Riemann-1.0: An Embodied World Action Model for Physical AI

Riemann Dynamics
research@riemanndynamics.ai

Physical AI is driving artificial intelligence beyond understanding the digital world toward perceiving, reasoning, and acting in the physical world. World Action Models (WAMs) have emerged as an effective paradigm for jointly modeling robot actions and world dynamics. However, existing approaches struggle to effectively leverage heterogeneous embodied experience, including egocentric human videos, handheld-gripper demonstrations (e.g., UMI), and heterogeneous robot trajectories, while lacking a unified causal formulation for jointly modeling robot actions, states, and world evolution. We present **Riemann-1.0**, a fully causal autoregressive World Action Model for embodied intelligence. Riemann-1.0 jointly models multi-view visual observations, robot states, and embodiment-specific actions within a unified causal autoregressive sequence, representing robot actions and world evolution as causal state transitions. Unlike existing WAMs based on joint generation, video-first prediction, or decoupled modeling paradigms, Riemann-1.0 unifies online robot policy execution and action-conditioned world simulation within a single model, enabling it to function as both an executable robot policy and a multi-embodiment visual world simulator. To scale embodied experience across heterogeneous data sources, we further develop a progressive embodied pretraining framework that unifies learning from egocentric human videos, handheld-gripper demonstrations, and heterogeneous robot trajectories under a shared World Action Modeling objective. Built upon a unified embodied data infrastructure containing more than 200K+ hours of interaction data, Riemann-1.0 progressively transfers large-scale embodied experience into executable robot manipulation capabilities. Riemann-1.0 achieves state-of-the-art performance across both simulation benchmarks and real-world manipulation tasks. It achieves success rates of 94.3% on RoboTwin2.0, 99.0% on LIBERO, and 62.6% on the long-horizon compositional benchmark RoboCasa-365, outperforming the previous best method by 8.4 percentage points. On long-horizon real-world manipulation tasks, Riemann-1.0 achieves a Success Rate (SR) of 85.0% and a Progress Success Rate (PSR) of 94.4%, exceeding the strongest open-source baseline by 15 percentage points in SR. These results demonstrate that unified World Action Modeling together with progressive embodied pretraining effectively transforms large-scale embodied experience into generalizable robot manipulation capabilities.
Website: https://riemann-dynamics.github.io/Riemann-1.0-Website

1. Introduction

Physical AI is driving artificial intelligence beyond understanding the digital world toward perceiving, reasoning, and acting in the physical world. Building intelligent agents for real-world environments requires Embodied Brain Models that unify environmental perception, world dynamics modeling, executable action generation, and long-horizon reasoning. Similar to how Large Language Models continuously improve

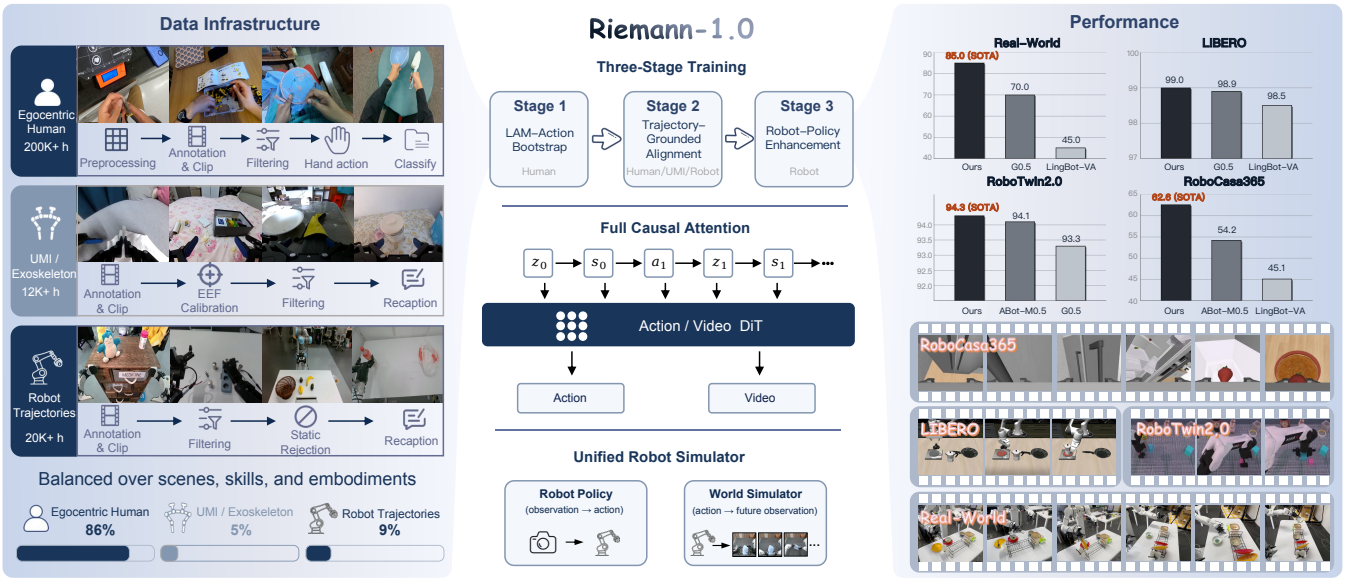

Figure 1: Overview of Riemann-1.0. Riemann-1.0 integrates a unified embodied data infrastructure, Progressive Embodied Pretraining, and a fully causal World Action Model into a unified framework for embodied intelligence. The resulting model simultaneously serves as an executable robot policy and an action-conditioned visual world simulator, achieving state-of-the-art performance across both simulation benchmarks and long-horizon real-world manipulation tasks.

by scaling textual knowledge, Embodied Brain Models should continuously evolve by scaling embodied experience. In recent years, World Action Models (WAMs) have emerged as an promising paradigm for jointly modeling robot actions and world dynamics (Ye et al., 2026b,a, Li et al., 2026, Yuan et al., 2026) , providing a unified framework for robot policy learning, predictive world modeling, and long-horizon planning. Together, these advances establish WAMs as a strong foundation for next-generation embodied intelligence.

However, embodied experience is inherently heterogeneous, multimodal, and embodiment-dependent. Substantial differences in observation modalities, action spaces, and supervision signals across data sources make unified large-scale learning fundamentally challenging. Egocentric human videos capture diverse real-world interactions, rich manipulation skills, and long-horizon task structures. Handheld-gripper demonstrations (e.g., UMI) naturally bridge human interactions with robot action spaces through robot-compatible manipulation, while robot trajectories provide precise, embodiment-specific, and executable control supervision. Rather than being redundant, these complementary data sources provide distinct forms of supervision: egocentric human videos contribute scalable interaction knowledge, handheld-gripper demonstrations bridge cross-embodiment action alignment, and robot trajectories ground embodiment-specific executable control. Motivated by this complementary supervision, we construct one of the largest embodied experience corpora to date, comprising 200K+ hours of data spanning thousands of diverse interaction skills across egocentric human videos, handheld-gripper demonstrations, and heterogeneous robot trajectories.

Despite the rapid progress of World Action Models, existing approaches remain limited in fully leveraging embodied experience at scale. First, existing pretraining strategies are tightly coupled with robot trajectories and therefore cannot effectively leverage large-scale weakly supervised human data, limiting knowledge transfer across heterogeneous embodiments. Second, existing World Action Model formulations—including

joint generation (Ye et al., 2026b), video-first prediction (Li et al., 2026), and decoupled action-video modeling (Yuan et al., 2026)—capture different aspects of robot interaction but fail to jointly model observations, robot states, actions, and world evolution within a unified causal process. Consequently, they cannot simultaneously support efficient robot policy learning and causally consistent action-conditioned world simulation within a unified model.

To address these challenges, we present Riemann-1.0, a Fully Causal Autoregressive World Action Model for embodied intelligence, as shown in Figure 1. Riemann-1.0 formulates robot interaction as a unified causal autoregressive sequence over multi-view visual observations, robot states, and embodiment-specific actions, where robot actions and world evolution are represented as causal state transitions. This fully causal formulation naturally aligns with the interaction process of real-world robots, enabling a single model to simultaneously function as both an executable robot policy and a multi-embodiment action-conditioned visual world simulator.

To effectively leverage large-scale embodied experience, we further propose Progressive Embodied Pretraining, a three-stage curriculum that progressively transfers embodied knowledge from weak supervision to executable robot control. The proposed curriculum follows a natural progression from weakly supervised embodied knowledge learning to executable action alignment, and finally embodiment-specific policy enhancement, enabling continuous capability transfer from large-scale embodied experience to executable robot manipulation. Specifically, a frozen Latent Action Model (LAM) first converts large-scale unlabeled egocentric videos into latent action representations, providing scalable weak action supervision for pretraining. Subsequently, 3D hand-annotated egocentric videos, handheld-gripper demonstrations, and heterogeneous robot trajectories progressively align latent actions with executable robot actions across heterogeneous embodiments. Finally, high-quality robot trajectories specialize the model for embodiment-specific action generation and downstream robot policy learning.

Extensive experiments demonstrate that Riemann-1.0 achieves state-of-the-art performance across both simulation benchmarks and long-horizon real-world manipulation tasks. Riemann-1.0 achieves success rates of 94.3% on RoboTwin2.0 (Chen et al., 2025a), 99.0% on LIBERO (Liu et al., 2023), and 62.6% on the long-horizon compositional benchmark RoboCasa-365 (Nasiriany et al., 2026), outperforming the best method by 8.4 percentage points. On long-horizon real-world manipulation tasks, it achieves 85.0% Success Rate (SR) and 94.4% Progress Success Rate (PSR), exceeding the strongest open-source baseline by 15 percentage points in SR. These results demonstrate that combining Fully Causal World Action Modeling with Progressive Embodied Pretraining effectively translates large-scale embodied experience into generalizable long-horizon robot manipulation capabilities.

Our contributions are summarized as follows:

- **Fully Causal Autoregressive World Action Model.** We propose a fully causal autoregressive World Action Model that jointly models multi-view visual observations, robot states, and embodiment-specific actions within a unified causal sequence. The proposed formulation unifies executable robot policy learning and action-conditioned visual world simulation within a single model.
- **Progressive Embodied Pretraining.** We develop a three-stage Progressive Embodied Pretraining framework that progressively transfers embodied knowledge from weakly supervised human interaction data to executable robot control. Built on more than 200K+ hours of embodied experience covering thousands of interaction skills, the proposed framework effectively unifies heterogeneous embodied experience under a shared World Action Modeling objective.

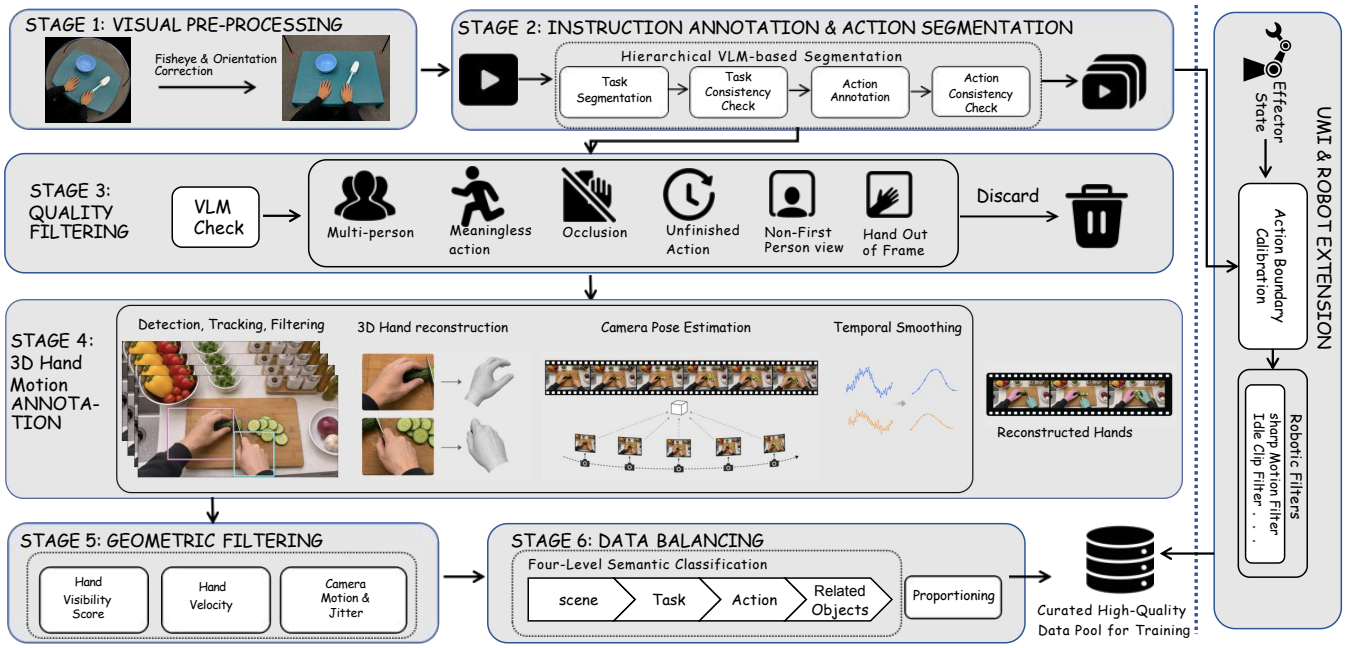

Figure 2: Unified embodied data processing pipeline. The proposed data engine transforms heterogeneous embodied experience into standardized action-level trajectories through six processing stages, including visual preprocessing, semantic annotation, quality filtering, 3D hand reconstruction, geometric filtering, and semantic-aware data balancing. Handheld-gripper demonstrations and robot trajectories are additionally refined through action calibration and robot-specific filtering. Together, these processes produce a unified embodied corpus that supports Progressive Embodied Pretraining and large-scale World Action Modeling.

- **Comprehensive Evaluation.** We demonstrate the effectiveness of the proposed framework through extensive evaluations on both simulation benchmarks and long-horizon real-world manipulation tasks. Riemann-1.0 achieves state-of-the-art performance across multiple benchmarks while significantly improving long-horizon real-world robot manipulation.

2. Data Infrastructure

Scaling World Action Models fundamentally depends on scaling embodied experience rather than robot trajectories alone. However, embodied experience is inherently heterogeneous across egocentric human videos, handheld-gripper demonstrations, and robot trajectories. These sources differ substantially in observation modalities, action representations, temporal granularity, and supervision fidelity, making unified large-scale pretraining fundamentally challenging.

To address this challenge, we build a unified embodied data infrastructure that automatically transforms heterogeneous embodied experience into a common action-level video–state–action representation. The proposed infrastructure consists of three tightly coupled components: multi-source embodied experience collection, a unified embodied data engine, and progressive supervision with multi-source data balancing. Together, they produce a scalable pretraining corpus containing more than 200K+ hours of embodied experience covering thousands of interaction skills, providing the data foundation for World Action Modeling.

2.1. Multi-Source Embodied Experience

We construct a large-scale embodied experience corpus containing more than 230K+ hours of embodied experience, including 200K+ hours of egocentric human videos, 12K+ hours of handheld-gripper and wearable demonstrations, and 20K+ hours of heterogeneous robot trajectories. The corpus spans diverse household, office, and industrial environments, covering thousands of interaction skills and a wide range of robot embodiments.

The three data sources provide complementary supervision. Egocentric human videos offer the broadest diversity of real-world interactions, object-centric manipulation skills, and long-horizon task compositions. Handheld-gripper and wearable demonstrations provide structured end-effector trajectories that naturally bridge human interactions and robot-compatible control. Robot trajectories provide precise embodiment-specific states and directly executable actions for downstream policy learning.

Rather than being redundant, these sources complement each other: human videos provide scalable interaction knowledge, handheld-gripper demonstrations reduce the embodiment gap between humans and robots, and robot trajectories provide executable control supervision. Together, they establish a natural supervision hierarchy for Progressive Embodied Pretraining.

2.2. Unified Embodied Data Engine

As illustrated in Figure 2, although the three data sources provide complementary supervision, their native formats are fundamentally different. Egocentric human videos provide visual observations with weak language supervision, handheld-gripper demonstrations provide structured end-effector trajectories and gripper states, while robot datasets differ substantially in embodiment-specific states, action spaces, coordinate systems, and control frequencies. Directly combining these heterogeneous sources would result in inconsistent supervision and unstable optimization.

To unify heterogeneous embodied experience, we build a unified embodied data engine that converts all data into a common action-level trajectory representation consisting of language instructions, embodiment identities, visual observations, states, actions, and semantic metadata. For each embodiment, state and action streams are temporally aligned, normalized, and organized under embodiment-specific canonical definitions, while validity masks handle variable-dimensional actions without forcing heterogeneous embodiments into a shared physical action space. This unified representation serves as a standardized interface between heterogeneous embodied data and the World Action Modeling objective.

For egocentric human videos, we first perform visual normalization and then apply a hierarchical VLM-based annotation pipeline that progressively decomposes long recordings into task-level and action-level clips with structured semantic annotations, including tasks, actions, scenes, manipulated objects, and temporal boundaries. Verification of consistency at task-level and action-level further improves annotation reliability. To bridge visual interactions and executable actions, we further reconstruct continuous 3D hand trajectories from video-only egocentric human data. Hand detection and tracking first establish temporally consistent hand trajectories, followed by MANO-based (Romero et al., 2022) 3D hand reconstruction to recover hand keypoints, wrist poses, and MANO parameters. In parallel, VGGT-Ω (Wang et al., 2026a) estimates camera intrinsics and frame-level camera poses, enabling hand trajectories to be organized in a consistent coordinate system under egocentric camera motion. A lightweight temporal refinement further suppresses jitter and short-term drift, producing temporally continuous hand-action trajectories for large-scale pretraining. Finally,

semantic consistency, interaction completeness, hand motion, visibility, and camera motion are jointly evaluated to retain high-quality interaction clips.

Unlike egocentric human videos, handheld-gripper demonstrations and robot trajectories already provide structured control supervision. Their processing therefore mainly focuses on temporal alignment, action normalization, and trajectory quality filtering. Gripper-state transitions are used to refine action boundaries, while camera-shake filtering, stationary end-effector rejection, and abnormal control filtering further improve data quality. After processing, all data sources are converted into temporally aligned action-level video–state–action trajectories under a unified semantic taxonomy, enabling scalable World Action Modeling across heterogeneous embodiments.

2.3. Progressive Supervision and Data Balancing

The processed corpus forms a continuum of supervision with progressively increasing action fidelity rather than a simple division between labeled and unlabeled data. Large-scale egocentric human videos first provide broad interaction dynamics and weak action supervision through pseudo actions generated by a frozen Latent Action Model. Human videos with reconstructed 3D hand trajectories, handheld-gripper demonstrations, and heterogeneous robot trajectories subsequently provide continuous real-action supervision for visual-action alignment. Finally, high-quality robot trajectories specialize the model for embodiment-specific action prediction and closed-loop policy execution. This supervision hierarchy naturally matches the three-stage training curriculum of Riemann-1.0, enabling the model to progressively transfer from open-world interaction knowledge to executable robot control.

The raw corpus exhibits severe long-tail distributions across data sources, scenes, tasks, skills, objects, and robot embodiments. Rather than balancing raw data volume, we construct a unified semantic taxonomy and perform semantic-aware sampling across multiple semantic dimensions. This strategy preserves the scale advantage of large human datasets while ensuring sufficient exposure to long-tail manipulation skills and low-resource robot embodiments.

Overall, our data infrastructure transforms heterogeneous embodied experience into a scalable supervision source for World Action Modeling, enabling World Action Models to continuously improve by scaling embodied experience rather than robot trajectories alone.

3. Riemann-1.0 Design

3.1. Preliminary

WAMs capture robot-environment dynamics by jointly modeling robot actions and their visual consequences. Given a task condition c, a visual latent z_t, a robot state s_t, and an action a_t, a WAM models the coupled evolution of control and observation over time.

Flow matching. Flow matching learns a continuous vector field that transports noise samples toward data samples. For a data variable x, Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, and noise level $\sigma \in [0, 1]$, we define the interpolated sample $x_\sigma = (1 - \sigma)x + \sigma\epsilon$. The target velocity along this path is $v^\star = \epsilon - x$. A conditional model v_θ is trained with $\mathcal{L}_{\text{fm}} = \mathbb{E}_{x,\epsilon,\sigma}[||v_\theta(x_\sigma, \sigma, c) - (\epsilon - x)||_2^2]$. This formulation naturally fits World

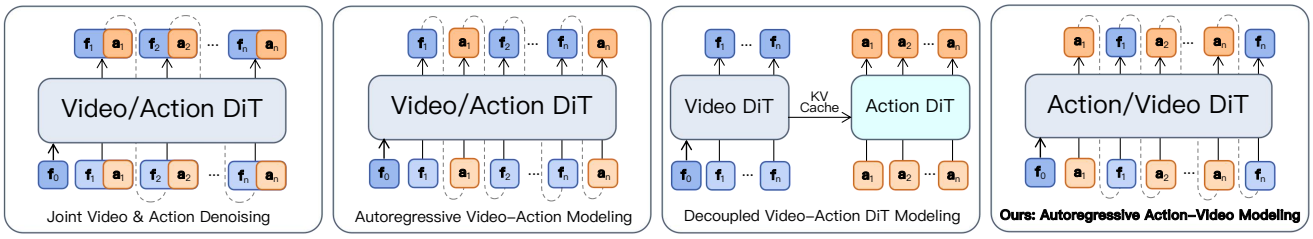

Figure 3: Comparison of joint, video-first, video-action separated DiT, and action-first WAM paradigms.

Action Modeling, enabling visual latents and continuous robot actions to be optimized within a unified velocity-regression framework.

Comparison of WAM paradigms. Existing WAM paradigms primarily differ in how they model the interaction between robot actions and future visual observations, leading to different trade-offs among policy learning, visual simulation, and inference efficiency (Figure 3). DreamZero-style methods (Ye et al., 2026b) jointly denoise visual latents and actions as $p(z_{1:T}, a_{1:T} \mid z_0, s_0, c)$. This tightly couples action and visual generation, but requires jointly modeling modalities that have different dimensionalities, temporal resolutions, and optimization characteristics. Video-first methods, such as LingBot-VA (Li et al., 2026), first generate future observations and then infer actions from the predicted visual trajectory, i.e., $p(z_{1:T} \mid z_0, c)\, p(a_{1:T} \mid z_{\leq T}, s_{\leq T}, c)$, thereby introducing additional inference latency. FastWAM-style methods (Yuan et al., 2026) instead decouple video and action generation by using two separate DiTs, improving modularity while still conditioning action prediction on predicted visual features rather than causal interaction histories. In contrast, Riemann-1.0 adopts a fully causal autoregressive action-video formulation, where robot actions are either predicted online or externally specified before future visual latents are generated. This fully causal formulation enables a single model to function both as an executable robot policy and as an action-conditioned visual world simulator.

3.2. Riemann-1.0: Fully Causal Action-Video World Model

Riemann-1.0 adopts a fully causal autoregressive factorization over actions and visual latents. As shown in Figure 4, at each autoregressive step, the model predicts the current action from the preceding visual observations, robot states, and action histories. The action is then appended to the causal context and used as a condition for predicting the corresponding visual latent. This ordering follows the causal interaction process of real robots, where actions are executed before their visual consequences are observed. During policy deployment, the predicted visual latent is replaced by the real observation returned by the environment after action execution. During visual simulation, the predicted visual latent is recursively fed back into the model to generate future action-conditioned observations:

$$p(a_{1:T}, z_{1:T} \mid z_0, s_0, c) = \prod_{t=1}^{T} p(a_t \mid z_{<t}, s_{<t}, a_{<t}, c)\, p(z_t \mid z_{<t}, s_{<t}, a_{\leq t}, c). \tag{1}$$

Robot states s_t are not generated by the model. Instead, they are directly observed from the environment or replayed from recorded trajectories and injected as conditioning signals for the next autoregressive step. This factorization preserves the causal interaction process: historical observations, robot states, and previous

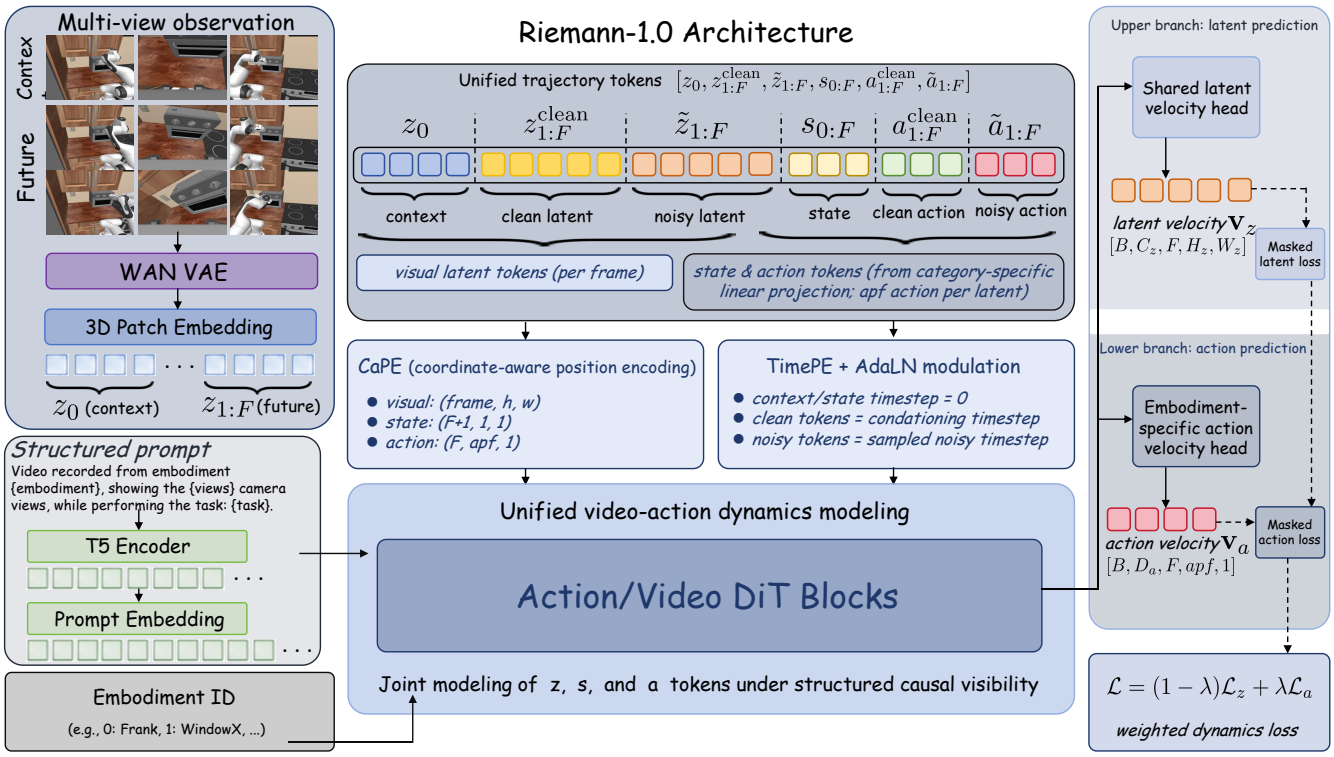

Figure 4: Causal action-video architecture of Riemann-1.0, jointly modeling action prediction and visual latent generation conditioned on task instructions, robot states, and interaction history.

actions determine the next robot action, which in turn conditions the next visual transition. The resulting observation together with the updated robot state then becomes the context for the following autoregressive step. Compared with GIGA-World-Policy (Ye et al., 2026a), which inherits its visual dynamics prior from large-scale video generation pretraining and subsequently adapts it for robot policy learning, Riemann-1.0 adopts the same fully causal action-video formulation throughout progressive embodied pretraining, post-training, and deployment. Under a unified World Action Modeling objective, Riemann-1.0 jointly learns latent or executable actions together with visual dynamics from the beginning of pretraining, allowing its dynamics prior to be directly shaped by embodied action supervision rather than transferred from a general-purpose video generation objective. Moreover, each predicted or observed visual consequence is recursively incorporated into the causal interaction history for subsequent action prediction, enabling long-horizon closed-loop interaction. This unified formulation minimizes the modeling mismatch between learned supervision and online execution, allowing the learned action-world dynamics to transfer naturally from large-scale embodied pretraining to downstream robot policy learning.

Eq. 1 defines the autoregressive factorization of the joint action-video rollout conditioned on the task prompt c, the initial visual latent z_0, and the initial state s_0. Specifically, Riemann-1.0 predicts actions from the preceding visual, state, and action histories, and generates each future visual latent from the same causal history together with the current action. Therefore, an action chunk is not only a policy output, but also a conditioning input to the visual dynamics model, which is the key interface that lets Riemann-1.0 act as a robot world simulator.

Inputs, prompt conditioning, and normalization. The language condition specifies the task instruction together with the robot embodiment and camera-view configuration:

```
Video recorded from embodiment {embodiment_type},
showing the {num_views} camera views, while performing the task:  {task}.
```

The language condition is encoded by a T5 encoder (Colin, 2020) and injected into the transformer through cross-attention, providing semantic task and embodiment context. A numerical embodiment ID is further used to select embodiment-specific action/state projections and prediction heads, enabling heterogeneous robots to share a unified backbone while preserving embodiment-specific control parameterizations. Visual observations are packed into a unified embodiment-specific multi-view canvas, resized to the model resolution, encoded into latent representations using the Wan VAE (Wan et al., 2025), and converted into transformer tokens through a 3D patch embedding. The first latent z_0 serves as the initial observation context, while future visual latents $z_{1:T}$ are generated autoregressively. Actions and robot states are transformed into embodiment-specific canonical representations and normalized using per-embodiment statistics. This normalization removes differences in coordinate systems, physical units, and action dimensionalities while preserving embodiment-specific control semantics, allowing heterogeneous embodiments to share a unified World Action Modeling interface.

Embodiment-specific state and action interfaces. Robots differ in action dimensions, state dimensions, coordinate conventions, and semantic layouts. Riemann-1.0 therefore shares the transformer backbone while specializing the input and output interfaces for each embodiment. For an embodiment e, raw action and state vectors are first converted to a canonical order defined by that embodiment. They are then padded to fixed model dimensions $d_a \le D_a$ and $d_s \le D_s$, where D_a and D_s denote the maximum action and state dimensions. The raw embodiment ID is mapped to a compact category slot, which indexes category-specific linear projections for action and state tokens. The action prediction head is category-specific as well. As a result, the model shares temporal and visual reasoning across embodiments while preserving the numerical action and state spaces of each robot embodiment.

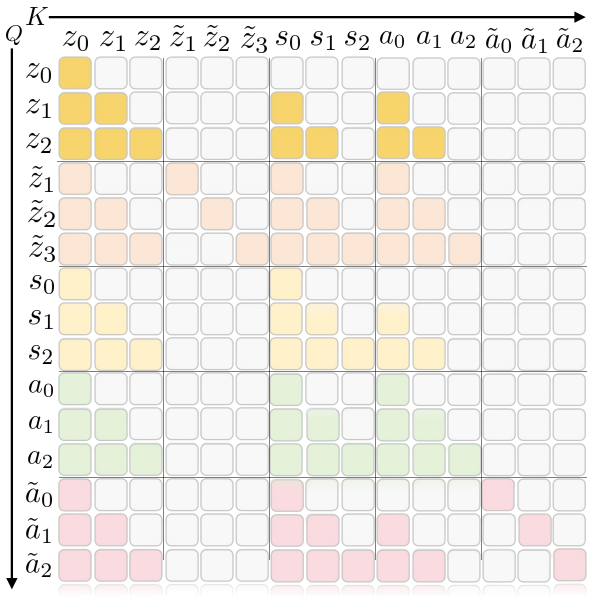

Figure 5: Teacher Forcing Attention Mask.

Token sequence and temporal alignment. For each training example, Riemann-1.0 serializes all modalities into a single transformer sequence: [context latent, clean latent, noisy latent, state, clean action, noisy action]. The context latent corresponds to the first observed frame. Clean latent and clean action tokens serve as teacher-forced history under the causal mask, while noisy latent and noisy action tokens are the flow-matching targets to be denoised. The action stream is temporally denser than the visual latent stream.

Fully causal training objective. Each visual latent summarizes a short segment of low-level control steps, determined by the VAE temporal stride and the frame sampling rate. During training, the action sequence is grouped by latent so that every predicted visual latent is paired with the corresponding chunk of robot actions. The state sequence contains one more element than the latent sequence, namely the initial state and the state observed after each latent transition. This organization aligns the training supervision with the causal rollout performed during inference.

To match online execution, each prediction should not access information unavailable at inference time. Riemann-1.0 therefore employs a structured attention mask instead of unrestricted self-attention. Clean tokens attend to previously observed clean tokens under the causal mask. Within a local generation block, bidirectional attention can be used to model intra-block structure. Noisy target tokens attend to preceding clean tokens and to other tokens within the same noisy block, but not to future clean tokens. This prevents leakage from future observations while preserving teacher forcing over the observed trajectory prefix.

The attention mask is constructed from three token attributes: a frame ID, a time ID, and a clean/noisy flag. As shown in Figure 5, clean tokens attend only to clean tokens from the current or preceding time steps. Noisy tokens attend to clean tokens from preceding time steps and to noisy tokens within the same frame. Padding tokens, including text padding tokens, are masked out to ensure stable training with variable-length sequences.

Latent and action flow-matching losses. Riemann-1.0 applies the flow-matching objective to two prediction heads. The visual-latent head predicts the velocity of noisy latent tokens:

$$\mathcal{L}_z = \mathbb{E}_{z,\epsilon,\sigma}\left[\left\| v_\theta^z(z_\sigma, \sigma, z_{<t}, s_{<t}, a_{\leq t}, c) - (\epsilon_z - z)\right\|_2^2\right]. \tag{2}$$

The action head predicts the velocity of noisy action tokens:

$$\mathcal{L}_a = \mathbb{E}_{a,\epsilon,\sigma}\left[\left\| v_\theta^a(a_\sigma, \sigma, z_{<t}, s_{<t}, a_{<t}, c) - (\epsilon_a - a)\right\|_2^2\right]. \tag{3}$$

The final objective balances visual dynamics and action prediction:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_z + \lambda\mathcal{L}_a, \tag{4}$$

where λ controls the relative weight of the action objective.

Loss masking. Separate validity masks are applied to the latent and action losses. The latent mask excludes padded frames and invalid latents at episode boundaries. The action mask excludes padded action channels, padded frames, and invalid low-level action steps, with the loss normalized only over valid channel-step entries rather than the entire padded tensor. This masking strategy is essential for multi-embodiment training, where different robots occupy different subsets of the shared padded action space.

The same validity information is applied during noise injection. Inactive padded regions are kept at zero for the clean sample, injected noise, noisy sample, and velocity target. This avoids a train-test mismatch in which padded channels would contain random Gaussian noise during training but zeros during deployment.

Autoregressive online inference. The training attention mask naturally enables online autoregressive inference with a growing KV cache. At the start of an episode, the first packed multi-view frame is encoded into z_0, and the initial state s_0 is embedded as a clean state token. Together with the text embedding, these tokens initialize the cache. The autoregressive online inference process is illustrated in Figure 6.

At denoising step t, the model samples a Gaussian noisy action tensor and denoises it using the flow scheduler while attending to the cached clean history and text condition. The resulting clean action chunk a_t is appended to the cache and executed for apf low-level control steps. The environment then returns new multi-view observations corresponding to the executed actions and the final state. The observations then is packed, encoded into z_t, and appended together with s_t to the cache as clean context for the next step.

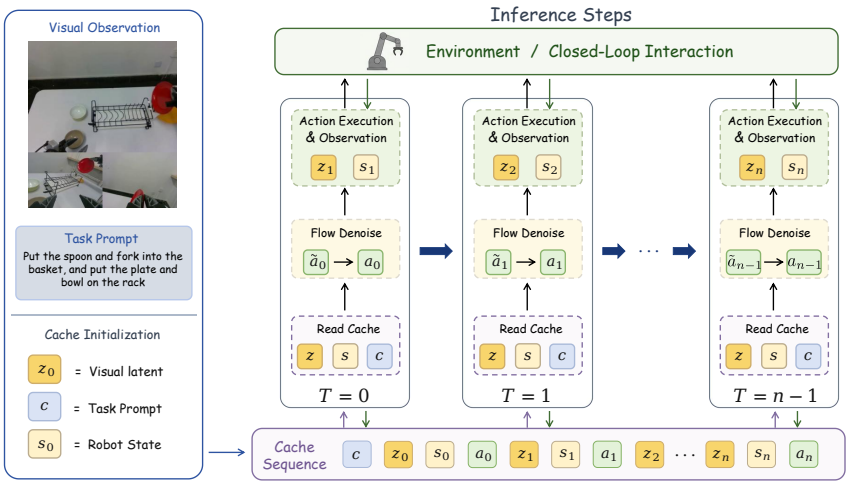

Figure 6: Autoregressive online inference of Riemann-1.0.

This procedure follows Eq. 1, where action generation precedes the corresponding visual latent update. During real-robot execution, the next latent is obtained by encoding the actual observation returned by the environment. During simulation, the same causal context together with the generated action chunk is fed to the latent head, and the predicted visual latent is decoded as the simulated camera observation. For long-horizon rollouts, the cache is maintained within a sliding temporal window. Once the window is full, the cache is reset and the most recent observation becomes the new context frame, enabling continuous online control while preserving causal consistency.

4. Training Recipe

Rima-WAM is trained with a three-stage curriculum that progressively shifts supervision from video-only pseudo actions to real action trajectories. This design is motivated by a practical data imbalance: video-only data is much larger and visually diverse, but it lacks executable robot actions, whereas real action datasets are smaller but provide direct supervision over robot behavior. The first stage uses a frozen Latent Action Model (LAM) to train large-scale unlabeled videos with pseudo-action supervision. The second stage grounds the model on a mixed corpus of UMI demonstrations, robot trajectories, and human videos with 3D hand annotations. The third stage continues on high-quality robot-only data to sharpen executable control. Across all stages, the model optimizes the same action-video objective,

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_z + \lambda\mathcal{L}_a,$$

where $\mathcal{L}_z$ is the visual-latent velocity loss and $\mathcal{L}_a$ is the action velocity loss. The action weight is increased from $\lambda = 0.1$ to 0.5 and then to 0.9, so the recipe first teaches broad visual dynamics with LAM-derived latent actions, then aligns the model with real multi-embodiment trajectories, and finally specializes it for robot execution. The overall three-stage pretraining framework is illustrated in Figure 7.

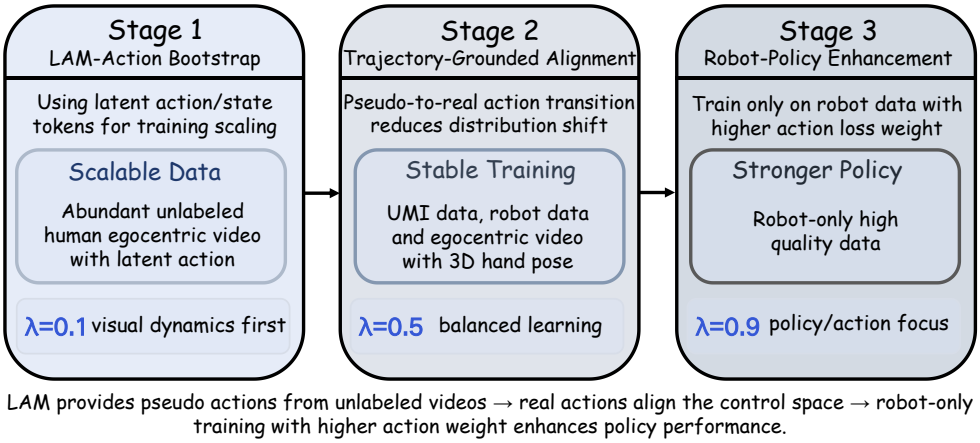

Figure 7: Three-stage WAM pretraining curriculum, from LAM-Action Bootstrap to Trajectory-Grounded Alignment and Robot-Policy Enhancement.

4.1. Pretraining

Stage I: LAM-Action Bootstrap. The first stage uses only unlabeled human manipulation videos. Since these videos provide rich visual interaction but lack executable action labels, we first train a Latent Action Model (LAM) (Routray et al., 2025) and then freeze it as a pseudo-action annotator. This stage aims to expose WAM to a large amount of manipulation dynamics before real robot action labels becomes available. LAM provides this bridge by forcing the motion between two frames to pass through a compact latent bottleneck, so that the inferred latent code captures the underlying visual transition that an action would have induced and can serve as pseudo action supervision for WAM training.

LAM is trained as a latent-action VAE over pairs of adjacent video frames. Given two frames (x_t, x_{t+1}), the encoder patchifies each frames into 16×16 image patches and prepends a learnable action-prompt token to the token sequence of each frame. A spatio-temporal transformer processes the two-frame token sequence, allowing the prompt token at the future frame to attend to both the current visual state and the resulting visual transition. The hidden state of this future-frame action-prompt token is projected to Gaussian moments $(\mu, \log \sigma^2)$, defining a posterior distribution over a 32-dimensional latent action space. A latent action is sampled with the reparameterization trick and injected into a decoder, which receives the current-frame visual patches and reconstructs the next frame. This architecture prevents the decoder from observing x_{t+1} directly; to minimize reconstruction error, the latent bottleneck must encode the transition information needed to explain how x_t evolves into x_{t+1}. The LAM objective combines next-frame reconstruction and a weak KL regularizer:

$$\mathcal{L}_{\text{LAM}} = \|\hat{x}_{t+1} - x_{t+1}\|_2^2 + \beta D_{\text{KL}}\big(q_\phi(z_t \mid x_t, x_{t+1}) \,\|\, \mathcal{N}(0, I)\big)\,.$$

The KL term is deliberately weak: its role is to regularize the latent distribution without collapsing the transition code into an overly generic prior. After training, we use the posterior mean μ rather than a sampled latent as the deterministic pseudo action, which eliminating sampling noise when annotating large-scale video corpora.

During WAM pretraining, the frozen LAM is applied densely over each video window. For every pair of adjacent frames in the dense stream, LAM predicts one pseudo transition code. These codes are then grouped

into action chunks that align with the WAM temporal interface: each visual latent corresponds to a short chunk of low-level action-like transitions. In parallel, the same clip is temporally subsampled and encoded by the Wan VAE into visual latents, with the first latent serving as context and the remaining latents serving as future visual targets. Thus Stage I presents WAM with the same structured inputs it will see later with real robot data, namely visual latents, state-like tokens, and dense action-like chunks, even though the original videos contain no action annotations. This stage is not intended to learn a deployable robot policy; with $\lambda = 0.1$, it mainly initializes the visual dynamics backbone and teaches the model how action-like motion tokens correlate with future visual change.

Stage II: trajectory-grounded alignment. The second stage replaces pseudo actions with action trajectories from a mixed multi-embodiment corpus. The data includes UMI demonstrations, robot trajectories, and human videos annotated with 3D hand poses and keypoints. These sources provide more reliable action supervision absent from video-only data while still covering diverse viewpoints, objects, tasks, and embodiments. Each trajectory is converted to the same WAM temporal interface: visual frames are packed according to the embodiment's camera layout, resized to the model resolution, and encoded into VAE latents; actions are grouped into dense chunks aligned with each visual latent; and states consist of the initial state plus one state after every latent transition. For heterogeneous robots, action and state vectors are mapped to an embodiment-specific canonical order, normalized with per-embodiment statistics, and padded into the shared tensor format used by the model. This stage uses $\lambda = 0.5$, making action prediction a central training signal while retaining visual-latent prediction as a dynamics regularizer. Its goal is to ground the autoregressive action-video interface in real trajectory supervision and align the shared backbone across different camera layouts, action spaces, and robot bodies. Compared with Stage I, the supervision is no longer only a latent explanation of visual motion; it now includes trajectories that can be interpreted as end-effector motion, gripper commands, joint commands, or hand-pose changes depending on the embodiment. This makes the action stream increasingly policy-relevant while preserving the broad visual dynamics learned from video-only data.

Stage III: robot policy enhancement. The final pretraining stage uses only high-quality robot demonstrations with real action-state supervision. Human video data, LAM pseudo actions, UMI trajectories, and 3D hand-pose human trajectories are excluded. This stage sharpens the model's executable action distribution after the previous stages have established broad visual dynamics and cross-embodiment trajectory alignment. We increase the action weight to $\lambda = 0.9$, so optimization focuses primarily on state-conditioned action prediction and trajectory consistency, while the latent objective continues to regularize the learned dynamics. The resulting model preserves the action-conditioned visual prediction ability of WAM, but its policy action head is more strongly biased toward robot-executable behavior. In practice, this stage reduces the mismatch between broad pretraining data and downstream robot deployment: the model has already learned how actions and observations interact, and the robot-only stage calibrates this prior toward real actuator conventions, contact-rich execution, and policy stability.

4.2. Post-training

After pretraining, we post-train Riemann-1.0 to adapt the general manipulation prior to downstream tasks. The post-training setting includes both real-world deployment tasks and simulation benchmarks. For real-

world evaluation, we collect teleoperation data on four tasks. In *ordered cube stacking*, the robot is required to stack four colored cubes in a specified color order, testing precise manipulation and instruction following. In *clothes folding*, the robot must manipulate a deformable object, testing whether the pretrained dynamics prior can adapt to non-rigid interaction. In *desk organization*, the robot tidies a table containing tissues, toys, chargers, tape, blocks, and pens, which requires long-horizon sequencing and fine-grained object handling. In *kitchen organization*, the robot places spoons, forks, bowls, and plates onto a kitchen rack, testing long-horizon manipulation in a cluttered household scene. For each real-world task, we collect three hours of teleoperated demonstrations. Instead of training task-specialized individual models, we aggregate all collected demonstration data to jointly fine-tune a single generalist model capable of executing all tasks. In addition, compared with the pretraining stage, we increase the action loss weight to 0.95 in post-training, which improves action execution performance.

5. Experiments

5.1. Real-World Experiments

5.1.1. Post-Training Data and Setup

After large-scale pretraining, we further adapt Riemann-1.0 to real-world deployment through real-robot post-training. We build four representative manipulation scenarios on the Tianji Marvin dual-arm robot: ordered cube stacking, clothes folding, desk organization, and kitchen organization. These tasks cover order-constrained rigid-object stacking, deformable-object manipulation, long-horizon tabletop rearrangement, and long-horizon kitchen storage, forming a compact but challenging suite for household robot deployment. We collect 15 demonstrations for each task using human teleoperation.

In **ordered cube stacking**, the robot follows a language instruction to identify colored cubes and assemble a four-layer stack following a specified color order. Four-layer stacking sharply increases the difficulty: small alignment errors at lower layers are amplified by later placements, and a single misaligned layer can destabilize the entire stack. Therefore, this task requires accurate visual recognition of object color and pose, planning of the grasping sequence, and precise pick-and-place execution.

In **clothes folding**, the robot manipulates a deformable garment rather than a rigid object. Clothes can wrinkle, stretch, and drift in pose during contact, so the robot must continuously adapt its action strategy based on the observed state. This task stresses deformable-object perception, bimanual coordination, and fine-grained contact control.

In **desk organization**, the robot must identify and rearrange multiple scattered desktop objects into their designated locations. A key challenge is inserting slender objects such as pens into a pen holder. This requires accurate perception of the pen pose and the holder opening, exploiting the rim of the holder to reorient the pen toward a vertical pose, and then controlling the gripper to complete a precise insertion.

In **kitchen organization**, the robot must complete an end-to-end tidying task in a kitchen scene. It first localizes and grasps utensils such as forks and spoons, then places them back into the target storage area. It must also grasp yellow and red plates and insert them vertically into a drying rack, stack bowls, and place the bowl stack on the upper layer of the rack, requiring long-horizon sequencing and stable placement under cluttered conditions.

We compare Riemann-1.0 with representative open-source VLA and WAM baselines, including DreamZero,

Table 1: Real-world manipulation results. SR and PSR are reported in percent. DreamZero* denotes our reproduced implementation based on the official open-source code, trained with our large-scale manipulation dataset.

Model	Ordered cube stacking		Kitchen organization		Clothes folding		Desk organization		Avg.	
	SR	PSR	SR	PSR	SR	PSR	SR	PSR	SR	PSR
DreamZero* (Ye et al., 2026b)	15.0	20.0	15.0	16.6	15.0	12.5	15.0	33.3	15.00	20.60
τ_0-WM (Zhou et al., 2026)	15.0	26.2	15.0	20.0	15.0	24.6	20.0	57.2	16.25	32.00
LingBot-VLA (Wu et al., 2026)	15.0	31.5	20.0	80.2	80.0	88.4	20.0	76.9	33.75	69.25
$\pi_{0.5}$ (Intelligence et al., 2025)	40.0	59.0	20.0	37.2	45.0	73.5	40.0	73.1	36.25	60.70
LingBot-VA (Li et al., 2026)	40.0	66.6	20.0	30.0	80.0	85.0	40.0	80.1	45.00	65.43
G0.5 (Galaxea Team, 2026)	80.0	89.5	35.0	47.8	**85.0**	90.0	**80.0**	93.4	70.00	80.18
Riemann-1.0 (Ours)	**85.0**	**91.6**	**90.0**	**98.4**	**85.0**	**92.5**	**80.0**	**95.2**	**85.00**	**94.43**

τ_0-WM, LingBot-VLA, $\pi_{0.5}$, LingBot-VA, and G0.5, under the same real-world manipulation setting. For DreamZero, we reproduce the model using the official open-source implementation, training a 5B-scale model on our collected large-scale manipulation dataset, denoted as DreamZero*.

5.1.2. Real-World Evaluation

We evaluate real-robot performance using two metrics. Success Rate (SR) measures whether the final task goal is completed. Progress Success Rate (PSR) measures the fraction of task progress completed according to task-specific intermediate milestones, which is especially important for long-horizon organization tasks where partial completion is meaningful even when the final state is not fully reached.

As shown in Table 1, Riemann-1.0 achieves the best average performance among all compared models, with an average SR of 85.00% and an average PSR of 94.43%. It maintains at least 80.0% SR on all four tasks and exceeds 91.0% PSR on every task, indicating that the model not only reaches the final goal more reliably but also makes stable progress through intermediate steps. These results suggest that post-training preserves the general manipulation prior learned during pretraining while adapting it to the target real-world robot, camera layout, and household task distribution.

Specifically, on ordered cube stacking, Riemann-1.0 reaches 85.0% SR and 91.6% PSR, improving over the strongest baseline G0.5 by 5.0 SR points and 2.1 PSR points. On kitchen organization, Riemann-1.0 obtains 90.0% SR, while the next best SR is 35.0% from G0.5; it also improves PSR over the strongest baseline LingBot-VLA by 18.2 points, indicating more reliable completion of long-horizon tableware storage. On clothes folding, Riemann-1.0 ties the best SR at 85.0% with G0.5 but achieves the highest PSR at 92.5%, suggesting better intermediate progress on deformable-object manipulation. On desk organization, Riemann-1.0 ties the best SR at 80.0% with G0.5 and further improves PSR from 93.4% to 95.2%, reflecting more stable execution when inserting slender objects and arranging cluttered desktop items. The execution process of the robot on these tasks is illustrated in Figure 8.

5.1.3. Compositional Generalization and Out-of-Domain Evaluation

Beyond evaluating the four tasks used for real-world post-training, we further test whether the deployed policy can reuse its learned manipulation skills under held-out instructions. As shown in Figure 9, we consider two setting.

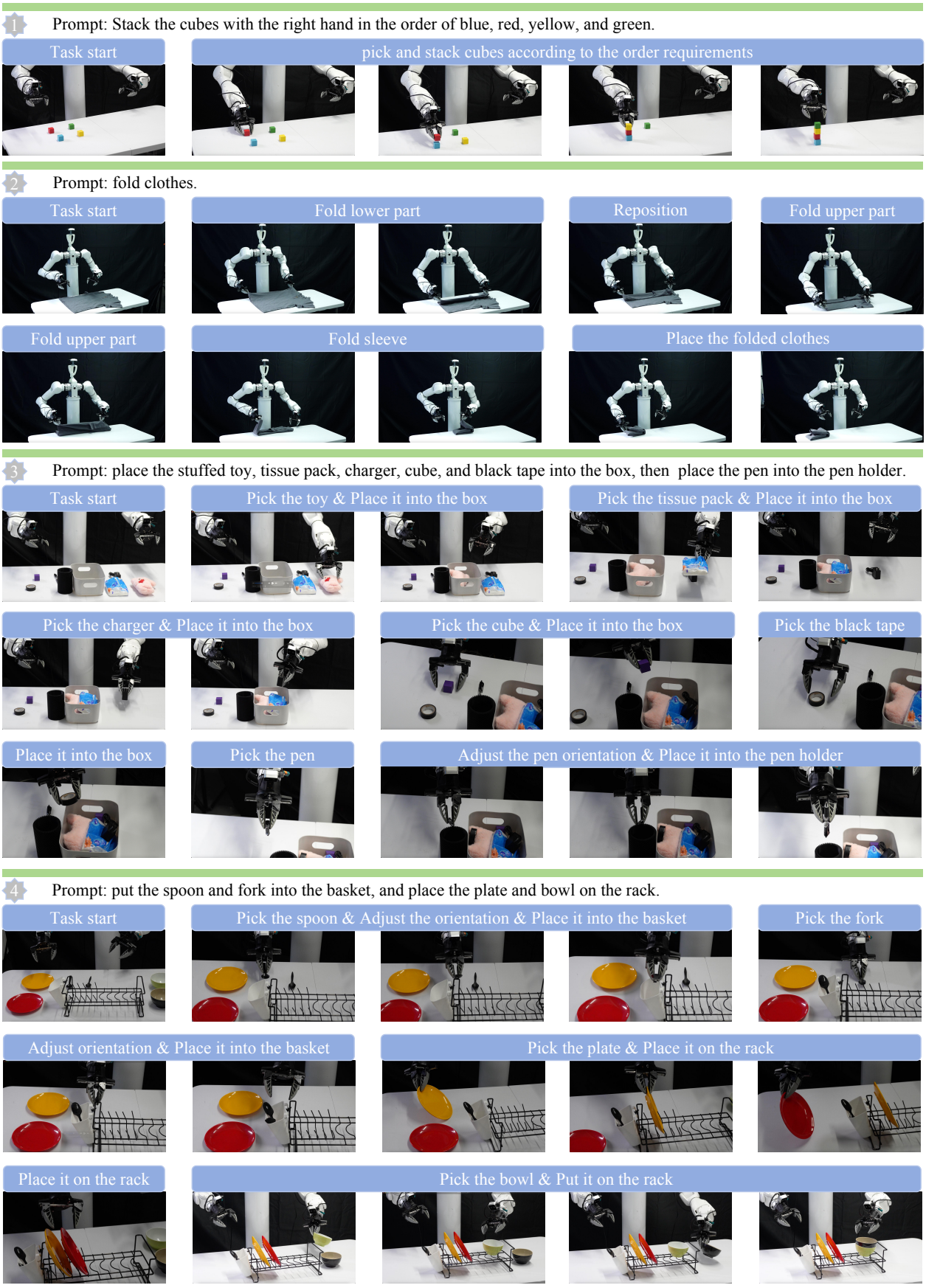

Figure 8: Robot execution processes for real-world tasks; clothes folding shows a single folding sequence.

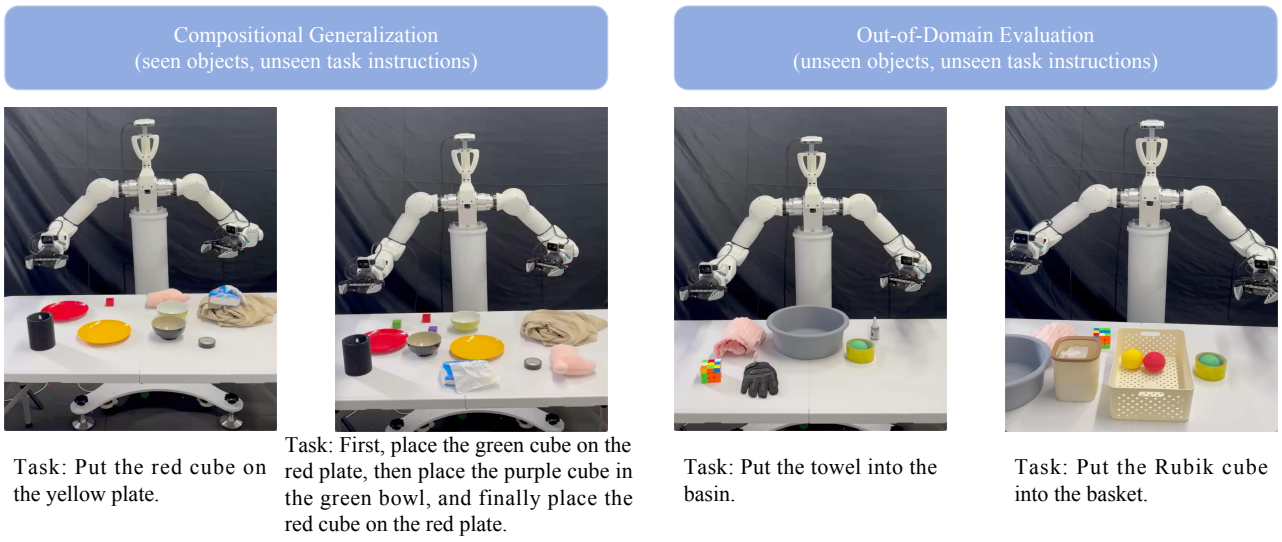

Figure 9: Illustration of compositional generalization and OOD tasks.

Table 2: Real-world compositional generalization and OOD results. Task rows report successful trials out of 10.

Setting	Task	Riemann-1.0 (Ours)	$\pi_{0.5}$	LingBot-VA
Compositional	Cube-to-bowl/plate placement	**8/10 (80.0)**	6/10 (60.0)	3/10 (30.0)
	+ color-order constraint	**5/10 (50.0)**	3/10 (30.0)	0/10 (0.0)
	Average	**65.0**	45.0	15.0
OOD	Rubik's cube to storage box	**10/10 (100.0)**	5/10 (50.0)	3/10 (30.0)
	Towel-to-basin placement	**7/10 (70.0)**	5/10 (50.0)	3/10 (30.0)
	Average	**85.0**	50.0	30.0
Overall average		**75.0**	47.5	22.5

Compositional generalization uses seen object categories but recombines them into unseen task goals. In the first task, the robot is instructed to grasp a specified cube and place it into a target receptacle, such as a bowl or a plate. This setting keeps the object category familiar while changing the object-receptacle relation. In the second task, the instruction further specifies object color, receptacle color, receptacle type, and execution order, e.g., *First, place the green cube on the red plate, then place the purple cube in the green bowl, and finally place the red cube on the red plate*.

Out-of-Domain (OOD) evaluation uses unseen scenes, objects, and task goals that are not included in the real-world post-training set. No additional demonstrations or parameter updates are provided for these tasks. The task definitions are: placing a Rubik's cube into a storage box and placing a towel into a basin. Since each task has a binary task completion criterion, we report Success Rate (SR) over 10 real-robot trials per task.

As shown in Table 2, Riemann-1.0 achieves the highest SR in both held-out settings. In compositional generalization, it succeeds in 8 out of 10 trials when placing cubes into a bowl or plate. When the instruction additionally specifies object color, receptacle color, receptacle type, and execution order, the task becomes

harder because the policy must jointly solve visual grounding and multi-step sequencing. Riemann-1.0 still reaches 50.0% SR under this stricter setting, leading to an average SR of 65.0% across the two compositional tasks.

In the OOD setting, Riemann-1.0 obtains an average SR of 85.0%. The model completes all Rubik's-cube-to-storage-box trials and retains 70.0% SR on towel-to-basin placement, showing that the policy can execute both rigid-object storage and deformable-object placement under held-out scenes and instructions. Across all four held-out tasks, Riemann-1.0 obtains an overall average SR of 75.0%. These results suggest that the pretrained manipulation prior and real-world post-training do not merely fit the training tasks, but transfer to new task compositions and out-of-distribution household scenarios.

5.2. Simulation Evaluation

We evaluate Riemann-1.0 on three simulation benchmarks: RoboCasa365, RoboTwin 2.0, and LIBERO. Following the standard protocol of each benchmark, we report task success rate as the primary metric. We keep the same WAM temporal interface across benchmarks: one visual latent corresponds to 16 low-level action steps.

5.2.1. RoboCasa365

For RoboCasa-365 (Nasiriany et al., 2026), we first pretrain on 300 tasks and then finetune on the Target-50 split. The 50 target tasks evaluation contains Atomic-Seen, Composite-Seen, and Composite-Unseen groups.

Table 3: Evaluation results on RoboCasa365 50 target tasks.

Model	Atomic-Seen	Composite-Seen	Composite-Unseen	Average
GR00T-N1.5 (Bjorck et al., 2025a)	60.6	35.0	33.3	43.7
Fast-WAM (Yuan et al., 2026)	59.1	36.4	33.2	43.5
LingBot-VA (Li et al., 2026)	63.5	37.3	32.1	45.1
ABot-M0.5 (Chen et al., 2026)	70.6	44.3	45.6	54.2
Riemann-1.0 (Ours)	**74.2**	**56.0**	**56.3**	**62.6**

As shown in Table 3, Riemann-1.0 achieves the best performance across all three RoboCasa365 categories, improving the average success rate from 54.2% to 62.6% over the strongest baseline, ABot-M0.5. Notably, Riemann-1.0 improves the performance on Composite-Seen and Composite-Unseen by 11.7 and 10.7 percentage points, respectively, demonstrating its strong compositional generalization capability and ability to transfer learned manipulation skills to unseen tasks.

5.2.2. RoboTwin 2.0

For RoboTwin 2.0 (Chen et al., 2025a), we train on 50 bimanual tasks. The training set contains clean scenes with 50 demonstrations per task, plus 25,000 demonstrations from heavily randomized scenes, i.e., 500 randomized demonstrations per task. Evaluation is reported under Clean and Randomized settings to measure both nominal performance and robustness to visual and physical perturbations. As reported in Table 4, Riemann-1.0 achieves the highest average success rate on RoboTwin 2.0, reaching average 94.3%

Table 4: Evaluation results on the RoboTwin 2.0 benchmark.

Model	Clean	Randomized	Average
$\pi_{0.5}$ (Intelligence et al., 2025)	82.7	76.8	79.8
ABot-M0 (Yang et al., 2026)	86.1	85.1	85.6
LingBot-VLA (Wu et al., 2026)	86.5	85.3	85.9
Qwen-VLA (Wang et al., 2026b)	86.1	87.2	86.7
Motus (Bi et al., 2026)	88.7	87.0	87.8
Fast-WAM (Yuan et al., 2026)	91.9	91.8	91.8
LingBot-VA (Li et al., 2026)	92.9	91.6	92.2
G0.5 (Galaxea Team, 2026)	93.7	92.8	93.3
ABot-M0.5 (Chen et al., 2026)	94.0	**94.2**	94.1
Riemann-1.0 (Ours)	**94.6**	94.0	**94.3**

across 50 bimanual tasks.

Table 5: Evaluation results on the LIBERO benchmark.

Method	Spatial	Object	Goal	Long	Average
π_0-Fast (Pertsch et al., 2025)	96.4	96.8	88.6	60.2	85.5
GR00T-N1 (Bjorck et al., 2025b)	94.4	97.6	93.0	90.6	93.9
π_0 (Black et al., 2024)	98.0	96.8	94.4	88.4	94.4
InternVLA-M1 (Chen et al., 2025b)	98.0	99.0	93.8	92.6	95.9
$\pi_{0.5}$ (Intelligence et al., 2025)	98.8	98.2	98.0	92.4	96.9
GR00T-N1.6 (Bjorck et al., 2025b)	97.7	98.5	97.5	94.4	97.0
OpenVLA-OFT (Kim et al., 2025)	97.6	98.4	97.9	94.5	97.1
Fast-WAM (Yuan et al., 2026)	98.2	**100.0**	97.0	95.2	97.6
Motus (Bi et al., 2026)	96.8	99.8	96.6	97.6	97.7
LingBot-VA (Li et al., 2026)	98.5	99.6	97.2	98.5	98.5
ABot-M0 (Yang et al., 2026)	98.8	99.8	99.0	96.6	98.6
Being-H0.5 (Luo et al., 2026)	99.2	99.6	**99.4**	97.4	98.9
G0.5 (Galaxea Team, 2026)	98.4	**100.0**	98.6	**98.6**	98.9
Riemann-1.0 (Ours)	**99.6**	**100.0**	97.6	**98.6**	**99.0**

5.2.3. LIBERO

For LIBERO (Liu et al., 2023), we train on four official suites: Spatial, Object, Goal, and Long. Each suite contains 10 tasks, and each task provides 50 demonstrations.

On LIBERO, as shown in Table 5, Riemann-1.0 reaches the overall average of 99.0%. Specifically, it achieves 99.6% on Spatial, 100.0% on Object, 97.6% on Goal, and 98.6% on Long.

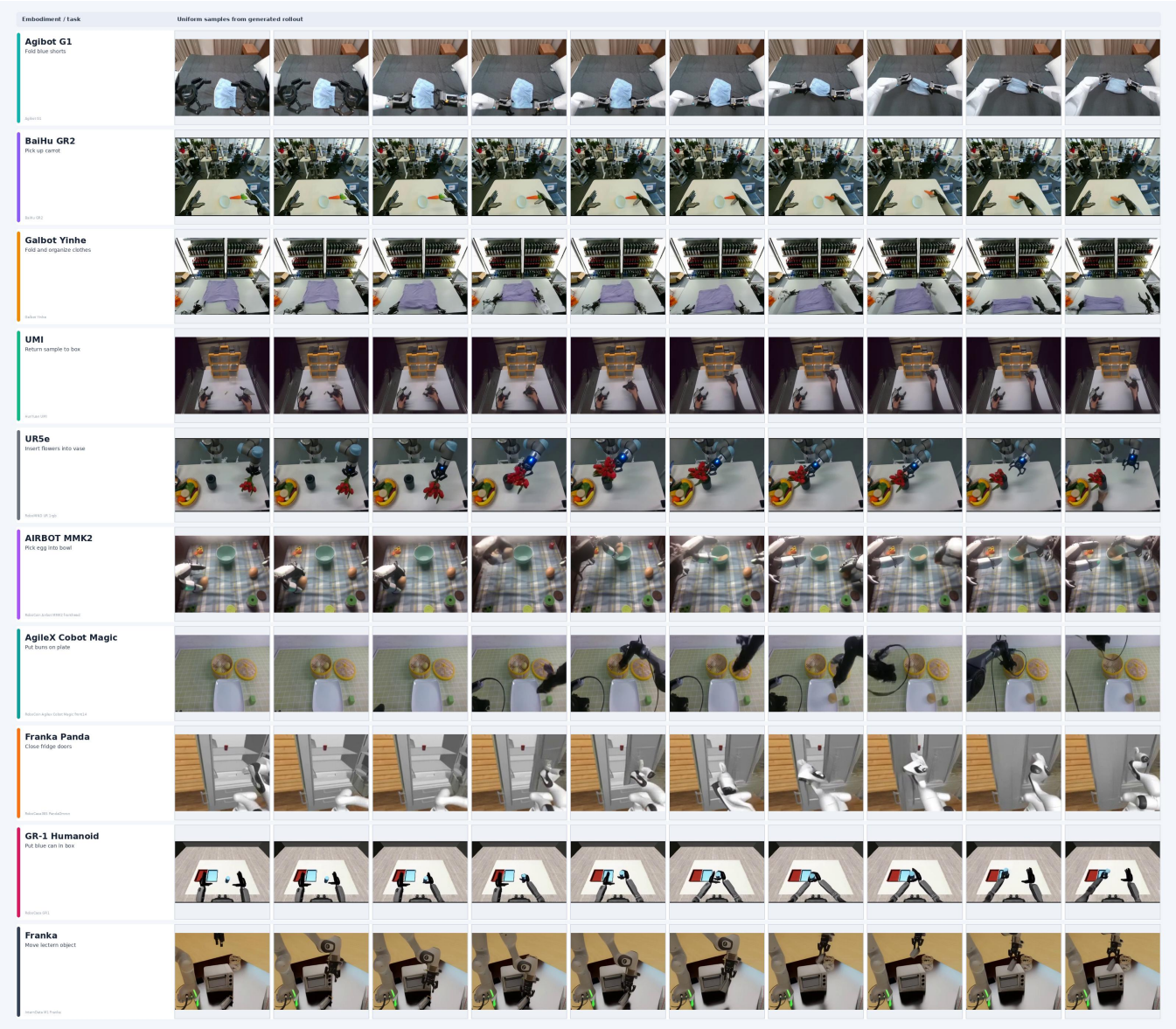

Figure 10: Multi-embodiment action-conditioned visual rollout. Generated multi-embodiment rollouts from Riemann-1.0. Each row presents a rollout for a different robot embodiment with an abbreviated task description, demonstrating its capability as a data-driven simulator across diverse robots, viewpoints, and manipulation tasks.

5.3. Action-Conditioned Visual Rollout as a Multi-Embodiment Simulator

Riemann-1.0 can also be used as an action-conditioned visual simulator. In this mode, the model is not asked to only output an executable action. Instead, it receives the current visual observation, task prompt, robot state, and a candidate future action trajectory, and predicts the visual consequences of executing the trajectory. The action trajectory is represented with the same embodiment-specific action interface used during policy training, then embedded as action tokens and inserted into the causal action-video sequence. These action tokens condition the visual-latent denoising process, so the generated frames are tied to the robot motion rather than sampled as an unconditional future video.

At inference time, we first encode the current observation into the VAE latent space and use it as the clean visual context. The future action chunk is either provided by the policy head, sampled from a candidate plan, or taken from a recorded trajectory for analysis. Conditioned on the prompt, state, clean visual context, and action tokens, the latent head rolls out future visual latents; these latents are then decoded back into RGB video. For longer horizons, the generated or observed visual output can be appended back to the context and the same procedure can be repeated autoregressively. This gives Riemann-1.0 a data-driven simulator interface: actions are treated as controllable inputs, while future camera observations are produced as the simulated consequences.

Figure 10 shows qualitative rollouts across heterogeneous embodiments and camera layouts. The examples cover humanoid robots, dual-arm systems, single-arm platforms, dexterous-hand robots, and simulation benchmarks such as RoboTwin (Mu et al., 2025) and RoboCasa-GR1 (Bjorck et al., 2025a). Despite the differences in viewpoint, embodiment geometry, gripper appearance, and task type, the generated videos preserve the main scene structure and exhibit action-consistent object and robot motion. These results suggest that Riemann-1.0 learns a shared action-conditioned visual dynamics prior that can generalize beyond a single robot body, making it useful for rollout inspection, action-plan comparison, and future-state imagination.

6. Related Work

Vision-language-action policies. Large vision-language-action (VLA) policies have become a central paradigm for general robot manipulation. RT-1 and RT-2 demonstrate that transformer-based policies can benefit from broad robot demonstrations and web-scale semantic pretraining (Brohan et al., 2022, 2023), while OpenVLA, π_0, GR00T-N1, and LingBot-VLA further scale open backbones, continuous action prediction, cross-embodiment data, and system-level robot learning pipelines (Kim et al., 2024, **?**, Bjorck et al., 2025b, Wu et al., 2026). These models are effective executable policies, but they are usually optimized as observation-to-action predictors rather than explicit predictors of the visual consequences of actions. At the same time, scaling embodied learning requires heterogeneous data sources. Egocentric human video datasets provide large-scale hand-object interactions and long-horizon activity structure (Damen et al., 2020, Grauman et al., 2023, Wang et al., 2023, Liu et al., 2022, 2024, Hoque et al., 2025), UMI and exoskeleton-glove demonstrations provide manipulation-centric views and device states closer to robot control (Chi et al., 2024, Tao et al., 2025), and robot trajectory datasets provide direct action-state supervision but are fragmented across embodiments, camera layouts, and control conventions (Open X-Embodiment Collaboration, 2023, Wu et al., 2024, AgiBot World Team, 2025, Intern Robotics, 2025, Galaxea AI, 2025, RoboCOIN Team, 2026). Riemann-1.0 is designed to use these sources jointly: human videos contribute scalable interaction knowledge, UMI-style data bridges human manipulation and robot-compatible control, and robot trajectories provide embodiment-specific executable supervision.

World action models. World models learn predictive dynamics for planning, evaluation, or policy learning; in robot manipulation, this direction has recently evolved into World Action Models (WAMs), which jointly model robot actions and future observations. Existing WAMs differ mainly in how they order action and video generation. DreamZero-style methods couple action and video through joint denoising (Ye et al., 2026b); LingBot-VA emphasizes native video-action pretraining on top of video generative models (Li et al., 2026); FastWAM studies the efficiency trade-off of using video modeling during training while keeping

inference action-centered (Yuan et al., 2026); and GIGA-World-Policy distinguishes future-visual supervision during training from efficient action decoding during inference (Ye et al., 2026a). Related controllable world-modeling systems also study action-conditioned video rollout and future-aware policy evaluation (Guo et al., 2025, Zhou et al., 2026). Beyond architecture, recent robot learning systems increasingly rely on staged training recipes, combining semantic pretraining, robot imitation learning, action-space alignment, data filtering, balancing, normalization, and downstream fine-tuning (Kim et al., 2024, **?**, Bjorck et al., 2025b, Wu et al., 2026, Open X-Embodiment Collaboration, 2023, Wu et al., 2024, AgiBot World Team, 2025, Kuramshin et al., 2026). Riemann-1.0 follows this trend but adapts it to WAM pretraining: it first learns from unlabeled human videos using LAM-derived pseudo actions, then grounds the action-video interface with mixed UMI, robot, and 3D hand-pose trajectories, and finally strengthens executable behavior with robot-only policy enhancement.

Policies and action-conditioned simulators. Most deployed robot policies are optimized for fast action prediction, while model-based simulators are optimized for predicting future observations or state transitions. Classical model-based control separates these roles, but video-based robot learning has increasingly blurred the boundary by using video prediction for planning, future representation learning, and policy supervision (Du et al., 2023a,b, Bruce et al., 2024, Cheang et al., 2024, Hu et al., 2024, Liang et al., 2025, Shen et al., 2025). The key requirement for a useful robot simulator is action conditioning: generated future observations should respond to the robot's proposed motion rather than extrapolating a likely video without control input. Riemann-1.0 addresses this requirement with a fully causal autoregressive formulation. Actions are predicted or provided before the corresponding visual latent, the action conditions the visual transition, and the resulting observation becomes context for the next step. Thus, when deployed as a policy, the model predicts the next action chunk from causal history and uses the real environment observation as the next context; when used as a simulator, the same action chunk conditions the latent head to generate the corresponding future visual observation. This dual interface connects executable control and model-based imagination within a single WAM.

7. Conclusion

We introduced Riemann-1.0, a fully causal autoregressive World Action Model for embodied intelligence. Riemann-1.0 addresses two central challenges in scaling robot foundation models: how to unify heterogeneous embodied experience across egocentric human videos, handheld-gripper demonstrations, and robot trajectories, and how to jointly model robot actions, states, and world evolution in a causal form aligned with real interaction. By combining a unified embodied data infrastructure with Progressive Embodied Pretraining over more than 200K+ hours of embodied experience, Riemann-1.0 transfers large-scale interaction knowledge from weak supervision to executable robot control. Across simulation and real-world evaluations, Riemann-1.0 achieves state-of-the-art performance, including 62.6% on RoboCasa365, 94.3% on RoboTwin 2.0, 99.0% on LIBERO, and 85.0% SR with 94.43% PSR on long-horizon real-world manipulation tasks. These results show that Riemann-1.0 provides a scalable path for transforming large-scale embodied experience into generalizable robot manipulation capabilities.

References

AgiBot World Team. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 35101–35113, 2026.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025a.

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

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.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. URL https://arxiv.org/abs/2212.06817.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. URL https://arxiv.org/abs/2307.15818.

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. *arXiv preprint arXiv:2402.15391*, 2024. URL https://arxiv.org/abs/2402.15391.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. URL https://arxiv.org/abs/2410.06158.

Ronghan Chen, Yandan Yang, Zuojin Tang, Dongjie Huo, Tong Lin, Haoning Wu, Haoyun Liu, Yuzhi Chen, Lulu Zheng, Botai Yuan, et al. Abot-m0. 5: Unified mobility-and-manipulation world action model. *arXiv preprint arXiv:2607.00678*, 2026.

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

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*, 2025b.

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

Raffel Colin. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21, 2020.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *arXiv preprint arXiv:2302.00111*, 2023a. URL https://arxiv.org/abs/2302.00111.

Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, et al. Video language planning. *arXiv preprint arXiv:2310.10625*, 2023b. URL https://arxiv.org/abs/2310.10625.

Galaxea AI. Galaxea open-world dataset. https://huggingface.co/datasets/GalaxeaAI/Galaxea-Open-World-Dataset, 2025. Dataset.

Galaxea Team. Galaxea g0.5 technical report. 2026. URL https://opengalaxea.github.io/G05/.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, et al. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives. *arXiv preprint arXiv:2311.18259*, 2023.

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. URL https://arxiv.org/abs/2510.10125.

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. URL https://arxiv.org/abs/2412.14803.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Intern Robotics. Interndata-a1-lerobot-v3.0-by-embodiment. https://huggingface.co/datasets/InternRobotics/InternData-A1-LeRobot-v3.0-by-embodiment, 2025. Dataset.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. URL https://arxiv.org/abs/2406.09246.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Artur Kuramshin, Özgür Aslan, Cyrus Neary, and Glen Berseth. Task robustness via re-labelling vision-action robot data. *arXiv preprint arXiv:2606.10918*, 2026.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026. URL https://arxiv.org/abs/2601.21998.

Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025. URL https://arxiv.org/abs/2508.00795.

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

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. *arXiv preprint arXiv:2401.08399*, 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. *arXiv preprint arXiv:2203.01577*, 2022.

Hao Luo, Ye Wang, Wanpeng Zhang, Sipeng Zheng, Ziheng Xi, Chaoyi Xu, Haiweng Xu, Haoqi Yuan, Chi Zhang, Yiqing Wang, et al. Being-h0. 5: Scaling human-centric robot learning for cross-embodiment generalization. *arXiv preprint arXiv:2601.12993*, 2026.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, et al. Robotwin: Dual-arm robot benchmark with generative digital twins. In *Proceedings of the computer vision and pattern recognition conference*, pages 27649–27660, 2025.

Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots. *arXiv preprint arXiv:2603.04356*, 2026.

Open X-Embodiment Collaboration. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

RoboCOIN Team. Robocoin: A comprehensive dataset for multi-platform bimanual robot manipulation. *arXiv preprint*, 2026.

Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*, 2022.

Sandeep Routray, Hengkai Pan, Unnat Jain, Shikhar Bahl, and Deepak Pathak. Vipra: Video prediction for robot actions. *arXiv preprint arXiv:2511.07732*, 2025.

Yichao Shen, Fangyun Wei, Zhiying Du, Yaobo Liang, Yan Lu, Jiaolong Yang, Nanning Zheng, and Baining Guo. VideoVLA: Video generators can be generalizable robot manipulators. *arXiv preprint arXiv:2512.06963*, 2025. URL https://arxiv.org/abs/2512.06963.

Stone Tao, Kenneth Zhou, Yixin Chen, Fangchen Wang, Kuan Fang, Wei Yang, Zhenjia Xu, Shuran Song, Pieter Abbeel, Jitendra Malik, and C. Karen Liu. Dexwild: Dexterous human interactions for in-the-wild robot policies. *arXiv preprint arXiv:2505.07813*, 2025.

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.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026a.

Qiuyue Wang, Mingsheng Li, Jian Guan, Jinhui Ye, Sicheng Xie, Yitao Liu, Junhao Chen, Zhixuan Liang, Jie Zhang, Xintong Hu, et al. Qwen-vla: Unifying vision-language-action modeling across tasks, environments, and robot embodiments. *arXiv preprint arXiv:2605.30280*, 2026b.

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. *arXiv preprint arXiv:2309.17024*, 2023.

Jun Wu, Ruyi Liu, Haoran He, Jiawei Li, Shiqi Cheng, Ting Zhang, Jingkai Zhang, Jialu Chen, Jiahang Chen, Yi Shen, Su Li, Yuxiang Zhou, Ziyu Lu, Yiran Wang, Yuxuan Kuang, Xiaojian Lei, Hengyi Chen, He Wang, Hao Zhu, and Hang Su. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, et al. LingBot-VLA: A pragmatic VLA foundation model. *arXiv preprint arXiv:2601.18692*, 2026. URL https://arxiv.org/abs/2601.18692.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, et al. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026.

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. *arXiv preprint arXiv:2603.17240*, 2026a. URL https://arxiv.org/abs/2603.17240.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026b.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026. URL https://arxiv.org/abs/2603.16666.

Pengfei Zhou, Shengcong Chen, Di Chen, Jiaxu Wang, Rongjun Jin, Bingwen Zhu, Yike Pan, Songen Gu, Kuanning Wang, Shufeng Nan, et al. τ_0-WM: A unified video-action world model for robotic manipulation. *arXiv preprint arXiv:2606.01027*, 2026. URL https://arxiv.org/abs/2606.01027.