation quality, with goal-image correctness and temporal plausibility remaining the two primary bottlenecks. Figure 7 illustrates this behavior on two held-out actions: conditioned on the initial observation and the action instruction, RxBrain predicts four future frames that preserve scene layout and reproduce the intended manipulation, closely tracking the ground-truth rollout, while the general-purpose Wan2.2-TI2V baseline introduces spurious objects and appearance changes and Cosmos3-Nano advances the motion more slowly.

On RxBrain-Bench-JointPlan, RxBrain achieves $S_{\text{plan}} = 0.68$. The per-criterion breakdown shows that language reasoning is the model's strongest capability, with observation understanding (0.83) and subtask planning (0.78) substantially outperforming goal-image correctness (0.52), indicating that visual imagination remains the primary bottleneck. Performance is consistently strong across most subsets (0.61–0.75), while the fine-grained tabletop benchmark *arctic* remains the most challenging (0.48). Under free-running rollouts, performance gradually decreases with planning horizon, from 0.69 at two steps to 0.55 at eight steps, reflecting accumulated autoregressive errors. Compared with the baselines (Table 4), RxBrain substantially outperforms the Cosmos3-Nano agent (0.521), the unified BAGEL-7B-MoT (0.503), and Qwen-Agent (0.431), leading across every judge criterion of S_{plan}, with the largest gains on subtask planning and chain completion.

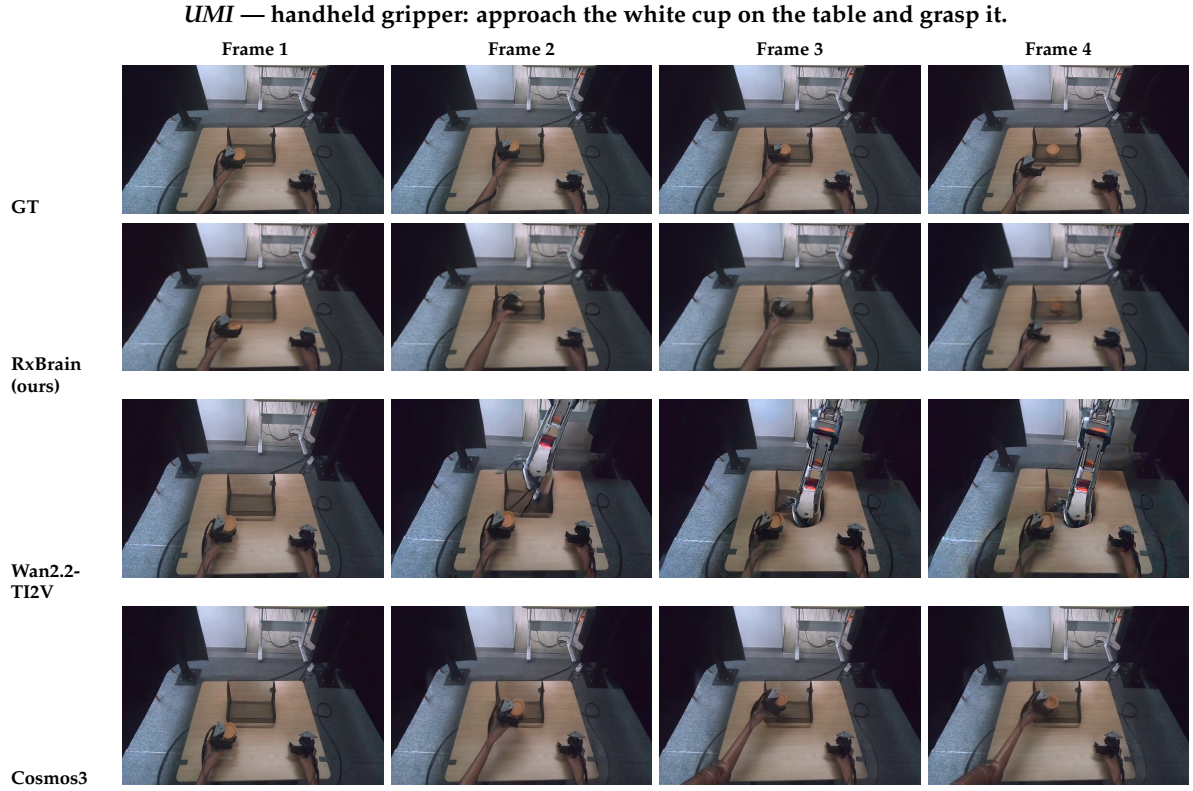


Figure 7: **RxBrain-Bench-WorldPred (*UMI*).** Four-frame future prediction for handheld manipulation. Rows compare the ground truth (*GT*), RxBrain (*HY-Unified*), the general-purpose I2V model *Wan2.2-TI2V*, and the embodied world model *Cosmos3*. RxBrain preserves the scene layout and reproduces the intended grasping behavior.

Table 4: **RxBrain-Bench-JointPlan: method comparison.** Held-outS_{plan} is the weighted MLLM-judge score; the five columns after it are its judge criteria, and DINO is the sixth rubric term (Image Similarity, folded into S_{plan}). Baselines: *Qwen-Agent* = Qwen3-VL-2B (reasoner) + Qwen-Image-Edit (generator); *Cosmos3-Nano* = a single omni model orchestrated as an agent (chat reasoner + I2V goal-image generator); *BAGEL-7B-MoT* = a unified interleaved image–text model (Deng et al., 2025) run as a planning agent. n differs slightly across methods due to judge content-filtering; all share the same held-out balanced subset (seed 0).

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)	0.521	0.67	0.62	0.26	0.47	0.63	**0.75**	540
BAGEL-7B-MoT	0.503	0.58	0.55	0.25	0.60	0.56	0.67	540
Qwen-Agent	0.431	0.57	0.53	0.20	0.44	0.53	0.52	540

Figures 8 show 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.

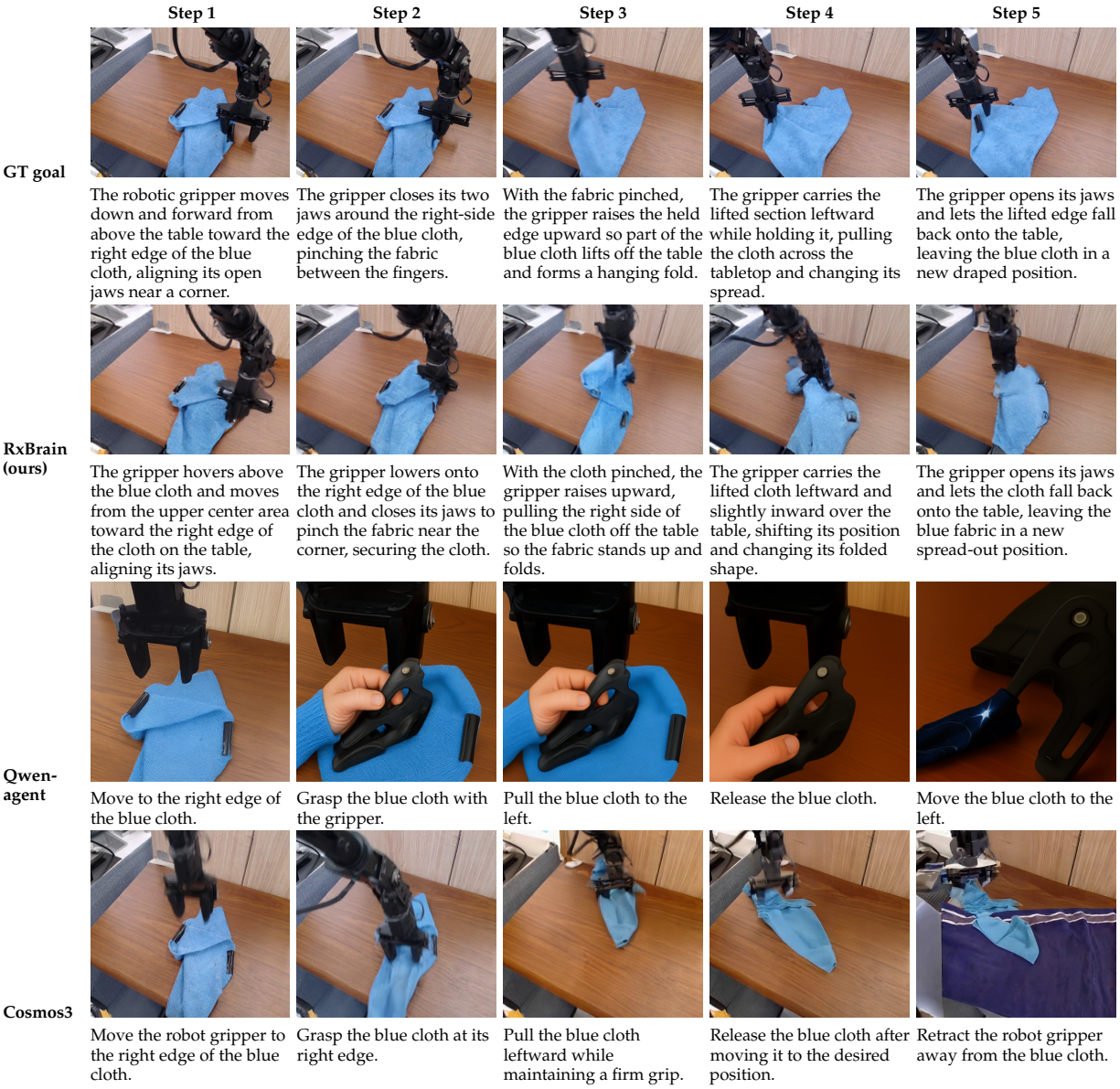


Figure 8: **RxBrain-Bench-JointPlan (*BridgeV2*), task "move to the right edge of the blue cloth, grasp it, pull it leftward, and release it."** rows are models, columns are planning steps, and each cell pairs the imagined goal frame with the model's generated sub-step text. RxBrain reproduces the grasp–pull–release sequence and the cloth's changing shape in step with the ground-truth plan (*GT goal*).

7 Extending RxBrain to Action Generation

Building on RxBrain, we extend the unified embodied model with an action-generation capability for robotic manipulation. Our goal is to investigate whether the world knowledge and state prediction priors acquired through multimodal pretraining can provide a strong foundation for action prediction without requiring large-scale action pretraining. To this end, this section presents the architecture of the proposed action model and evaluates its effectiveness on real-world robotic systems.

7.1 Action Model Architecture

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.

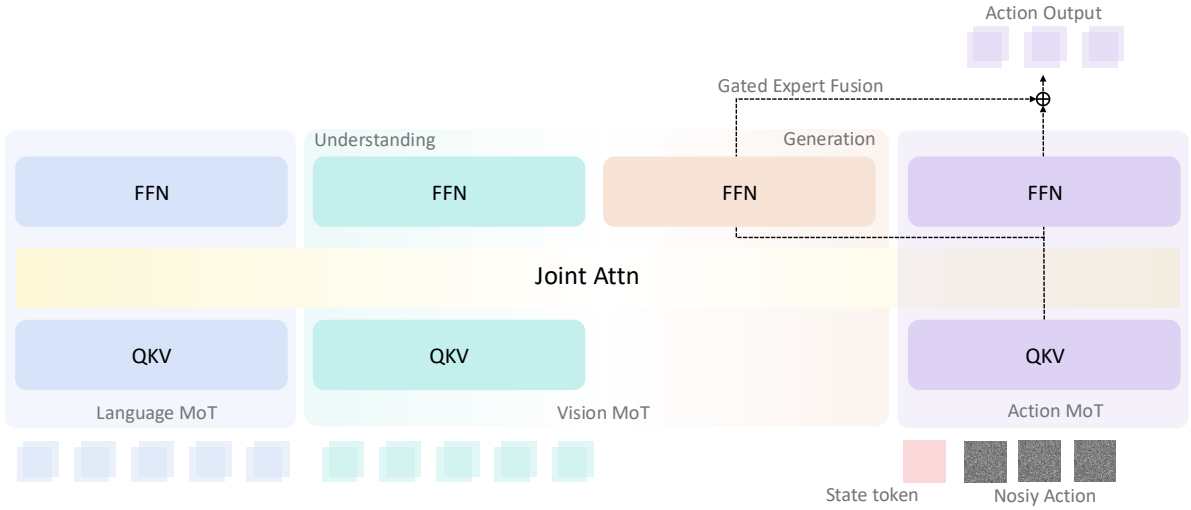


Figure 9: **Action model architecture.** A modality-specialized Action MoT branch uses dedicated attention and feed-forward parameters while sharing global self-attention with language, vision, and state tokens. A zero-initialized channel-wise gate transfers features from the pretrained generation expert 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

$$\mathrm{FFN}_a^{\mathrm{fused}}(x_a) = \mathrm{FFN}_a(\mathrm{LN}_a(x_a)) + g \odot \mathrm{FFN}_g(\mathrm{LN}_g(x_a)), \tag{12}$$

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,

$$x_a^{\mathrm{out}} = x_a + \mathrm{FFN}_a^{\mathrm{fused}}(x_a). \tag{13}$$

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,

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

. Following (Black et al., 2024) and (Physical Intelligence et al., 2025), the action chunk draws a single per-chunk flow time $s \sim \text{Beta}(1.5, 1)$ 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.

7.2 Real-world Deployment

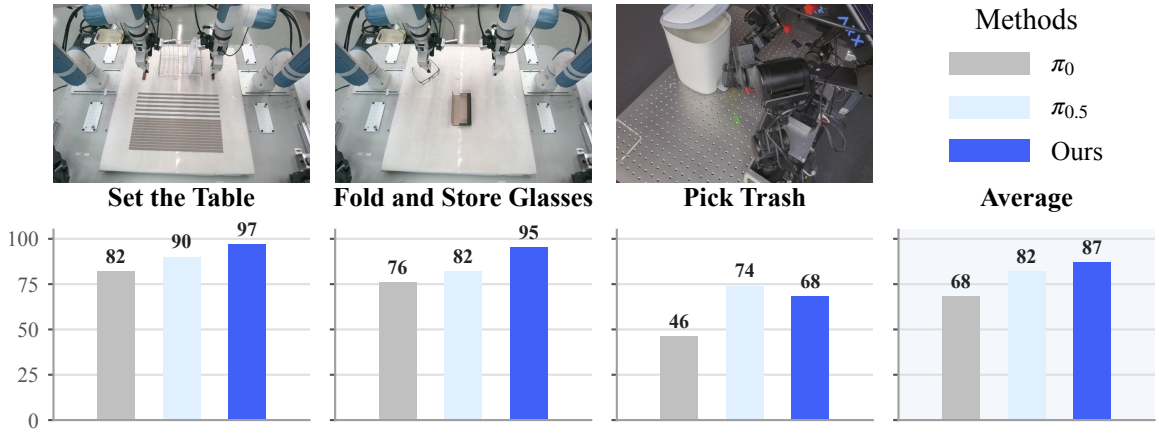


Figure 10: **Real-world robot evaluation on three manipulation tasks.** We compare our method with π_0 and $\pi_{0.5}$ on three manipulation tasks. Our method achieves the highest success rate on average.

Experimental Setup and Tasks. We further evaluate our method on two real-world robot embodiments: the DOBOT X-Trainer and an ARX dual-arm platform. The first two tasks are conducted on the DOBOT X-Trainer, while the third task is evaluated on the ARX platform. As illustrated in Fig. 10, we consider three multi-stage household manipulation tasks.

In *Set the Table*, the robot must retrieve a plate, a fork, and a knife from a storage rack and arrange them into a valid place setting on the table. In *Fold and Store Glasses*, the robot is required to fold the temples of a pair of eyeglasses and place the folded eyeglasses into a glasses case. In *Pick Trash*, the robot must first open the lid of a trash bin and then pick up and deposit a piece of waste into the bin.

The baselines and our model are all trained on the same real-world robot data. In addition, we use the joint planning data construction pipeline illustrated in Section 3 to annotate joint subgoal planning labels on real-world robot data, and these annotated data are incorporated into the training of our model.

Results. We evaluate each policy with 100 real-robot trials on each task and compare our method against two strong vision-language-action baselines, π_0 and $\pi_{0.5}$, using task success rate as the evaluation metric. As shown in Fig. 10, our method consistently outperforms both baselines across the three tasks. Specifically, our method achieves success rates of 97% on *Set the Table*, 95% on *Fold and Store Glasses* and 68% on *Pick Trash*, resulting in an average success rate of 87%. In comparison, π_0 and $\pi_{0.5}$ achieve average success rates of 68% and 82%, respectively. These results demonstrate that our action generation design can effectively perform complex manipulation behaviors, even without additional action pretraining. This suggests that the proposed approach can learn a reliable action generation policy from the pre-trained model itself, enabling robust manipulation in real-world environments.

8 Related Work

8.1 Vision-Language Models for Embodied Understanding

Vision-language models (VLMs) bridge visual perception and language. Early contrastive methods such as CLIP (Radford et al., 2021) align images and text in a shared embedding space, providing transferable representations for downstream recognition. Subsequent generative VLMs connect a pretrained vision encoder to a large language model: Flamingo (Alayrac et al., 2022) introduces gated cross-attention for few-shot multimodal learning, BLIP-2 (Li et al., 2023) bridges frozen encoders and LLMs with a lightweight querying transformer, and LLaVA (Liu et al., 2023) shows that simple visual instruction tuning can enable strong open-ended visual dialogue. More recent model families such as Qwen-VL (Bai et al., 2023; Wang et al., 2024; Bai et al., 2025) and InternVL (Chen et al., 2024; Zhu et al., 2025) scale data, resolution, and encoder capacity, achieving robust captioning, visual question answering, OCR,

long-video understanding, and spatial grounding. Natively multimodal models such as GPT-4o (OpenAI, 2024) further extend this trend by coupling visual and linguistic processing more tightly within a single model. Together, these models provide increasingly reliable scene understanding, which has begun to empower embodied agents that must ground perception in the physical world.

Building on this understanding capability, a growing line of work develops VLMs tailored to embodied scenarios. PaLM-E (Driess et al., 2023) incorporates continuous sensor inputs into a language model for embodied reasoning and planning, while RT-2 (Brohan et al., 2023a) casts robot actions as language tokens and transfers web-scale vision-language knowledge to robotic control. Recent embodied VLMs further improve spatial grounding and task-level planning. RoboBrain (Ji et al., 2025; Team, 2025) targets robotic manipulation by modeling planning, affordance perception, and trajectory prediction, and RynnBrain (Dang et al., 2025; 2026) emphasizes egocentric cognition, spatiotemporal grounding, and physical-space reasoning. MolmoAct (Lee et al., 2025; Fang et al., 2026) introduces action reasoning models that transform observations and instructions into depth-aware perception tokens, editable spatial trajectory traces, and low-level robot actions, making the planning process more interpretable and steerable. Cosmos-Reason1 (Azzolini et al., 2025) further focus on physical common sense and embodied reasoning, producing next-step decisions through multimodal reasoning. These models significantly strengthen embodied perception, spatial reasoning, and action planning. However, they mainly output language-level decisions, symbolic plans, trajectories, or actions, rather than explicitly generating future visual states.

8.2 Generative World Models

In parallel with VLMs, world models aim to predict how the environment evolves over time. Recent video generation models show that large-scale generative training can capture rich visual dynamics and synthesize temporally coherent future frames (Wan et al., 2025; Kondratyuk et al., 2023; Blattmann et al., 2023; Bruce et al., 2024; Google DeepMind, 2024; Lin et al., 2026). Beyond pixel-level generation, latent predictive models such as V-JEPA (Bardes et al., 2024) and V-JEPA 2 (Assran et al., 2025) learn to predict future or masked spatiotemporal representations, showing that video-based self-supervision can support understanding, prediction, and planning. These works suggest that video prediction and latent prediction provide useful forms of visual imagination.

Recent embodied world action models (WAMs) bring this predictive ability closer to robot control. Video Prediction Policy (VPP) (Hu et al., 2024) uses representations from pretrained video diffusion models to guide robot policy learning through predicted future visual features. Ctrl-World (Guo et al., 2025) develops a controllable multi-view generative world model for policy-in-the-loop rollout, enabling fine-grained action-conditioned prediction and long-horizon robot policy evaluation in imagination. LingBot-VA (Li et al., 2026) jointly learns frame prediction and policy execution in a causal autoregressive diffusion framework, interleaving visual and action tokens for closed-loop robot control. DreamZero (Ye et al., 2026b) further formulates world action models as zero-shot policies by jointly predicting future world states and actions with a pretrained video diffusion backbone. These models explicitly connect future prediction with action generation, making them closer to embodied imagination than standard VLMs. However, they mainly emphasize generative prediction and policy learning, while explicit scene understanding remains limited. They do not naturally support language-based inspection of imagined scenes or semantic verification of whether a predicted future state satisfies a task goal.

8.3 Unified Multimodal Models for Understanding-Generation

A more recent line of work seeks to unify visual understanding and generation within a single model, rather than treating them as separate systems. One family couples an autoregressive language backbone with a diffusion objective for images: Transfusion (Zhou et al., 2025a) trains a single transformer with a language-modeling loss on text tokens and a diffusion loss on image patches, so that one model both understands and generates. Chameleon (Chameleon Team, 2024) instead tokenizes images and text into a shared discrete vocabulary and models them with a single early-fusion autoregressive objective. Building on such mixed-modal training, BAGEL (Deng et al., 2025) scales an interleaved image-text model and reports emerging capabilities such as image editing and free-form visual manipulation, while unified frameworks like Janus (Wu et al., 2024) decouple the visual encoding pathways for understanding and generation to reduce interference between the two objectives. These models demonstrate that understanding and generation can share parameters and benefit from joint training.

More recently, this unification has been pushed toward embodied and physical-world settings. Cosmos3 (Agarwal et al., 2026) combines visual understanding, visual generation, world simulation, and action modeling within a unified Mixture-of-Transformers (MoT) architecture. This provides a strong foundation for embodied agents by enabling models to both interpret the current scene and imagine possible future states.

Moving toward next-generation embodied cognition foundation models, we argue that models should not only possess strong embodied language reasoning and world state prediction capabilities, but also provide language-visual subgoal guidance, where high-level goals can be translated into intermediate visual objectives for planning and verification. Cosmos3 adopts specialized reasoning and generation pathways with different task modes to achieve multimodal world modeling. Inspired by this direction, we explore a more unified architecture that enables embodied reasoning, world state prediction, and joint language-visual subgoal planning within an interleaved reasoning process. This motivates the design of RxBrain.

8.4 Action Generation with Unified Models

Embodied action generation has increasingly been studied using large pretrained models, including VLM-based (Yuan et al., 2026; Wu et al., 2026; Chen et al., 2025; Physical Intelligence et al., 2025; Kim et al., 2024) and World Model based (Gao et al., 2026; Ye et al., 2026c; Li et al., 2026; Ye et al., 2026a; Kim et al., 2026)policies. Recent works explore unified multimodal models for embodied action generation, where actions are modeled jointly with visual and language modalities. UP-VLA (Zhang et al., 2025) incorporates visual understanding and future prediction objectives into VLA training, while MM-ACT (Liang et al., 2025) introduces a unified discrete diffusion framework that jointly models images, language, and actions. Beyond multimodal action generation, RynnVLA-002 (Cen et al., 2025) further integrate action modeling with world dynamics by jointly learning future states and executable behaviors. Motus (Bi et al., 2025) further explores unified world modeling for embodied intelligence by integrating multiple paradigms, including vision-language-action modeling, world modeling, video generation, inverse dynamics, and video-action joint prediction, into a MoT framework. InternVLA-A1 (Cai et al., 2026) and BagelVLA (Hu et al., 2026) incorporate visual reasoning, generation, and action prediction, while Cosmos3 (Agarwal et al., 2026) unifies language, vision, audio, and physical actions within a unified physical AI foundation model.

9 Conclusion

We introduced RxBrain, an embodied cognition foundation model that represents embodied plans through joint language-visual reasoning and imagination. The core idea is that embodied planning should connect the logical structure of a task with the physical states to be achieved: language specifies planning steps, constraints, and decisions, while visual imagination grounds them in goal and intermediate world states. To support this capability, we developed a unified multimodal architecture, an automatic pipeline for constructing joint text-visual planning supervision from embodied videos, and Rxbrain-Bench for evaluating visual understanding, visual goal imagination, and joint embodied planning. Experiments suggest that RxBrain can maintain general multimodal capabilities while beginning to produce plans with complementary textual and visual components. Meanwhile, action generation based on RxBrain achieves strong performance on real-world robotic tasks. We hope this work provides an initial step toward embodied foundation models that move beyond isolated understanding or generation and enable joint subgoal planning through symbolic and visual representations.

Limitations. While RxBrain demonstrates the potential of joint language-visual reasoning and imagination for embodied planning, it still has several limitations. First, the current model is relatively small in scale, which may limit its ability to handle highly complex reasoning, fine-grained visual imagination, and long-horizon planning. Scaling both the model and the training data is likely to further improve the quality and consistency of its joint planning outputs. Second, although RxBrain can generalize across diverse embodied scenarios, fully out-of-distribution environments, embodiments, or task domains still require additional fine-tuning to achieve better performance. Third, interleaved generation involves two distinct generative paradigms—autoregressive modeling for text and flow matching for visual outputs—which introduces discrepancies between training and inference across modalities. Although we mitigate this issue to some extent by shifting ground-truth VAE latents during training, we have not yet incorporated reinforcement learning or other sequence-level optimization methods to further align and improve the interleaved generation process. Addressing this limitation requires further improvements to both the model architecture and the underlying training infrastructure.

Future work. Future work will focus on two directions. First, we will scale RxBrain to larger model sizes. We expect larger models to strengthen joint textual-visual reasoning and imagination, enabling more coherent planning primitives, more accurate world state prediction, and more reliable joint subgoal planning. Beyond improving the quality of coupled joint subgoal plan representations, we also aim to enhance the model's agentic autonomy, allowing it to better decompose open-ended tasks, monitor progress, revise plans, and interact with environments in a more self-directed manner. These directions will further move RxBrain toward a more capable embodied cognition foundation model.

A Contributors and Acknowledgements

Project Sponsors: Zhengyou Zhang† and Han Hu

Project Leaders: Yufei Huang and Yuchun Guo

Core Contributors: Haotian Liang*, Mingkang Chen*, Xiaomeng Zhu, Xiangli Shi, Kaixuan Wang, Yunxuan Mao, Weijie Zhou, Ling Chen

Contributors: Shirong Zeng, Yueyu Long, Yuchen Si, Yajuan Zhu, Xingyu Zhou, Minghui Wang, Wanjia He, Xin Yang, Lingzhu Xiang, Zhiqing Liu, Bohan Ma, Xiran Huang, Xuantang Xiong, Zisheng Lu, Tianshuo Yang, Zhiheng Liu

We appreciate contributions from Ping Luo and Yao Mu for fruitful discussions.

† Corresponding author. * Equal contribution.

B Data Sources

Section 3 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 3.2. 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., 2025). 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; Lee et al., 2025). 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 (RoboCOIN Team, 2025). 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 (Lee et al., 2025). 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; Lee et al., 2025). 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; Lee et al., 2025). 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; Lee et al., 2025). 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 (Lee et al., 2025). 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 (Lee et al., 2025). 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. The current project bibliography contains no verified EgoVerse citation, so 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. Because the current project bibliography contains no verified citation for it, 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 3.1 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 $S_i = \left(I_i^{\text{start}}, I_i^{\text{end}}, a_i, d_i, m_i\right)$ of Equation 1, 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 3.3). 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

$$\delta_k = \frac{1}{|\Omega|} \sum_{u \in \Omega} \left| \psi(f_{j_k})[u] - \psi(f_{j_{k-1}})[u] \right|, \tag{14}$$

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

$$\theta_\delta = \max\Big(Q_{p_{\text{lo}}}(\bar{\delta}),\ Q_{0.5}(\bar{\delta}) + \beta \left[Q_{p_{\text{hi}}}(\bar{\delta}) - Q_{0.5}(\bar{\delta}) \right] \Big), \tag{15}$$

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 $Q_{p_{\text{hi}}}$ and the median as an adaptive dispersion margin scaled by β, and the floor $Q_{p_{\text{lo}}}$ 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 $p_{\text{lo}} = 0.8$, $p_{\text{hi}} = 0.9$, $\beta = 0.2$, $\Delta_{\min} = 0.2\,\text{s}$, $g_{\max} = 0.5\,\text{s}$, $A_{\max} = 50$, 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 $j^\star$,

$$\hat{\jmath}_0 = \min\{\, j \in \mathcal{G} \ :\ j \ge j^\star \,\} \ \ge\ j^\star, \tag{16}$$

so $\hat{\jmath}_0$ 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 16 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

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 5: **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*

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.

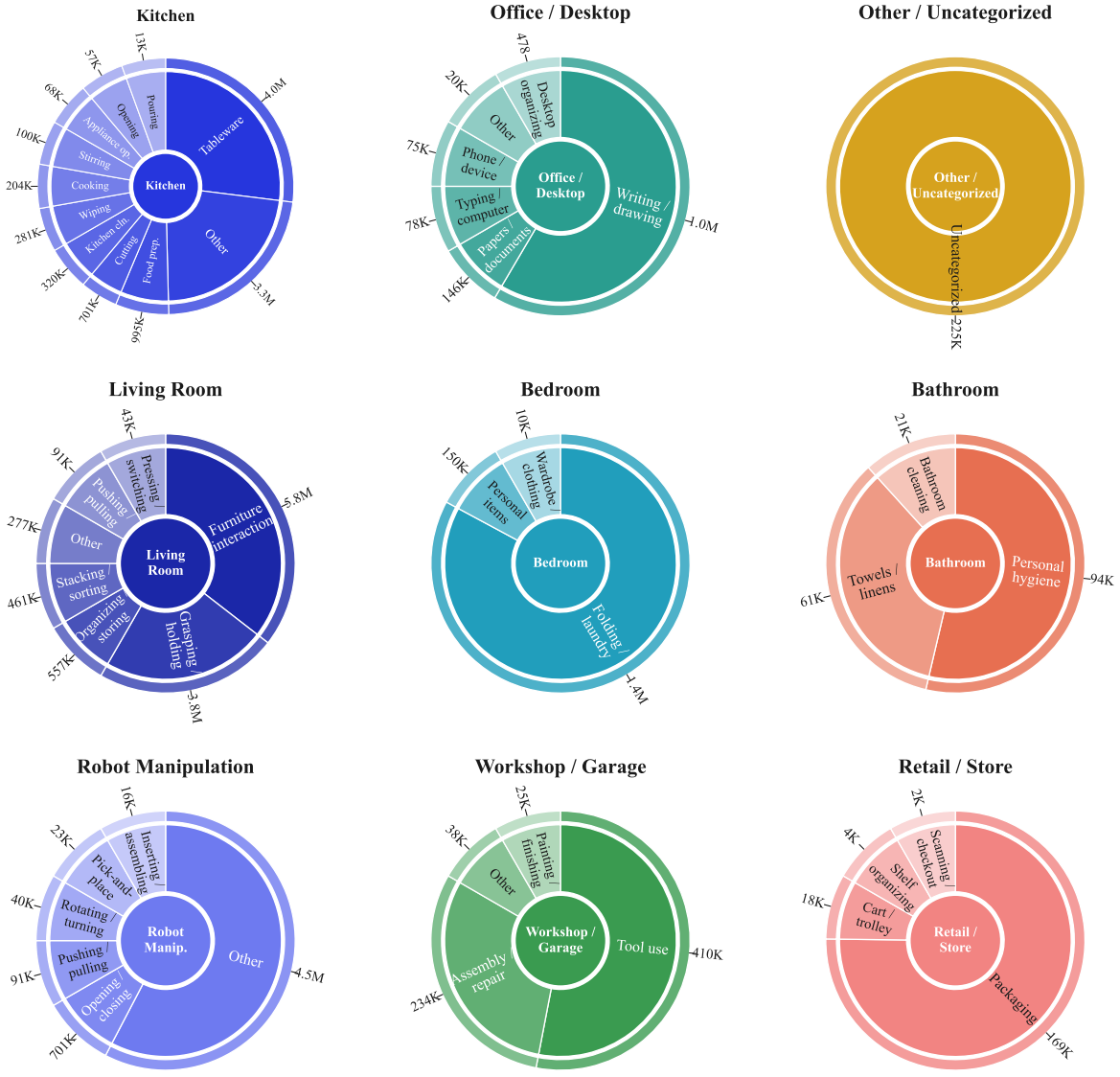
C.2 Quality Verification Review Items

Table 5 expands the Quality Verification summary in Section 3.2 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.



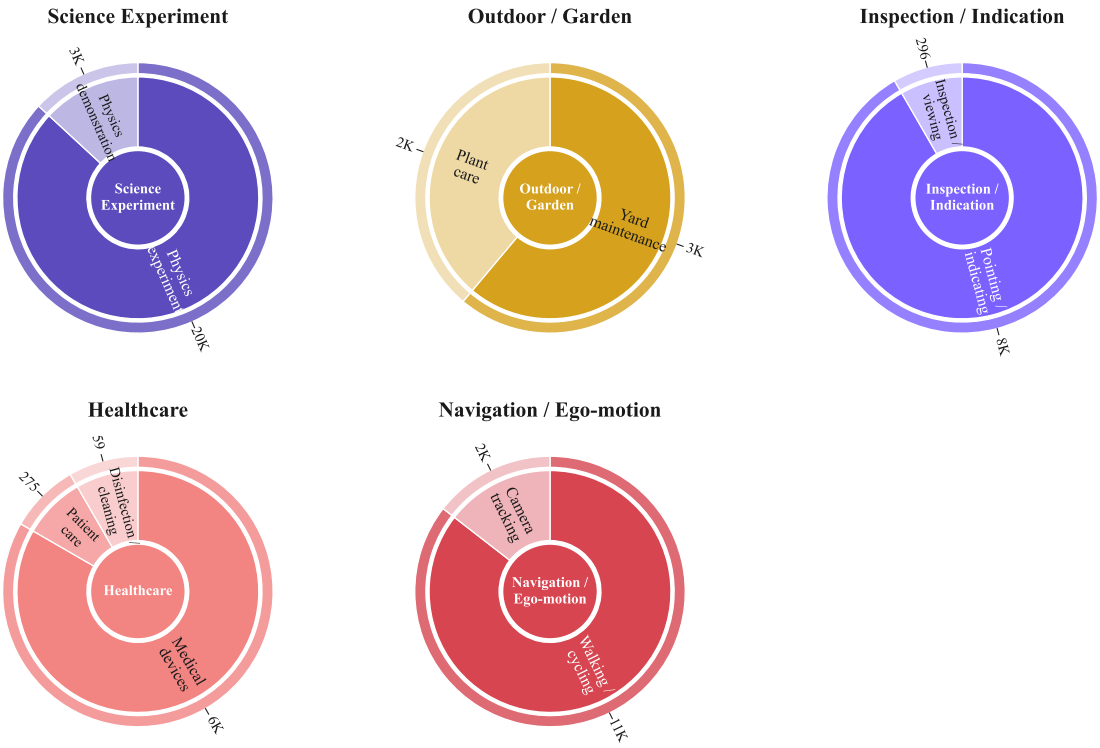


Figure 11: **Per-scene taxonomy sunburst charts (14 scene categories).** Each panel is one self-contained chart for a single scene category, flowing left-to-right and top-to-bottom across three columns (and across pages as needed). The colored center disc names the scene category; the ring around it 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 7 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.

Table 6: **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%

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 6). 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 and COYO) 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 mid-training corpus together with LLaVA-OneVision-1.5-Instruct and the MAmmoTH SI/OV collections)

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: **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)
Initialization	HY-Embodied-0.5 (all modules trainable)	Stage-1 final checkpoint
Loss weights ($\lambda_{\text{CE}}, \lambda_{\text{FM}}$)	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	$\leq$ 256 px (VAE downsample 16)	$\leq$ 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 is drawn from six buckets mixed at the integer ratio 100:12:150:70:90 (Σ=422), which sets each bucket's expected share of optimization steps (Table 8). The mixture is deliberately understanding-heavy: general and embodied understanding (b2) is the single largest source at 34.6%, and the two embodied generative buckets—multi-frame world-model prediction (b3) and interleaved planning (b4)—together account for 37.0%, while text-to-image data (b0, b1) preserves general image-generation quality. Understanding buckets are sampled length-source.

Table 8: **Stage 2 data mixture.** Six buckets mixed at integer ratio 100:12:150:70:90 (Σ=422); the per-step share is each bucket's ratio divided by 422.

Bucket	Content	Scale	Ratio	Per-step share
b0	T2I base (BLIP3o cascade-filtered)	12.0M	100	23.1%
b1	T2I premium (BLIP3o-60k + Self-Build T2I dataset-690k)	0.75M	12	2.8%
b2	Understanding / MMU (spatial CoT + knowledge + pointing)	20.8M	150	34.6%
b3	Embodied multi-frame world model	8.5M	70	16.2%
b4	Embodied interleaved planning	10.3M	90	20.8%

F RxBrain-Bench: Generation-Side Evaluation Protocol

This appendix details the evaluation methodology for the two *generative* tracks of RxBrain-Bench—interleaved planning (IL) and multi-frame video generation (MF)—including the task definitions, the VLM-as-judge protocol and rubrics, the local perceptual pass, and how each baseline is composed into a comparable generator.

Task definitions and scale. *IL (interleaved planning)* takes an observation (1–3 current-scene frames) and a task instruction, and produces an interleaved plan $[(\text{subtask text}_1, \text{goal image}_1), \dots]$: at every step the model emits both the next subtask in language and the goal image depicting the state after that subtask. The held-out set contains 4,756 trajectories across 11 datasets (`jaka`, `egodex`, `xtrainer`, `tasterob`, `taco`, `umi`, `egodex_full`, `arctic`, `bridgev2`, `let`, `let_base`), with planning depth n=1–5 (counts n_1=1536, n_2=749, n_3=366, n_4=1366, n_5=739). For cross-model comparison we draw a `balanced_subset(seed=0)` of 540 records stratified by dataset × depth, so all agents are judged on the identical records. *MF (multi-frame generation)* takes an observation (1–4 frames) and an instruction and produces a short opening text plus N continuous future frames depicting the frame-by-frame unfolding of a *single* action (a motion clip, not a multi-step plan). The held-out set contains 1,116 continuous trajectories, each with exactly 4 ground-truth keyframes (3 intermediate + 1 end).

Judge protocol. Both tracks are scored by a GPT-5.5 VLM-as-judge (Azure Responses API), with one multi-image call per record that is forced to emit a JSON schema of per-criterion scores in $[0, 1]$; we average votes=2 independent calls. All images are resized to a longest side of 512 px before upload to bound token cost. The IL judge (Table 9) receives the task, the observation frames, and per step the GT subtask text, the model subtask text, the GT goal image, and the model-generated image; the MF judge (Table 10) receives the task, the observation, the model-generated frame sequence, and the GT future frames.

Table 9: **IL judge rubric** (weighted sum S_{plan}). The first five criteria are scored by GPT-5.5; image similarity is supplied by the local perceptual pass (DINO cosine). Per-step criteria are averaged over the steps of a record.

Weight	Criterion	Granularity	What it measures
10%	observation_understanding	record	whether the plan reads the current obs state correctly
25%	subtask_planning	per-step	whether each subtask is a reasonable next action
25%	goal_image_correctness	per-step	whether the generated image depicts the subtask goal state (vs. GT goal)
20%	subtask_goal_consistency	per-step	whether the generated image matches the model's own subtask text
10%	chain_completion	record	whether executing the full chain would complete the overall task
10%	image_similarity	per-step	local DINO cosine

Table 10: **MF judge rubric** (weighted sum S_{gen}). MF is a single action, so all five criteria are record-level.

Weight	Criterion	What it measures
15%	observation_continuity	whether the first generated frame continues the obs naturally
30%	action_correctness	whether the frames execute the instructed action on the correct object
20%	goal_completion	whether the final frame reaches the task's implied goal state
20%	temporal_plausibility	whether inter-frame motion is smooth and physically plausible
15%	gt_trajectory_match	whether the generated sequence matches the GT future at the action level

Perceptual pass. For every (GT, generated) frame pair we compute LPIPS, CLIP cosine, and DINO cosine locally on CPU; the DINO cosine is used as the IL *image_similarity* term. Because S_{plan} is a fixed-weight sum that simply drops a missing term without renormalizing, a record whose perceptual pass has not run forfeits the full 10% image-similarity weight rather than redistributing it; the perceptual pass must therefore precede judging for every record. To hide latency, generation (GPU-bound) is overlapped with the perceptual pass (CPU) and the judge (API): a rolling driver launches perceptual→judge on each batch of ~ 70 completed records, de-duplicating by a unique per-record key so no record is judged twice.

IL baselines: composing agents into interleaved planners. The judge is architecture-agnostic: any pipeline that emits a (text, goal image) pair per step can be scored by the same rubric. Each baseline is a REASONER (scene → stepwise text) plus a GENERATOR (subtask + previous frame → goal image) run under *free-running* rollout—the step-k goal image is conditioned on the step-$(k-1)$ *generated* image (step 0 on the last obs frame), which is exactly the error-cascading regime the judge scores.

- **Qwen-Agent** (S_{plan}=0.431): a heterogeneous pair of specialist models. Reasoner Qwen3-VL-2B-Instruct (greedy, prompted to emit exactly n steps); generator Qwen-Image-Edit as standard img2img editing (`image`=previous frame, `prompt`=subtask, cfg=4.0, 40 steps). The generator has no end-to-end goal modeling, giving the lowest goal-image and image-similarity scores.
- **Cosmos3-Nano** (S_{plan}=0.522): a single omni model run in two passes. The reasoner is a plain VLM chat call; the generator cannot produce an obs-conditioned single image (its pure-image mode is text-to-image and discards the conditioning frame), so we instead generate a short image-to-video clip (first frame = previous image, 17 frames, 30 steps, cfg=5.0) and take its last frame as the goal image. Because the chat and video services are mutually exclusive, planning and generation run as two separate service passes.
- **BAGEL-7B-MoT** (S_{plan}=0.503): a single unified model in one local pass—the simplest of the three. The reasoner is the understanding path of `interleave_inference`; the generator is BAGEL's native

image editing, where the previous frame's VAE latent enters the generation context and the classifier-free-guidance image scale (cfg_img=2.0, with cfg_text=4.0, timestep shift 3.0, 30 steps) pulls sampling toward the obs-conditioned prediction—no image-to-video detour is needed.

Overall Cosmos3-Nano (0.522) > BAGEL (0.503) > Qwen-Agent (0.431): both unified omni models beat the heterogeneous pair, and the stronger reasoner (Cosmos3-Nano) converts its advantage into a higher planning score, whereas BAGEL's native editing yields the best image–text consistency of the three.

MF baselines: continuous video generation. MF needs no reasoner—only obs + instruction → video. Both baselines are image-to-video models that natively accept a single first frame, so both use the obs *last* frame (the GT t_0 anchor) as the first frame with the instruction in the prompt, generate a continuous clip, drop the first frame, and evenly subsample 4 frames aligned to the GT's 4 keyframes.

- **Wan2.2-TI2V-5B** (S_{gen}=0.397): a general-purpose I2V diffusion model (49 frames). It has the strongest observation continuity (0.73) but weak action correctness (0.39)—the characteristic gap of a general video model on fine-grained embodied manipulation (visual fluency ≠ correct action).
- **Cosmos3-Nano** (S_{gen}=0.575): an embodied omni world model served as an I2V endpoint (first frame + prompt). Its advantage is concentrated in action semantics—action correctness 0.60 vs. Wan's 0.39.

Cosmos3-Nano (0.575) leads Wan2.2-TI2V (0.397) across the board, with the gap driven mainly by action correctness and goal completion.

Section 3 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 3.2. 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.

G Inference Acceleration for Real-Robot Deployment

In real-robot experiments the policy is served through the asynchronous inference stack of HyVLA-0.5 (Zhang et al., 2026). Because the unoptimized model has high per-step latency, asynchronous serving injects observation lag that surfaces as jitter and degrades fine manipulation, motivating the infrastructure-level acceleration described here. Every optimization is constrained to be *numerically lossless* at the single-trajectory level: for each candidate we run an open-loop ground-truth rollout over a full trajectory, overlay the predicted end-effector trajectory against ground truth per dimension, and quantify position, orientation, and gripper error. An optimization is accepted only when its fitted curves and errors match the baseline, after which we compare its latency.

The policy generates an action chunk by iterating a flow-matching Euler ODE. Each forward pass processes a fixed prefix of ≈ 989 tokens (text, vision, and proprioceptive state) together with a 16-token action chunk, and one prediction requires K forward passes (K=2 at deployment). Profiling attributes the 210 ms baseline not to matrix multiplication but to Mixture-of-Transformers (MoT) modality dispatch: every layer recomputes the text/vision/action token routing via `nonzero(modality_mask==k)`, so a single forward triggers ≈ 1091 `aten::nonzero` calls (174 ms CPU), compounded by `index_put_` (48 ms), `index` (27 ms), and `copy_` (21 ms). The boolean indexing and `.any()` guards force repeated GPU→CPU synchronization and fragment the kernel stream, serializing the pipeline. Because the token modality layout is fixed per request, this routing can be computed once and reused.

Table 11: **Per-frame inference latency**. All configurations are numerically equivalent within the bf16 rounding level (position < 0.65 mm, orientation < 0.73°).

Configuration	Latency (ms)	Speedup
Baseline (per-layer boolean dispatch)	210	1.00×
+ prefix KV cache & action-only decode	*(lossless)*	
+ de-synchronized dispatch & CUDA graph	143	1.47×

We therefore apply three lossless optimizations. **(i) Prefix KV caching:** the prefix is invariant across the K ODE steps, so we run it once per observation, cache the per-layer keys and values, and reuse them, avoiding recomputation over the ≈ 989 prefix tokens. **(ii) Action-only fast decoding:** given the cached prefix, each ODE step forwards only the 16 action tokens—attending as queries over the cached prefix and evaluating only the action-expert FFN—reducing per-step cost from $O(989{+}16)$ to $O(16)$. **(iii) De-synchronized modality dispatch** (the largest single gain): we cache the precomputed integer routing indices keyed by `id(modality_mask)` and reuse them across all 32 layers, and replace boolean indexing with `index_select` (gather) and `index_copy_` (scatter). The rewrite is numerically equivalent (ascending indices, identical write-back positions) and removes the per-layer `nonzero` calls and their synchronization. Finally, the fixed-shape decoding step is captured and replayed with a CUDA graph to eliminate per-kernel launch overhead.

Table 11 summarizes the result. The combined optimizations reduce per-frame latency from 210 ms to 143 ms (-32%, $1.47\times$) while keeping the output numerically equivalent to the baseline (position deviation < 0.65 mm, orientation $< 0.73^\circ$, i.e. at the bf16 rounding level), satisfying the latency budget for real-time closed-loop control. We further verified that FP8 quantization and image-token reduction yield no net gain at this model scale (hidden size 2048), making 143 ms the lossless latency floor.

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.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: A visual language model for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 23716–23736, 2022.

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.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

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.

Adrien Bardes, Quentin Garrido, Jean Ponce, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv:2404.08471*, 2024.

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.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

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.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *International Conference on Machine Learning (ICML)*, 2024.

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.

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.

Jun Cen, Siteng Huang, Yuqian Yuan, Kehan Li, Hangjie Yuan, Chaohui Yu, Yuming Jiang, Jiayan Guo, Xin Li, Hao Luo, Fan Wang, Fan Wang, and Deli Zhao. Rynnvla-002: A unified vision-language-action and world model. *arXiv preprint arXiv:2511.17502*, 2025.

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

Xinyi Chen, Yilun Chen, Yanwei Fu, Ning Gao, Jiaya Jia, Weiyang Jin, Hao Li, Yao Mu, Jiangmiao Pang, Yu Qiao, et al. Internvla-m1: A spatially guided vision-language-action framework for generalist robot policy. *arXiv preprint arXiv:2510.13778*, 2025.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024.

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.

Yufan Deng, Zilin Pan, Hongyu Zhang, Xiaojie Li, Ruoqing Hu, Yufei Ding, Yiming Zou, Yan Zeng, and Daquan Zhou. Rethinking video generation model for the embodied world. *arXiv preprint arXiv:2601.15282*, 2026.

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.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024.

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, et al. Molmoact2: Action reasoning models for real-world deployment. *arXiv preprint arXiv:2605.02881*, 2026.

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

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

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.

Google DeepMind. Genie 2: A large-scale foundation world model. https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/, 2024. Blog post.

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.

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*, 2025.

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.

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.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023.

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.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning (ICML)*, 2023.

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.

Haotian Liang, Xinyi Chen, Bin Wang, Mingkang Chen, Yitian Liu, Yuhao Zhang, Zanxin Chen, Tianshuo Yang, Yilun Chen, Jiangmiao Pang, et al. Mm-act: Learn from multimodal parallel generation to act. *arXiv preprint arXiv:2512.00975*, 2025.

Weixin Liang, Lili Yu, Liang Luo, Srinivasan Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, et al. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *arXiv preprint arXiv:2411.04996*, 2024.

Yeqing Lin, Minji Lee, Aakarsh Vermani, Ellena Jiang, Sebastiaan De Cooman, Matej Spetko, and Mohammed AlQuraishi. Fast and ultra-capable protein design: Advancing the frontier through atomistic se(3)-equivariance with genie 3. *bioRxiv*, 2026. doi: 10.64898/2026.05.01.722168. URL https://www.biorxiv.org/content/10.64898/2026.05.01.722168v1.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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. 3dsrbench: A comprehensive 3d spatial reasoning benchmark. *arXiv preprint arXiv:2412.07825*, 2024.

OpenAI. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

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.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pp. 8748–8763. PMLR, 2021.

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.

RoboCOIN Team. RoboCOIN: An open-sourced bimanual robotic data collection for integrated manipulation. *arXiv preprint arXiv:2511.17441*, 2025. Open-sourced by FlagOpen; full author list to be confirmed.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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

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*, 2025.

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*, 2026a.

Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, et al. Advancing open-source world models. *arXiv preprint arXiv:2601.20540*, 2026b.

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.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

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.

Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, and Ping Luo. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*, 2024.

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)*, 2025.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, Yiyu Ren, Kejia Zhang, Hui Yu, Jingmei Zhao, Shuai Zhou, Zhenqi Qiu, Houlong Xiong, Ziyu Wang, Zechen Wang, Ran Cheng, Yong-Lu Li, Yongtao Huang, Xing Zhu, Yujun Shen, and Kecheng Zheng. A pragmatic vla foundation model, 2026. URL https://arxiv.org/abs/2601.18692.

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.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, Min Cao, Peng Li, Qiuping Deng, Wenjun Mei, Xiaofeng Wang, Xinze Chen, Xinyu Zhou, Yang Wang, Yifan Chang, Yifan Li, Yukun Zhou, Yun Ye, Zhichao Liu, and Zheng Zhu. Gigaworld-policy: An efficient action-centered world–action model, 2026a. URL https://arxiv.org/abs/2603.17240.

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, 2026b. URL https://arxiv.org/abs/2602.15922.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026c.

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.

Haoqi Yuan, Zhixuan Liang, Anzhe Chen, Ye Wang, Haoyang Li, Pei Lin, Yiyang Huang, Zixing Lei, Tong Zhang, Jiazhao Zhang, et al. Qwen-robotmanip technical report: Alignment unlocks scale for robotic manipulation foundation models. *arXiv preprint arXiv:2606.17846*, 2026.

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.

He Zhang, Lingzhu Xiang, Haitao Lin, Zeyu Huang, Minghui Wang, Dingyan Zhong, Yubo Dong, Yihao Wu, Yongming Rao, Dongsheng Zhang, Wanjia He, Ling Chen, Kai Huang, Jiahao Chen, Sichang Su, Xumin Yu, Ziyi Wang, Chengwei Zhu, Xiao Teng, Yuchun Guo, Yufeng Zhang, Yuandong Liu, Rui Wang, Zisheng Lu, Han Hu, and Zhengyou Zhang. Hy-embodied-0.5-vla: From vision-language-action models to a real-world robot learning stack. *arXiv preprint arXiv:2606.14409*, 2026.

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.

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.

Fengzhe Zhou, Jiannan Huang, Jialuo Li, Deva Ramanan, and Humphrey Shi. Pai-bench: A comprehensive benchmark for physical ai, 2025c. URL https://arxiv.org/abs/2512.01989.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.