A Latent World-Action Model with Jointly Aligned Reasoning

Hao Luo[*1,2] **Wanpeng Zhang**[*1,2] **Yicheng Feng**[*1,2] **Sipeng Zheng**[*2] **Haiweng Xu**[3,2]
Chaoyi Xu[1,2] **Ziheng Xi**[4,2] **Yuhui Fu**[1,2] **Zongqing Lu**[†1,2]
[1]School of Computer Science, Peking University [2]BeingBeyond
[3]School of Electronics Engineering and Computer Science, Peking University
[4]Department of Automation, Tsinghua University

Abstract

Despite recent progress, existing general robot policies, particularly Vision-Language-Action models, are still primarily trained to map observations directly to actions using sparse action supervision. As a result, they often learn shallow observation-to-action correlations instead of deeper representations of world dynamics, including object motion, physical interaction, and long-horizon task progression. Recent world-action models attempt to address this limitation through future-frame prediction, but pixel-space rollout is computationally expensive and poorly aligned with the abstractions required for control, forcing the model to reconstruct visual details irrelevant to action. We present **Being-H0.7**, a latent world-action model that introduces future-aware reasoning into VLA-style policies without generating future frames. Our model inserts learnable latent queries between perception and action, forming an explicit reasoning interface for action generation. To train this interface, we adopt a future-informed dual-branch design: a deployable prior branch infers latent states from the current context, while a training-only posterior branch replaces latent queries with embeddings from future frames. Joint alignment between the two branches enables the prior branch to learn predictive, action-relevant latent structure directly from current observations. Under a controlled training setup for comprehensive validation, Being-H0.7 achieves state-of-the-art or competitive performance across diverse benchmarks, while further demonstrating strong deployability on real-robot manipulation tasks.

1 Introduction

Generalist robotic policies are rapidly evolving from task-specific controllers into large embodied models capable of following instructions, perceiving diverse environments, and executing long-horizon tasks across robot embodiments. A dominant paradigm is the Vision-Language-Action model (VLA) [1, 2], which adapts pretrained MLLMs to map multimodal observations and language instructions directly to robot actions [3, 4]. Despite recent progress, this formulation remains fundamentally reactive and limited in generalization, particularly when successful control requires anticipating how the environment evolves through interaction. Because action supervision is sparse and highly correlated with demonstrations, VLAs can overfit to shortcut associations between visual cues and actions instead of learning structured representations of object dynamics, contact, affordances, and task progress.

World modeling offers a natural way to introduce future awareness into robot learning [5, 6, 7]. Recent world-action models [8, 9] often adopt an "image-then-act" paradigm, tightly coupling future visual prediction with action generation [10, 11, 12]. While intuitive, pixel-space prediction is an

*Equal contribution. †Corresponding author: zongqing.lu@pku.edu.cn

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

indirect substrate for manipulation: robots rarely need to reconstruct photorealistic future frames, and dense visual prediction can waste substantial capacity on appearance details irrelevant to control, while introducing significant training and inference overhead.

These limitations motivate a latent formulation of world-action modeling. Rather than explicitly predicting future frames, we argue that future observations should supervise latent future-relevant factors that are useful for action generation. The goal is to learn a compact, action-oriented reasoning space that captures predictive information for control while remaining tightly coupled to the policy's action pathway. Such a formulation preserves the efficiency and deployability of direct VLA-style policies while augmenting them with future-aware reasoning.

In this work, we present **Being-H0.7**, a latent world-action model with joint alignment. Being-H0.7 introduces a small set of learnable latent queries between multimodal context and action tokens, forming an explicit reasoning interface between perception and action. These queries attend to the instruction, observations, and robot states, summarize interaction-relevant information into a compact latent state, and condition subsequent action generation, enabling future-aware control without explicit frame prediction.

A central challenge is that such latent reasoning factors are not directly annotated. To address this, we propose a future-informed dual-branch formulation. To be specific, in our model, a deployable prior branch infers latent predictive states from current observations alone based on learnable latent queries. Meanwhile, a training-only posterior branch replaces the latent queries with embeddings extracted from future observations while keeping the action-generation pathway structurally matched. By jointly aligning the hidden states of the two branches at latent reasoning positions, the prior branch learns to produce posterior-like, action-useful representations without access to future observations. At inference time, the posterior branch is removed entirely, allowing Being-H0.7 to perform direct action prediction without costly future-frame rollout. Our contributions are three-fold:

- We reformulate world-action modeling as learning action-oriented latent predictive factors instead of reconstructing future pixels, and introduce Being-H0.7, which uses learnable latent queries as an explicit reasoning interface between perception and action.
- We propose a future-informed joint alignment objective that transfers privileged future information from subsequent observations to a deployable prior branch, together with lightweight regularization for stable latent learning.
- Across four simulation benchmarks, Being-H0.7 achieves state-of-the-art or competitive performance. Extensive ablations and real-world evaluations further show that it combines the predictive benefits of world models with the efficiency and deployability of direct VLA-style policies.

2 Related Work

Vision-Language-Action Model (VLA). The VLA [2, 13, 14, 15] have emerged as one of the most promising paradigms for general robotic manipulation. Built upon pretrained vision-language models (VLMs) [16, 17, 18], they enable direct action prediction from multimodal observations. A major line of progress lies in action-head design: early methods rely on autoregressive action tokenization [2, 19], while recent approaches increasingly adopt diffusion- or flow-based generators [13, 20, 3], improving precision and efficiency for complex control. To support higher-level reasoning, some works further introduce planning modules or structured intermediate representations, including Chain-of-Thought prompting [21], hierarchical planning [22, 23, 24], and spatial abstractions [25] such as bounding boxes [26], dense correspondence fields [27], 3D points [28], or trajectory traces [29]. However, these approaches remain largely reactive, predicting actions from current observations without explicitly modeling how the environment evolves through interaction.

World-Action Model (WAM). Recent work has increasingly explored video generation and world modeling for robot control, motivated by the observation that video models capture temporal dynamics and future evolution largely absent from static vision-language pretraining. Some works leverage video models as predictive representation learners or transferable world priors, followed by separate action decoding modules [30, 31, 32, 33]. Another line of research more tightly couples future video and action prediction within unified architectures [34, 35, 36], showing that jointly modeling visual futures and actions can improve generalization and data efficiency. Recent methods further build on powerful pretrained video foundation models to enable closed-loop control, causal rollout,

and planning over predicted futures [37, 38]. Works like DreamZero [8], Cosmos Policy [9] and LingBot-VA [39] exemplify this trend, demonstrating that stronger video priors can improve policy generalization and embodiment transfer. Fast-WAM [40] further shows that retaining video co-training while removing test-time future rollout can preserve strong action performance with substantially lower latency. Our work is related to this emerging world-action modeling paradigm, but differs in that we do not rely on explicit future video generation. Instead, future information is used to shape a latent reasoning space that directly participates in action generation.

Additionally, a related line of work explores future prediction and future-aware alignment in latent space for action generation [41, 42, 43, 44]. Recent approaches introduce supervision from future representations [45, 46, 47, 48], align action-side hidden states with future-conditioned features [49], or compress future observations into action-relevant conditions [50]. Other works introduce explicit or latent reasoning into VLA policies, either distilling verbalizable reasoning into compact latent plans [51] or constructing latent reasoning tokens from future-derived supervision [52, 46]. Earlier, Play-LMP [53] used a posterior over the full trajectory to regularize a latent plan inferred from the current observation and goal. However, future signals are typically attached directly to the action pathway, treated as a fixed target (*e.g.*, frozen future visual features [43, 44] or predefined future knowledge [45]), or learned through a staged procedure that first learns a future-derived target, then predicts it from the current context. In contrast, our posterior is itself a future-conditioned policy branch that passes through the shared backbone and is jointly optimized by the same action objective, so the alignment target adapts toward manipulation-relevant information rather than being fixed in advance. Specifically, we introduce future-aligned representations into explicit latent queries before action generation, and jointly align prior and posterior branches over these queries, enabling the latent space to adaptively organize future-aware yet context-inferable information as a form of latent reasoning for action decoding.

Human-Centric Learning. The high cost of robot demonstrations has motivated learning from human data, which offers far broader coverage of objects, environments, and long-horizon interactions. One line of work scales robot-compatible demonstrations through portable interfaces such as UMI [54, 55], recently exceeding 10,000 hours [56]. Another learns from large human video corpora [57, 58, 59, 60, 61], first for visual representation learning [62, 63] and later with structured supervision such as latent actions [64, 49, 65], future prediction [66, 67], and geometric cues [68, 69, 70, 71]. Closer to policy learning, recent works recover executable signals via hand pose estimation [72], wrist tracking, or retargeted trajectories [73] for VLA pretraining [74, 14, 75, 76]. We follow this direction and further inject future-aligned structure into latent reasoning queries, learning compact action-relevant future representations from large-scale human videos.

3 Method

3.1 Latent Queries as a Perception-to-Action Reasoning Interface

An effective embodied model should not only react to the current context, but also encode predictive information about how interactions may evolve. VLAs and video-generative WAMs capture two complementary aspects of this goal. The VLA provides an efficient observation-to-action pathway, but packs an explicit space for future-aware reasoning. In contrast, the video-generative WAM introduces predictive structure through dense visual forecasting, but pixel reconstruction is computationally expensive and often mismatched to the abstraction required for control. We bridge these paradigms through a latent reasoning space: an explicit, action-oriented interface that organizes future-relevant information before low-level action generation.

As illustrated in Fig. 1 (Left), we instantiate this interface by inserting a small set of learnable latent queries between the multimodal context and action tokens. Let x denote the instruction, $o_{t-H+1:t}$ the observation history of horizon H, and s the robot state. With $c_t = [x;\, o_{t-H+1:t};\, s_t]$, we insert K latent queries $Q \in \mathbb{R}^{K\times d}$ before the noised action chunk a_τ, yielding $S^{\text{latent}} = [c_t;\, Q;\, a_\tau]$, where a_τ denotes the action chunk at diffusion timestep τ. This placement is intentional. The latent queries lie between perception and action, serving as an intermediate workspace for organizing context-derived information before action prediction. Through layer-by-layer propagation, the queries integrate task- and interaction-relevant signals from multimodal context and form a compact action-oriented latent state, which subsequently conditions action generation. As a result, the model no longer maps abstract multimodal semantics directly into dense low-level actions, but instead reasons through

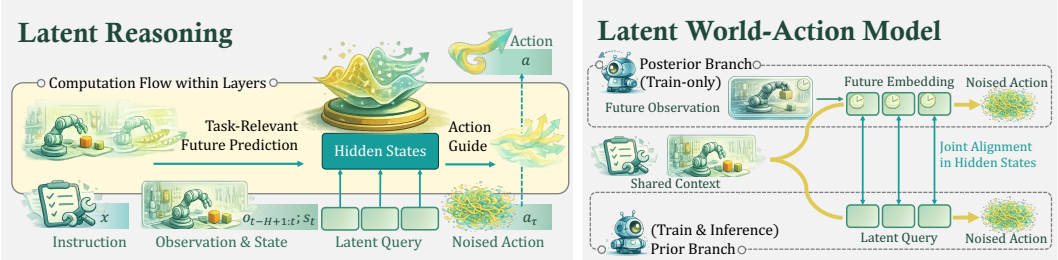


Figure 1: **Overview of our latent reasoning and world-action modeling. (Left)** Learnable latent queries are inserted between multimodal context and action tokens, forming a compact reasoning interface for action generation. **(Right)** A posterior branch replaces the queries with future embeddings during training, and joint alignment transfers future-aware structure to the deployable prior branch, yielding a latent world-action model that requires no future rollout at inference time.

an intermediate latent state before decoding actions. Architecturally, these queries are learnable placeholder tokens similar to register tokens [77]; the distinction lies in how their hidden states are shaped, as they directly condition action generation and are aligned with future-conditioned posterior states. However, latent queries alone do not guarantee future-aware reasoning. When trained only with action supervision, the latent states may collapse to weak intermediate features or shallow cues sufficient for local action decoding. We therefore introduce the future-informed joint alignment mechanism in Section 3.2, which explicitly shapes the latent reasoning space using subsequent observations during training.

3.2 Future-Informed Joint Alignment for Latent World-Action Modeling

The latent query interface provides a deployable pathway for organizing action-relevant information before action generation. However, with action supervision alone, it may still learn shallow correlations rather than predictive structure. To turn it into a latent WAM, we introduce future information supervision during training while keeping inference dependent only on the current context.

Dual-Branch Design. We adopt a dual-branch design that differs only at the latent reasoning positions. The deployable prior branch uses learnable latent queries, while a training-only posterior branch replaces them with future embeddings extracted from subsequent observations:

$$S^{\text{prior}} = [c_t;\, Q;\, a_\tau], \qquad S^{\text{post}} = [c_t;\, z^{\text{post}};\, a_\tau], \qquad z^{\text{post}} = E(\tilde{o}_{t+1:t+T}) \in \mathbb{R}^{K\times d}. \tag{1}$$

Here, E is a temporal visual encoder implemented by a frozen pretrained ViT followed by a Perceiver resampler. The two branches share the same current context and action-generation pathway: the prior branch infers latent reasoning states from the current observation, while the posterior branch provides a future-informed counterpart of the same interface.

Joint Alignment. The posterior branch provides a future-informed counterpart of the perception-to-action interface. Rather than reconstructing future frames, it converts subsequent observations into latent states that are trained to support action generation. We align the prior and posterior hidden states at the matched latent positions. Let $h_\ell^{\text{prior}}, h_\ell^{\text{post}} \in \mathbb{R}^{K\times d}$ denote the latent hidden states at the ℓ-th aligned layer of the learnable queries and future embeddings, respectively. The alignment loss is

$$\mathcal{L}_{\text{align}} = \frac{1}{L}\sum_{\ell=1}^{L} \frac{1}{Kd} \left\| h_\ell^{\text{prior}} - h_\ell^{\text{post}} \right\|_2^2, \tag{2}$$

where L is the number of aligned layers. Through this joint alignment, the prior branch learns to infer posterior-like, action-relevant latent states from the current context alone. Future observations therefore act as privileged supervision for the reasoning interface rather than pixel reconstruction targets. At inference time, the posterior branch and future encoder are removed entirely, enabling direct action prediction based on the prior branch without future-frame generation or visual rollout.

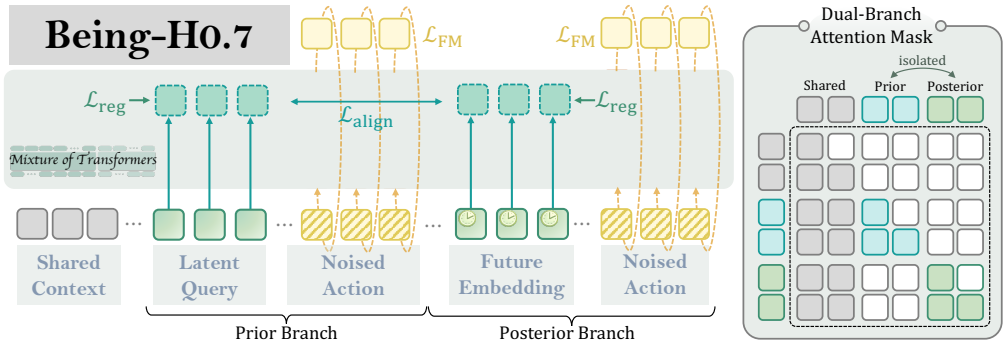


Figure 2: **Architecture of Being-H0.7.** We pack the prior and posterior branches into a single sequence with shared context and optimize them jointly. The posterior branch replaces latent queries with future embeddings, while the two branches are coupled through hidden-state alignment and lightweight regularization. A dual-branch attention mask isolates the prior and posterior branches while preserving access to the shared context for efficient training.

3.3 Efficient Dual-Branch Implementation

We implement the latent world-action model in a structurally simple and training-efficient way, as illustrated in Fig. 2. We adapt a Mixture-of-Transformers [78] structure following [76], where action and state vectors are processed by a dedicated Action Expert, while other signals are processed by a larger Understanding Expert. Instead of running two separate forward passes, we pack the prior and posterior branches into a single sequence. The two branches share the same context tokens, and their branch-specific tokens are placed at separate latent reasoning positions. Specifically, the prior branch inserts learnable latent queries before the action tokens, while the posterior branch replaces these queries with future embeddings of the same shape.

To preserve the intended dual-branch structure within one packed sequence, we apply a dual-branch attention mask. Shared context tokens are visible to both branches, while the prior and posterior branch tokens are not allowed to attend to each other. Thus, the two branches are coupled only through the alignment loss applied to corresponding latent reasoning positions. In addition, we assign identical positional IDs to corresponding prior and posterior token positions, ensuring that latent queries and future embeddings remain structurally matched throughout the Transformer layers. This design enables the model to efficiently realize the latent world-action modeling through the dual-branch formulation within a single backbone forward pass.

For action generation, we apply a flow-matching objective to both prior and posterior branches. Let a denote the ground-truth action chunk, $\tau \in [0,1]$ the sampled flow time, and $\epsilon \sim \mathcal{N}(0, I)$ the Gaussian noise. We construct the interpolated action $a_\tau = \tau a + (1-\tau)\epsilon$ and target velocity $u_\tau = a - \epsilon$ along the linear probability path. The two branches predict velocity fields $v_\theta^{\text{prior}}(a_\tau, c, Q)$ and $v_\theta^{\text{post}}(a_\tau, c, z^{\text{post}})$, respectively. The combined flow-matching loss is

$$\mathcal{L}_{\text{FM}} = \left\| v_\theta^{\text{prior}}(a_\tau, c, Q) - u_\tau \right\|_2^2 + \left\| v_\theta^{\text{post}}(a_\tau, c, z^{\text{post}}) - u_\tau \right\|_2^2 . \tag{3}$$

Since hidden-state alignment alone may admit trivial solutions, we apply lightweight anti-collapse regularization to the latent reasoning states of both branches from two aspects: norm preservation and spectral diversity. For a latent hidden state h, we use the norm regularizer

$$\mathcal{R}_{\text{norm}}(h) = \left[\text{ReLU}(\gamma - \|h\|_2)\right]^2 , \tag{4}$$

where γ is a predefined threshold. This prevents the aligned states from collapsing toward vanishing magnitude. For spectral diversity, let $H \in \mathbb{R}^{M \times n}$ denote the projection of a collection of latent hidden states from one branch at one aligned layer onto a random n-dimensional subspace, where M is the number of collected latent states. We normalize each row of H to unit norm to obtain $\hat{H}$, compute the Gram matrix $G = \hat{H}\hat{H}^\top$, and let $\{\lambda_i\}_{i=1}^M$ be its eigenvalues. With the normalized

spectrum $p_i = \lambda_i / \sum_j \lambda_j$, we define

$$\mathcal{R}_{\text{rank}}(H) = \sum_{i=1}^{M} p_i \log p_i. \tag{5}$$

Minimizing this negative spectral entropy encourages a flatter spectrum and discourages directional collapse of the latent reasoning space. Applying both to the aligned latent states across all aligned layers, we obtain $\mathcal{L}_{\text{reg}} = w_{\text{norm}}\mathcal{R}_{\text{norm}} + w_{\text{rank}}\mathcal{R}_{\text{rank}}$. In practice, we pretrain the model on mixed human and robot manipulation data following the unified sequence format of UniHand 2.0 [76]. This provides a shared interface for cross-embodiment action learning and allows heterogeneous manipulation trajectories to be trained within the same latent world-action framework. Together with the prior–posterior alignment loss $\mathcal{L}_{\text{align}}$ defined in the previous section, the final training objective is

$$\mathcal{L} = \mathcal{L}_{\text{FM}} + w_{\text{align}}\mathcal{L}_{\text{align}} + \mathcal{L}_{\text{reg}}. \tag{6}$$

4 Experiments

4.1 Implementation details

Across all experiments, the policy uses RGB-only observations. Context frames are resized to 224×224, while future frames for the posterior branch are resized to 256×256. Being-H0.7 adopts a Mixture-of-Transformers architecture, using InternVL3.5 [79] as the understanding expert and Qwen3 [80] as the action expert. To maintain a consistent visual embedding space, both context and future observations use V-JEPA2.1 [81] as the encoder, while only the context encoder remains trainable. We choose V-JEPA2.1 because it compactly encodes multi-frame temporal context and shows strong action-related representations in prior probing studies [82]; the method is not tied to this choice, as DINOv3 performs comparably (Appendix B). During pretraining, we use an observation frame horizon of $H = 4$, an action chunk length of $T = 20$, and $K = 16$ latent reasoning queries. The posterior branch observes the same number of future embeddings, yielding one-to-one alignment with prior latent queries. Prior-posterior alignment is applied to the last 9 transformer layers. We jointly optimize both branches using the flow-matching action objective, the prior–posterior latent alignment loss, and anti-collapse regularization, with $w_{\text{align}} = 10^{-3}$, $\gamma = 1.0$, and $w_{\text{norm}} = w_{\text{rank}} = 10^{-4}$. More details are available in Appendix A. Pretraining uses mixed human and robot manipulation trajectories following the unified sequence format from [76]. For downstream post-training, we optimize only the action objective and latent alignment loss on task-specific demonstrations, without anti-collapse regularization. For real-robot tasks, we use three platforms: PND Adam-U, Unitree G1, and Franka FR3 (see Appendix C.1).

4.2 Results on Simulation Benchmarks

We evaluate Being-H0.7 on four widely used simulation benchmarks: **1) LIBERO** [83]: a lifelong tabletop manipulation benchmark with four task suites (Goal, Object, Spatial, and Long). Following [83], we train on all suites, evaluate 500 trials per suite, and report the average success rate. **2) RoboCasa** [84]: a large-scale benchmark for long-horizon household manipulation in diverse kitchen environments. We evaluate 24 tasks in the few-shot setting using 50 demonstrations per task, with 50 trials per task on held-out scenes to assess robustness to unseen objects and kitchen layouts. **3) LIBERO-plus** [85], a benchmark for robustness and zero-shot generalization under controlled environmental perturbations. Following [85], we evaluate both a model trained on standard LIBERO and a variant further finetuned on LIBERO-plus. **4) CALVIN** [86]: a long-horizon multi-task benchmark across four environments (A/B/C/D). Following the standard protocol, we evaluate on ABCD→D (testing on seen environment D) and ABC→D (testing on unseen D) splits over 1,000 instruction sequences of 5 consecutive tasks, reporting the average number of completed tasks per sequence. Before tuning on the downstream tasks, we use 15K hours of robot data together with 20K hours of human video data for the pre-training, comparable in scale to prior works such as $\pi_{0.5}$.

Across all four simulation benchmarks, Being-H0.7 achieves state-of-the-art overall performance and the highest average rank (see Tab. 1). On LIBERO, it reaches a 99.0% average success rate across all suites. On RoboCasa, it achieves 62.9%, demonstrating strong robustness in diverse unseen kitchen environments. Under environmental perturbations in LIBERO-plus, Being-H0.7 obtains an 84.8%

Table 1: Comparison on embodied manipulation benchmarks. CALVIN/CALVIN* denote "ABCD → D"/"ABC → D", and LIBERO-plus* denotes finetuning on LIBERO-plus.

Model	Size	LIBERO	LIBERO-plus	LIBERO-plus*	RoboCasa-50	CALVIN	CALVIN*
# VLA							
π0 [3]	3B	94.4	53.6	-	42.4	-	3.92
π0-FAST[87]	3B	85.5	61.6	-	-	-	-
gr00t-N1.6 [4]	3B	93.9	-	-	36.0	4.60	4.24
π0.5 [88]	3B	96.9	77.4	-	41.4	4.06	4.13
MINT-4B [89]	4B	98.7	80.1	84.1	-	4.57	-
ABot-M0 [90]	4B	98.6	80.5	-	-	-	-
Being-H0.5 [76]	2B	98.9	78.5	83.1	53.5	4.63	4.48
# World Model							
UVA [34]	-	-	-	-	50.0	-	-
DreamVLA [45]	-	92.6	-	-	-	-	4.44
JEPA-VLA [91]	-	96.4	25.6	-	-	-	-
VLA-JEPA [44]	-	96.1	79.5	-	-	-	-
Cosmos-Policy [9]	2B	98.5	-	-	**67.1**	-	-
Fast-WAM [40]	6B	97.6	-	-	-	-	-
Being-H0.7	3B	**99.0**	**84.8**	**87.1**	62.9	**4.65**	**4.48**

zero-shot success rate, further improving to 87.1% after fine-tuning on LIBERO-plus, highlighting its resilience against shifted viewpoints, novel textures, and sensor noise. Finally, on CALVIN, Being-H0.7 completes an average of 4.65 and 4.48 tasks out of 5 on the ABCD→D and ABC→D splits, respectively, highlighting strong long-horizon execution and zero-shot generalization.

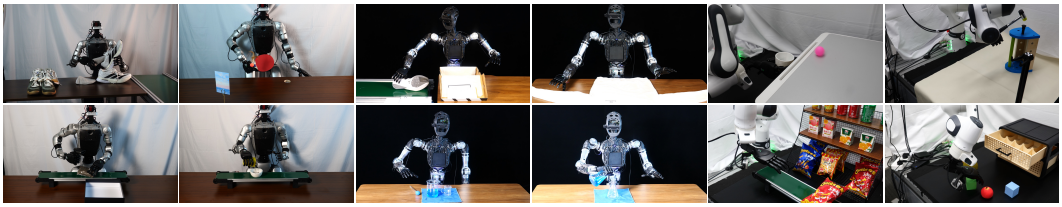


Figure 3: **Overview of real-robot tasks**, illustrating the task scenes across Unitree G1, PND Adam-U, and Franka FR3, spanning five ability-oriented evaluation suites.

4.3 Real-robot Experiments

Task Suites. We design 12 tasks and group them into five ability-oriented suites: dynamic scene, physical reasoning, motion reasoning, long-horizon execution, and generalization. These suites are intentionally compositional, as many tasks simultaneously require multiple capabilities, such as reacting to moving targets while also reasoning about object trajectories and containment. We summarize the task set and evaluation prompts in Tab. 12 in the Appendix, while Fig. 3 visualizes the corresponding real-world task scenes. Each task is assigned one primary suite together with optional overlap tags, and suite-level scores are computed over all tasks sharing the corresponding tag. Each suite targets a different source of difficulty. Dynamic Scene tasks require timely reactions to moving objects or changing environments. Physical Reasoning evaluates understanding of gravity, fluid transfer, deformable contact, containment, and tool use. Motion Reasoning focuses on trajectory anticipation, relative velocity, and contact timing. Long Horizon Execution stresses sequential consistency and subgoal memory across multiple stages. Finally, Generalization measures robustness under shifted layouts, shelf levels, novel containers, and unseen object instances.

Evaluation Protocol. All policies are deployed via a black-box inference server, ensuring an identical execution stack across methods. For each task, we predefine scene layouts and initial conditions, then randomize both the evaluated policy endpoint and rollout order during testing. Task success is recorded using a fixed binary criterion while the active policy endpoint remains hidden to the operator. Unless otherwise specified, each task is evaluated over 20 blind trials per method. This protocol is particularly important for suites that evaluate not only final task completion but also reaction quality under dynamic interactions. Tasks involving moving objects, deformable materials, or fluid

interactions can be sensitive to small timing differences in policy updates. The shared deployment infrastructure and blind-evaluation procedure ensure consistent comparisons across methods.



Figure 4: **Comparison of real-robot success rates (%).** across five task suites. Each task is evaluated over 20 blind trials, and suite scores are averaged over all tasks with the corresponding suite tag.

Result Analysis. Figure 4 summarizes suite-level success rates computed from the overlap-tag aggregation described above. Being-H0.7 achieves the best performance across all five suites, covering reactive, physical, sequential, and generalization-oriented tasks on all three embodiments. Importantly, the gains are consistently distributed across the benchmark rather than concentrated in a single task category. **1) Dynamic and motion-centric suites.** The largest improvements appear in these two suites, which contain the most timing-sensitive tasks, including catching rolling balls, racket-based redirection, pouring into moving containers, and conveyor interaction. Such scenarios rapidly penalize stale state estimates, where even small delays or contact errors can push the interaction beyond the feasible control window. Among the baselines, Fast-WAM performs strongest in these reactive settings, consistent with its focus on low-latency action generation. Being-H0.7 further improves over this baseline, suggesting that robust reactive behavior depends not only on inference speed, but also on future-aware latent representations that capture object motion, contact timing, and downstream action consequences. **2) Physical and long-horizon execution.** Tasks in these suites, such as pipette transfer and funnel pouring, require not only dexterity but also reasoning about containment, gravity, and tool-mediated interaction. Being-H0.7 consistently outperforms the baselines, indicating that its learned world-action prior improves both reactive control and causal consistency across longer and physically constrained manipulation sequences. **3) Generalization.** This suite evaluates robustness under different changes. Here, $\pi_{0.5}$ and Being-H0.5 remain competitive, making the benchmark balanced and practically meaningful. Nevertheless, Being-H0.7 achieves the strongest overall performance, suggesting that its latent predictive prior generalizes effectively across both embodiment and scene variations.

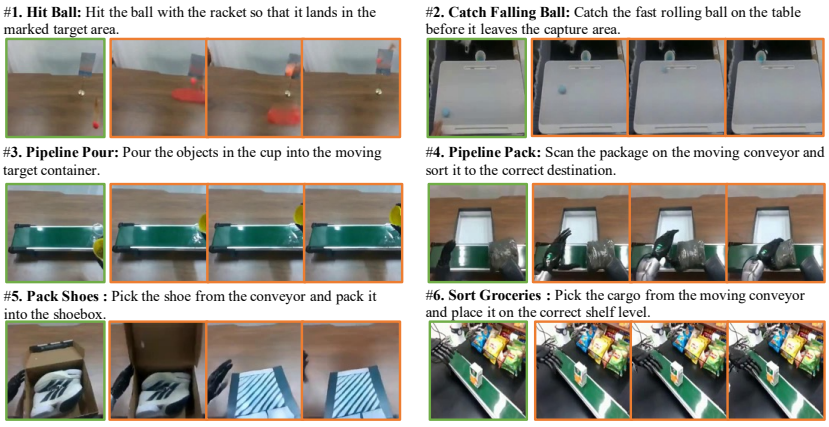


Figure 5: **Qualitative visualization of the latent reasoning.**

Linear Retrieval Probe. To evaluate whether the prior latent encodes future dynamics, we use a lightweight linear retrieval probe that tests whether future-relevant information is linearly decodable from the latent reasoning space. Given a prior latent $h_{(i)}^{\text{prior}}$ inferred from current observation, a

Table 2: Comparison of different linear retrieval probe settings. At test time, the ground-truth future observation is ranked among 32 candidate observations.

Metric	Noise	Latent	Image + Noise	Image + Latent
Top-1 Acc (%)	3.1	61.8	27.4	63.3
Top-5 Acc (%)	15.6	94.1	65.7	94.7
Top-10 Acc (%)	31.3	98.6	83.7	98.8

linear layer projects the normalized latent into the future-observation embedding space encoded by DINOv3 [92]: $\hat{v}_i = W\,\mathrm{Norm}(h_{(i)}^{\mathrm{prior}})$. The probe is trained with an in-batch contrastive loss against the normalized embedding v_i^+ of the corresponding ground-truth future observation:

$$\mathcal{L}_i = -\log \frac{\exp\left(\mathrm{sim}(\hat{v}_i, v_i^+)/\kappa\right)}{\sum_{j=1}^{B} \exp\left(\mathrm{sim}(\hat{v}_i, v_j^+)/\kappa\right)}, \tag{7}$$

where $\mathrm{sim}(\cdot,\cdot)$ denotes cosine similarity, κ is the temperature, and B is the batch size. During evaluation, the ground-truth target is defined as the observation 1 second into the future, while negatives are sampled from other videos of the same task with similar temporal progress. Retrieval performance is measured by ranking the ground-truth future among all candidates using cosine similarity. This setup directly tests whether the prior latent contains sufficient predictive signals to distinguish the correct future evolution from plausible alternatives.

Across the four settings, *Noise*, *Latent*, *Image+Noise*, and *Image+Latent* in Table 2, a consistent trend can be observed. Here, *Image+Noise* and *Image+Latent* are constructed by concatenating the DINOv3 [92] current observation embedding with either a random noise vector or the prior latent, respectively. Compared with random noise, the prior latent leads to substantially better retrieval performance both with and without image context, indicating that the prior latent contains structured predictive information that is useful for identifying the correct future observation. Notably, this advantage persists even when strong visual context from the current image is provided, suggesting that the latent encodes future-relevant signals beyond a trivial representation of the input frame. To test whether this property stems from joint alignment rather than the learnable queries themselves, we further apply the probe to a Being-H0.5-style single-branch baseline and to Being-H0.7 without alignment under a controlled pretraining setting (Appendix B). Without alignment, latent queries do not consistently outperform observation tokens, whereas with joint alignment they improve Top-1/Top-5/Top-10 accuracy from 61.5%/85.7%/90.3% to 65.8%/92.2%/97.6%, indicating that joint alignment makes future information more explicitly organized and linearly accessible at the latent interface.

Visualization of the Latent Reasoning. To more directly reveal what is encoded in this *latent reasoning space*, the current observation and the latent hidden states from the prior branch of Being-H0.7 can be used jointly as conditions for a video generation model [10] to synthesize future task states. Although Being-H0.7 does not explicitly reconstruct future frames during inference, the resulting visualizations in Fig. 5 suggest that its latent representations already capture predictive information about how the world will evolve. This provides further evidence that Being-H0.7 functions as a latent world action model: rather than modeling the future at the pixel level, it learns an internal predictive representation that is sufficient to support action generation.

4.4 Ablation Studies

We further ablate the main design choices behind the latent world-action model. Our goal is to determine whether the gains arise from the proposed prior-posterior joint alignment and the anti-collapse regularization used during latent pretraining.

Source of improvements. We progressively build Being-H0.7 from a Being-H0.5-style single-branch baseline and evaluate on LIBERO-plus, RoboCasa, and RoboTwin 2.0 [93] (Clean-Train $\rightarrow$ Randomized-Eval, 50 tasks $\times$ 100 rollouts). As shown in Tab. 3, the gains come from three complementary factors. The training-only posterior branch alone helps by providing a future-conditioned, IDM-like auxiliary task for the shared backbone; joint alignment brings a further gain by transferring this future-aware information into the deployable prior branch; and large-scale pretraining

Table 3: Progressive decomposition of the improvements. The first three rows are trained from scratch on downstream data.

Variant	LIBERO-plus	RoboCasa	RoboTwin 2.0
Single-branch baseline	74.1	50.1	42.7
+ Posterior branch	77.3	52.2	45.2
+ Joint alignment	80.4	55.8	50.1
+ Large-scale pretraining	**84.8**	**62.9**	**59.2**

Table 4: Ablation studies on alignment strategy and anti-collapse regularization.

(a) Alignment strategy. All variants are trained from scratch on downstream data.

Variant	LIBERO-plus	RoboCasa
w/o latent alignment	77.3	52.2
prior → posterior alignment	75.2	49.3
joint prior–posterior alignment	**80.4**	**55.8**

(b) Anti-collapse regularization.

Pre. Reg.	Post. Reg.	RoboCasa
✗	✗	60.7
✗	✓	58.2
✓	✗	**62.9**
✓	✓	59.6

contributes the rest. Under a controlled pretraining setup (1K hours of robot data and 1K hours of human video) with five downstream runs per variant, joint alignment still improves over the posterior-only variant by +5.1% on LIBERO-plus and +3.8% on RoboTwin, well beyond the standard deviations of 0.5–1.2% (Appendix B.1).

Effect of joint alignment. We compare three ways of using the future-informed posterior branch on LIBERO-plus and RoboCasa: (i) *w/o latent alignment*, where the prior and posterior branches are trained only with the action objective; (ii) *prior → posterior alignment*, where the prior is aligned to a detached posterior target; and (iii) *joint prior-posterior alignment*, our default design, where the two branches are optimized jointly with latent-state alignment. All variants are trained from scratch on the corresponding benchmark without large-scale human-robot pretraining, isolating the effect of the alignment strategy. As shown in Tab. 4a, joint alignment performs best on both LIBERO-plus and RoboCasa. Removing latent alignment weakens the transfer of future information to the deployable prior branch. One-way alignment performs even worse, suggesting that a fixed posterior target can over-constrain the prior rather than forming a shared latent reasoning space. These results indicate that the key is not simply adding a future-conditioned branch, but jointly aligning the two branches so that the latent space remains both future-informed and inferable from current observations. A FLARE-style [43] fixed target using frozen DINOv3 [92] future features reaches only 44.7% on RoboTwin 2.0, close to the posterior branch without alignment (45.2%) and well below joint alignment (50.1%), supporting an adaptive, action-optimized posterior target (Appendix B).

Effect of anti-collapse regularization. We further ablate the norm-preservation and spectral-diversity regularizers on RoboCasa. As shown in Tab. 4b, applying regularization during pretraining gives the best performance, while adding it during downstream post-training is not beneficial. This suggests that anti-collapse regularization is mainly needed when the latent reasoning space is first shaped from large-scale data. After pretraining, the latent space is already stabilized, and downstream adaptation should focus on task-specific action fitting and prior–posterior alignment. Accordingly, our final model applies regularization only during pretraining. Sensitivity analyses of the loss coefficients, the number of latent queries, future frames, and aligned layers, as well as comparisons of visual encoders and observation history, are provided in Appendix B.

5 Conclusion

We introduced Being-H0.7, a latent world-action model that bridges direct action prediction and world modeling through a compact latent reasoning space. By aligning a deployable prior branch with a future-aware posterior branch, the model injects future-relevant priors into action generation without requiring costly pixel-level rollout or video generation at inference time. Combined with large-scale human video pretraining, Being-H0.7 provides an effective and scalable framework for embodied learning, enabling strong performance across tasks while maintaining efficient deployment.

Acknowledgments and Disclosure of Funding

This work was supported by NSFC in part under Grant 62450001 and 62476008. The authors would like to thank the anonymous reviewers for their valuable comments and advice. The authors declare no competing interests.

References

[1] 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.

[2] 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.

[3] 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.

[4] NVIDIA, 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*, 2025.

[5] Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, Yihang Chen, Jie Liu, Yansong Cheng, Yao Yao, Jiayi Zhu, Yihao Meng, Kecheng Zheng, Qingyan Bai, Jingye Chen, Zehong Shen, Yue Yu, Xing Zhu, Yujun Shen, and Hao Ouyang. Advancing open-source world models. *arXiv preprint arXiv:2601.20540*, 2026.

[6] Jake Bruce, Michael D 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 *Forty-first International Conference on Machine Learning*, 2024.

[7] Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[8] 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*, 2026.

[9] 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.

[10] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[11] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[12] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025.

[13] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

[14] Yicheng Feng, Wanpeng Zhang, Ye Wang, Hao Luo, Haoqi Yuan, Sipeng Zheng, and Zongqing Lu. Spatial-aware VLA pretraining through visual-physical alignment from human videos. *arXiv preprint arXiv:2512.13080*, 2025.

[15] Ye Wang, Sipeng Zheng, Hao Luo, Wanpeng Zhang, Haoqi Yuan, Chaoyi Xu, Haiweng Xu, Yicheng Feng, Mingyang Yu, Zhiyu Kang, et al. Rethinking visual-language-action model scaling: Alignment, mixture, and regularization. *arXiv preprint arXiv:2602.09722*, 2026.

[16] Andreas Steiner, André Susano Pinto, Michael Tschannen, Daniel Keysers, Xiao Wang, Yonatan Bitton, Alexey Gritsenko, Matthias Minderer, Anthony Sherbondy, Shangbang Long, et al. PaliGemma 2: A family of versatile VLMs for transfer. *arXiv preprint arXiv:2412.03555*, 2024.

[17] 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.

[18] Wanpeng Zhang, Zilong Xie, Yicheng Feng, Yijiang Li, Xingrun Xing, Sipeng Zheng, and Zongqing Lu. From pixels to tokens: Byte-pair encoding on quantized visual modalities. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

[20] Zhixuan Liang, Yizhuo Li, Tianshuo Yang, Chengyue Wu, Sitong Mao, Tian Nian, Liuao Pei, Shunbo Zhou, Xiaokang Yang, Jiangmiao Pang, et al. Discrete diffusion VLA: Bringing discrete diffusion to action decoding in vision-language-action policies. *arXiv preprint arXiv:2508.20072*, 2025.

[21] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

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

[23] Fanqi Lin, Ruiqian Nai, Yingdong Hu, Jiacheng You, Junming Zhao, and Yang Gao. OneTwoVLA: A unified vision-language-action model with adaptive reasoning. *arXiv preprint arXiv:2505.11917*, 2025.

[24] Jaden Clark, Suvir Mirchandani, Dorsa Sadigh, and Suneel Belkhale. Action-free reasoning for policy generalization. *arXiv preprint arXiv:2502.03729*, 2025.

[25] Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. *arXiv preprint arXiv:2510.12276*, 2025.

[26] Brent Griffin. Mobile robot manipulation using pure object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 561–571, 2023.

[27] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. CURL: Contrastive unsupervised representations for reinforcement learning. In *International conference on machine learning*, pages 5639–5650. PMLR, 2020.

[28] Andreas Ten Pas and Robert Platt. Using geometry to detect grasp poses in 3D point clouds. In *Robotics Research: Volume 1*, pages 307–324. Springer, 2017.

[29] Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, et al. MolmoAct: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025.

[30] 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.

[31] Jonas Pai, Liam Achenbach, Victoriano Montesinos, Benedek Forrai, Oier Mees, and Elvis Nava. mimic-video: Video-action models for generalizable robot control beyond VLAs. *arXiv preprint arXiv:2512.15692*, 2025.

[32] Yao Feng, Hengkai Tan, Xinyi Mao, Chendong Xiang, Guodong Liu, Shuhe Huang, Hang Su, and Jun Zhu. Vidar: Embodied video diffusion model for generalist manipulation. *arXiv preprint arXiv:2507.12898*, 2025.

[33] Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Jingbin Cai, Si Liu, Jianlan Luo, et al. Genie envisioner: A unified world foundation platform for robotic manipulation. *arXiv preprint arXiv:2508.05635*, 2025.

[34] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025.

[35] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025.

[36] 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.

[37] 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. *arXiv preprint arXiv:2512.13030*, 2025.

[38] 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.

[39] 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.

[40] 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.

[41] Hao Luo and Zongqing Lu. Learning video-conditioned policy on unlabelled data with joint embedding predictive transformer. In *International Conference on Learning Representations*, 2025.

[42] Aleksandar Vujinovic and Aleksandar Kovacevic. ACT-JEPA: Novel joint-embedding predictive architecture for efficient policy representation learning. *arXiv preprint arXiv:2501.14622*, 2025.

[43] Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loïc Magne, Avnish Narayan, You Liang Tan, Guanzhi Wang, Qi Wang, Jiannan Xiang, Yinzhen Xu, Seonghyeon Ye, Jan Kautz, Furong Huang, Yuke Zhu, and Linxi Fan. FLARE: Robot learning with implicit world modeling. In *Annual Conference on Robot Learning*, 2025.

[44] Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. VLA-JEPA: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026.

[45] Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, XinQiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. DreamVLA: A vision-language-action model dreamed with comprehensive world knowledge. In *Annual Conference on Neural Information Processing Systems*, 2025.

[46] Zhuoyang Liu, Jiaming Liu, Hao Chen, Jiale Yu, Ziyu Guo, Chengkai Hou, Chenyang Gu, Xiangju Mi, Renrui Zhang, Kun Wu, et al. LaST_0: Latent spatio-temporal chain-of-thought for robotic vision-language-action model. *arXiv preprint arXiv:2601.05248*, 2026.

[47] Han Zhao, Jingbo Wang, Wenxuan Song, Shuai Chen, Yang Liu, Yan Wang, Haoang Li, and Donglin Wang. FRAPPE: Infusing world modeling into generalist policies via multiple future representation alignment. *arXiv preprint arXiv:2602.17259*, 2026.

[48] Wanpeng Zhang, Hao Luo, Sipeng Zheng, Yicheng Feng, Haiweng Xu, Ziheng Xi, Chaoyi Xu, Haoqi Yuan, and Zongqing Lu. Conservative offline robot policy learning via posterior-transition reweighting. *arXiv preprint arXiv:2603.16542*, 2026.

[49] Hao Luo, Ye Wang, Wanpeng Zhang, Haoqi Yuan, Yicheng Feng, Haiweng Xu, Sipeng Zheng, and Zongqing Lu. Joint-aligned latent action: Towards scalable VLA pretraining in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[50] Yue Su, Sijin Chen, Haixin Shi, Mingyu Liu, Zhengshen Zhang, Ningyuan Huang, Weiheng Zhong, Zhengbang Zhu, Yuxiao Liu, and Xihui Liu. World guidance: World modeling in condition space for action generation. *arXiv preprint arXiv:2602.22010*, 2026.

[51] Chi-Pin Huang, Yunze Man, Zhiding Yu, Min-Hung Chen, Jan Kautz, Yu-Chiang Frank Wang, and Fu-En Yang. FAST-thinkact: Efficient vision-language-action reasoning via verbalizable latent planning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[52] Shuanghao Bai, Jing Lyu, Wanqi Zhou, Zhe Li, Dakai Wang, Lei Xing, Xiaoguang Zhao, Pengwei Wang, Zhongyuan Wang, Cheng Chi, Badong Chen, and Shanghang Zhang. Latent reasoning VLA: Latent thinking and prediction for vision-language-action models. In *International Conference on Machine Learning*, 2026.

[53] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on Robot Learning*, pages 1113–1132, 2020.

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

[55] Mengda Xu, Han Zhang, Yifan Hou, Zhenjia Xu, Linxi Fan, Manuela Veloso, and Shuran Song. DexUMI: Using human hand as the universal manipulation interface for dexterous manipulation. *arXiv preprint arXiv:2505.21864*, 2025.

[56] Gen Robot. 10Kh-RealOmin-OpenData. Hugging Face Datasets, 2025.

[57] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[58] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18995–19012, 2022.

[59] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-Exo4D: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19383–19400, 2024.

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

[61] 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.

[62] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2022.

[63] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *The Eleventh International Conference on Learning Representations*, 2022.

[64] Jiange Yang, Yansong Shi, Haoyi Zhu, Mingyu Liu, Kaijing Ma, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. Como: Learning continuous latent motion from internet videos for scalable robot learning. *arXiv preprint arXiv:2505.17006*, 2025.

[65] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

[66] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023.

[67] 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.

[68] Hanzhi Chen, Boyang Sun, Anran Zhang, Marc Pollefeys, and Stefan Leutenegger. VidBot: Learning generalizable 3D actions from in-the-wild 2d human videos for zero-shot robotic manipulation. In *CVPR*, 2025.

[69] Alexey Gavryushin, Xi Wang, Robert JS Malate, Chenyu Yang, Xiangyi Jia, Shubh Goel, Davide Liconti, René Zurbrügg, Robert K Katzschmann, and Marc Pollefeys. MAPLE: Encoding dexterous robotic manipulation priors learned from egocentric videos. *arXiv preprint arXiv:2504.06084*, 2025.

[70] Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, et al. Magma: A foundation model for multimodal AI agents. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14203–14214, 2025.

[71] 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.

[72] Hao Luo, Yicheng Feng, Wanpeng Zhang, Sipeng Zheng, Ye Wang, Haoqi Yuan, Jiazheng Liu, Chaoyi Xu, Qin Jin, and Zongqing Lu. Being-H0: vision-language-action pretraining from large-scale human videos. *arXiv preprint arXiv:2507.15597*, 2025.

[73] Ruihan Yang, Qinxi Yu, Yecheng Wu, Rui Yan, Borui Li, An-Chieh Cheng, Xueyan Zou, Yunhao Fang, Xuxin Cheng, Ri-Zhao Qiu, et al. EgoVLA: Learning vision-language-action models from egocentric human videos. *arXiv preprint arXiv:2507.12440*, 2025.

[74] Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. EgoMimic: Scaling imitation learning via egocentric video. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13226–13233. IEEE, 2025.

[75] Qixiu Li, Yu Deng, Yaobo Liang, Lin Luo, Lei Zhou, Chengtang Yao, Lingqi Zeng, Zhiyuan Feng, Huizhi Liang, Sicheng Xu, et al. Scalable vision-language-action model pretraining for robotic manipulation with real-life human activity videos. *arXiv preprint arXiv:2510.21571*, 2025.

[76] Hao Luo, Ye Wang, Wanpeng Zhang, Sipeng Zheng, Ziheng Xi, Chaoyi Xu, Haiweng Xu, Haoqi Yuan, Chi Zhang, Yiqing Wang, et al. Being-h05: Scaling human-centric robot learning for cross-embodiment generalization. *arXiv preprint arXiv:2601.12993*, 2026.

[77] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations*, 2024.

[78] Weixin Liang, LILI YU, Liang Luo, Srini Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. Mixture-of-transformers: A sparse and scalable architecture for multi-modal foundation models. *Transactions on Machine Learning Research*, 2025.

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

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

[81] Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

[82] Dujun Nie, Fengjiao Chen, Qi Lv, Jun Kuang, Xiaoyu Li, Xuezhi Cao, and Xunliang Cai. LARY: A latent action representation yielding benchmark for generalizable vision-to-action alignment. In *European Conference on Computer Vision*, 2026.

[83] 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.

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

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

[86] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

[87] 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.

[88] 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.

[89] Renming Huang, Chendong Zeng, Wenjing Tang, Jintian Cai, Cewu Lu, and Panpan Cai. Mimic intent, not just trajectories. *arXiv preprint arXiv:2602.08602*, 2026.

[90] 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.

[91] Shangchen Miao, Ningya Feng, Jialong Wu, Ye Lin, Xu He, Dong Li, and Mingsheng Long. JEPA-VLA: Video predictive embedding is needed for VLA models. *arXiv preprint arXiv:2602.11832*, 2026.

[92] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

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

[94] Jialong Li, Xuxin Cheng, Tianshu Huang, Shiqi Yang, Ri-Zhao Qiu, and Xiaolong Wang. AMO: Adaptive motion optimization for hyper-dexterous humanoid whole-body control. *arXiv preprint arXiv:2505.03738*, 2025.

[95] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

A Architecture and Implementation Details

In this section, we provide additional implementation details of Being-H0.7. Our model largely follows the backbone in [76], including the Mixture-of-Transformers design and the unified action space for cross-embodiment training. On top of this backbone, Being-H0.7 introduces latent reasoning queries, a training-only posterior branch, and a packed dual-branch implementation for future-informed latent alignment.

A.1 Mixture-of-Transformers Backbone

Being-H0.7 adopts the Mixture-of-Transformers (MoT) backbone used in [76]. The core idea is to separate high-level multimodal understanding from low-level action generation, while still allowing the two pathways to interact through shared attention. Each Transformer block contains a shared self-attention module and modality-specific feed-forward experts. Language and visual tokens are routed to the understanding expert, which is responsible for semantic reasoning, spatial grounding, and task-level interpretation. State and action tokens are routed to the action expert, which specializes in continuous control and flow-based action generation.

The shared attention allows action tokens to attend to instruction, visual observations, proprioceptive states, and latent reasoning tokens. The expert-specific feed-forward layers reduce interference between semantic understanding and motor control, which is important for cross-embodiment learning where the model must preserve general visual-language reasoning while adapting to heterogeneous action distributions across robots. In our implementation, we instantiate the understanding expert with InternVL3.5 and the action expert with Qwen3. Both context observations and future observations are encoded with V-JEPA2.1. The context visual encoder is trainable, while the future visual encoder used by the posterior branch is frozen. All experiments use RGB-only observations. Current context images are resized to 224×224, and future frames used by the posterior branch are resized to 256×256.

A.2 Unified Action Space

Being-H0.7 also inherits the unified action space from [76]. This design provides a shared output interface for heterogeneous human and robot embodiments. Different embodiments expose different native control spaces: a parallel-gripper arm may use end-effector pose and gripper commands, a dexterous hand may include high-dimensional finger articulation, and a humanoid robot may involve bimanual arm and hand control. Instead of assigning each embodiment an independent action head, we map all states and actions into a fixed, semantically organized action vector.

The unified action vector is partitioned into physical slots. Each slot corresponds to a grounded control quantity, such as end-effector translation, end-effector rotation, gripper state, finger articulation, arm joint positions, or embodiment-specific control dimensions. For an embodiment e, its native state and action are mapped into the unified space by sparse assignment functions:

$$s = \Phi_e(s^{(e)}), \qquad a = \Phi_e(a^{(e)}), \tag{8}$$

where $s^{(e)}$ and $a^{(e)}$ denote the embodiment-specific state and action. The mapping function Φ_e places valid dimensions into their corresponding semantic slots. Slots that are not defined for the current embodiment are filled with zeros and ignored by the loss.

We maintain a binary validity mask $m^{(e)}$ for each embodiment. The action loss is applied only to valid action dimensions:

$$\mathcal{L}_{\mathrm{FM}}^{(e)} = \left\| m^{(e)} \odot \left(v_\theta(a_\tau, c_t) - u_\tau \right) \right\|_2^2, \tag{9}$$

where c denotes the multimodal context, a_τ is the noised action chunk, u_τ is the flow-matching target velocity, and $\odot$ denotes element-wise multiplication. This masking mechanism allows data from different embodiments to update the same action expert without forcing incompatible dimensions to share supervision.

Human hand motion is also treated as one embodiment in this unified space. The global wrist motion is aligned with end-effector pose slots, while finger articulation is placed into fine-manipulation slots. This allows large-scale egocentric human videos to provide action-relevant supervision for reaching,

grasping, contact, and object manipulation. For downstream robot deployment, the model predicts a full unified action chunk, but only the valid dimensions of the target embodiment are decoded and sent to the robot controller.

Actions are predicted as chunks rather than single-step commands. For each trajectory segment, the target action is represented as

$$a_{1:T} \in \mathbb{R}^{T \times d_a}, \tag{10}$$

where d_a is the dimensionality of the unified action vector. We use an action chunk length of $T = 20$ and $d_a = 200$. The same slot structure and validity mask are applied at every timestep in the action chunk.

A.3 Future Latent Encoder

The posterior branch requires a compact representation of future observations that has the same token format as the learnable latent queries in the prior branch. We implement this module as a frozen visual backbone followed by a two-stage Perceiver encoder. The frozen backbone extracts patch-level visual features from future frames, while the Perceiver modules compress variable-length spatio-temporal and multi-view visual tokens into a fixed number of future latent tokens.

Let $\tilde{o}_{0:T}$ denote the future observation segment used only during training. For each sample, the future observations may contain multiple camera views. We first encode the future frames with a frozen ViT backbone, instantiated as V-JEPA2.1 in our experiments. The resulting patch tokens are projected to a lower-dimensional Perceiver hidden space before latent compression. The ViT parameters are frozen throughout training, so gradients only update the projection layer and the two Perceiver modules.

The encoder uses two stages. The first stage is a per-view temporal Perceiver shared across all views. For each view, we concatenate the patch tokens from its future frames into a single spatio-temporal token sequence:

$$F_v \in \mathbb{R}^{T_f N_p \times d_p}, \tag{11}$$

where T_f is the number of future frames, N_p is the number of visual patch tokens per frame, and d_p is the Perceiver hidden dimension. A set of K_1 learnable Perceiver queries cross-attends to this sequence and produces a compact view-level representation:

$$Z_v^{(1)} = \mathrm{Perceiver}_1(F_v) \in \mathbb{R}^{K_1 \times d_p}. \tag{12}$$

The same $\mathrm{Perceiver}_1$ is shared across all views and all samples.

The second stage aggregates information across views. For a sample with V available views, we concatenate the per-view latents:

$$Z^{(1)} = [Z_1^{(1)}; \ldots; Z_V^{(1)}] \in \mathbb{R}^{VK_1 \times d_p}. \tag{13}$$

Since different samples may contain different numbers of valid views, we pad the view-latent sequence to the maximum number of views in the batch and use a key-padding mask to ignore padded tokens. A second Perceiver then compresses the multi-view sequence into K_2 future latent tokens:

$$z^{\mathrm{post}} = \mathrm{Perceiver}_2(Z^{(1)}) \in \mathbb{R}^{K_2 \times d}. \tag{14}$$

The output dimension d is set to the hidden size of the VLA backbone, so that z^{post} can directly replace the prior latent queries in the posterior sequence.

In both stages, the Perceiver follows a query-based compression design. Given input tokens F and learnable latent queries L, cross-attention uses L as queries and F as keys and values, so that the latent tokens read information from the visual sequence. Self-attention is then applied over the latent tokens themselves, enabling information exchange across latent slots. A feed-forward network further updates the latent tokens. Stacking this block converts variable-length visual tokens into a fixed-size latent set.

In our implementation, the future encoder uses $T_f = 4$ downsampled future frames. Each Perceiver stage contains 2 layers with 8 attention heads. The Perceiver hidden dimension is set to $d_p = 256$. The first-stage per-view Perceiver produces $K_1 = 32$ latent tokens for each view, and the second-stage cross-view Perceiver produces $K_2 = K = 16$ posterior latent tokens after aggregating all available views. The final posterior latent dimension is projected to the VLA hidden size, so that the posterior tokens can directly replace the $K = 16$ learnable latent queries in the prior sequence.

We add fixed sinusoidal positional encodings before each Perceiver stage. For the first stage, the positional encoding is factorized into temporal, height, and width components, so that the Perceiver can distinguish both frame order and patch location. For the second stage, the positional encoding is factorized into view index and latent-slot index, allowing the cross-view Perceiver to preserve view identity and per-view latent ordering. In our final model, the output future latents are used only by the training-time posterior branch and are discarded at inference.

A.4 Packed Prior–Posterior Sequence

Being-H0.7 introduces latent reasoning queries between the multimodal context and the noised action tokens. Following the notation in the main text, let x denote the instruction, $o_{t-H+1:t}$ the observation history, s_t the robot state, and a_τ the noised action chunk used by flow matching. We use an observation horizon of $H = 4$ frames and form the shared context as $c_t = [x;\, o_{t-H+1:t};\, s_t]$. The deployable prior branch inserts $K = 16$ learnable latent reasoning queries $Q \in \mathbb{R}^{K\times d}$ before the noised action tokens as $S^{\text{prior}} = [c_t;\, Q;\, a_\tau]$. These latent queries serve as an intermediate reasoning interface between perception and action generation.

During training, we additionally construct a posterior branch that has access to future observations. The posterior branch replaces the learnable latent queries with future embeddings as $S^{\text{post}} = [c_t;\, z^{\text{post}};\, a_\tau]$, where $z^{\text{post}} = E(\tilde{o}_{t+1:t+T}) \in \mathbb{R}^{K\times d}$. Here, E is the future latent encoder detailed in Appendix A.3. The encoder outputs $K = 16$ future embeddings, matching the number and dimensionality of the prior latent queries. This one-to-one correspondence makes the prior and posterior latent positions directly alignable.

A naive implementation would run the prior and posterior branches as two separate forward passes. Instead, we pack both branches into a single sequence:

$$S^{\text{pack}} = [c_t;\, Q;\, a_\tau^{\text{prior}};\, z^{\text{post}};\, a_\tau^{\text{post}}]. \tag{15}$$

The two action segments use the same noised action sample and the same flow-matching target. This packed formulation allows prior and posterior branches to be optimized in one forward pass, while preserving branch separation through the attention mask described below.

A.5 Dual-Branch Attention Mask and Positional Encoding

The packed sequence uses a block-structured dual-branch attention mask. The shared context tokens are visible to both branches. Prior-branch tokens can attend to the shared context and prior-branch tokens, but cannot attend to posterior-branch tokens. Similarly, posterior-branch tokens can attend to the shared context and posterior-branch tokens, but cannot attend to prior-branch tokens. Equivalently, the attention visibility is

$$\begin{aligned} Q, a_\tau^{\text{prior}} &\to \{c_t, Q, a_\tau^{\text{prior}}\}, \\ z^{\text{post}}, a_\tau^{\text{post}} &\to \{c_t, z^{\text{post}}, a_\tau^{\text{post}}\}. \end{aligned} \tag{16}$$

There is no direct cross-branch attention between the prior and posterior branch. Therefore, future information cannot leak into the deployable prior branch through attention. The two branches interact only through the explicit latent alignment loss.

We also assign matched positional IDs to corresponding prior and posterior positions. The i-th prior latent query and the i-th posterior future embedding use the same positional ID. Similarly, corresponding prior and posterior action tokens share matched positional IDs. This design ensures that the hidden states at aligned positions are structurally comparable across the two branches.

Let $h_\ell^{\text{prior}}, h_\ell^{\text{post}} \in \mathbb{R}^{K\times d}$ denote the prior and posterior latent hidden states at the ℓ-th aligned Transformer layer. We apply latent alignment over the last $L = 9$ Transformer layers:

$$\mathcal{L}_{\text{align}} = \frac{1}{L}\sum_{\ell=1}^{L} \frac{1}{Kd} \left\| h_\ell^{\text{prior}} - h_\ell^{\text{post}} \right\|_2^2. \tag{17}$$

Aligning only the last several layers avoids over-constraining early multimodal processing, while still encouraging the final latent reasoning states to absorb future-informed structure.

A.6 Training and Inference

[76] supports a unified multi-task objective that combines text generation, continuous flow matching, and masked motion-token prediction. For simplicity and controlled validation, Being-H0.7 retains only the continuous flow-matching action loss in our experiments. We do not use the text-generation loss or the masked motion-token prediction loss. This isolates the effect of latent reasoning and prior–posterior alignment from additional auxiliary objectives.

For action generation, we use a flow-matching objective. Given a ground-truth action chunk a, Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, and flow time $\tau \in [0, 1]$, we construct the interpolated action $a_\tau = \tau a + (1 - \tau)\epsilon$ and target velocity $u_\tau = a - \epsilon$. Both the prior and posterior branches predict velocity fields:

$$v_\theta^{\text{prior}}(a_\tau, c_t, Q), \qquad v_\theta^{\text{post}}(a_\tau, c_t, z^{\text{post}}). \tag{18}$$

The flow-matching loss is applied to both branches:

$$\mathcal{L}_{\text{FM}} = \left\| v_\theta^{\text{prior}}(a_\tau, c_t, Q) - u_\tau \right\|_2^2 + \left\| v_\theta^{\text{post}}(a_\tau, c_t, z^{\text{post}}) - u_\tau \right\|_2^2. \tag{19}$$

For cross-embodiment data, the validity mask is applied to the supervised action dimensions. The full pretraining objective is

$$\mathcal{L} = \mathcal{L}_{\text{FM}} + w_{\text{align}}\mathcal{L}_{\text{align}} + w_{\text{norm}}\mathcal{R}_{\text{norm}} + w_{\text{rank}}\mathcal{R}_{\text{rank}}, \tag{20}$$

where $\mathcal{R}_{\text{norm}}$ and $\mathcal{R}_{\text{rank}}$ are lightweight anti-collapse regularizers applied to the latent reasoning states. We set the alignment weight to $w_{\text{align}} = 10^{-3}$ and the two regularization weights to $w_{\text{norm}} = w_{\text{rank}} = 10^{-4}$. During downstream post-training, we optimize the action objective and the latent alignment loss on task-specific demonstrations, but remove the anti-collapse regularization.

At inference time, the posterior branch, future encoder, and future observations are removed entirely. The deployed policy uses only the prior sequence $[c_t;\, Q;\, a_\tau]$. Starting from Gaussian noise, we solve the learned flow with a small number of denoising steps to obtain the executable action chunk. In all experiments, we use 4 denoising steps at inference. The model outputs an action chunk in the unified action space, after which the embodiment-specific validity mask and inverse mapping select the executable dimensions for the target robot controller.

B Additional Ablation Studies

Unless otherwise specified, the experiments in this section are trained from scratch on RoboTwin 2.0 [93] under the Clean-Train $\rightarrow$ Randomized-Eval setting (50 tasks, 100 rollouts per task), and vary one factor at a time while keeping the backbone, data, observation history, and optimization settings fixed.

B.1 Controlled Pretraining with Multiple Runs

The alignment ablation in Tab. 4a is trained from scratch on downstream data. To verify that the benefit of joint alignment holds under pretraining and exceeds run-to-run variance, we pretrain all variants under the same constrained setting with 1K hours of robot data and 1K hours of human video, and only vary the posterior and alignment design. We then report the mean $\pm$ standard deviation over five independent downstream post-training runs in Tab. 5. Joint alignment improves over the posterior-only variant by +5.1% on LIBERO-plus and +3.8% on RoboTwin 2.0, and over the single-branch baseline by +5.9% and +7.1%, respectively. These gains are clearly larger than the observed standard deviations of 0.5–1.2%, confirming that the improvement does not arise only from additional pretraining data.

B.2 Fixed Future Target

To distinguish our jointly learned posterior target from directly aligning policy states to fixed future visual features, as in FLARE [43], we add a fixed-target variant on top of the single-branch baseline, where frozen DINOv3 [92] features of future frames serve as the hidden-state target. As shown in Tab. 6, the fixed target provides useful future guidance, but its gain is close to that of adding the posterior branch without alignment. Joint alignment yields a substantially larger gain, supporting the benefit of an adaptive, action-optimized posterior target.

Table 5: Controlled pretraining comparison (1K hours robot data + 1K hours human video). We report mean ± standard deviation over five downstream post-training runs.

Variant	LIBERO-plus (%)	RoboTwin 2.0 (%)
Single-branch baseline	76.8 ± 1.2	46.2 ± 0.8
Posterior branch w/o alignment	77.6 ± 0.5	49.5 ± 1.1
Joint prior–posterior alignment	**82.7 ± 0.8**	**53.3 ± 0.7**

Table 6: Comparison with a fixed future target on RoboTwin 2.0.

Variant	Success Rate (%)
Single-branch baseline	42.7
Fixed DINOv3 future target	44.7
Posterior branch w/o alignment	45.2
Joint prior–posterior alignment	**50.1**

B.3 Sensitivity to Loss Coefficients

Since anti-collapse regularization hurts performance during downstream post-training (Tab. 4b), we vary the coefficients during the constrained pretraining described in Appendix B.1 and disable the regularization during downstream adaptation. Other coefficients remain fixed at their defaults, with $\gamma = 1.0$. As shown in Tab. 7, performance is stable around the defaults. A larger alignment coefficient or smaller regularization coefficients have limited impact, whereas insufficient alignment or excessive anti-collapse regularization causes clearer degradation.

Table 7: Sensitivity to loss coefficients on RoboTwin 2.0. Defaults are $w_{\text{align}} = 10^{-3}$ and $w_{\text{norm}} = w_{\text{rank}} = 10^{-4}$.

Coefficient	$0.2\times$ default	Default	$5\times$ default
w_{align}	51.8	**53.3**	52.9
w_{norm}	52.2	**53.3**	51.2
w_{rank}	52.5	**53.3**	50.9

B.4 Sensitivity to Architectural Choices

We vary the number of latent queries, the number of future frames, and the number of aligned Transformer layers, one at a time, with defaults of 16 latent queries, 4 future frames, and alignment over the final 9 layers. As shown in Tab. 8, performance is stable for 16–32 latent queries and 4–8 future frames; we retain 16 queries and 4 frames as compact defaults since the larger settings provide only marginal gains. Aligning the final 9 layers performs best: aligning fewer layers provides weaker supervision, while aligning 12 layers appears to over-constrain earlier multimodal processing.

B.5 Visual Encoder and Observation History

Visual encoder. We replace the V-JEPA2.1 visual encoder with DINOv3 or the native InternVL3.5 visual encoder. As shown in Tab. 9 (left), V-JEPA2.1 and DINOv3 differ by only 0.4%, showing that the method is not tied to V-JEPA2.1, while the native InternVL3.5 encoder is moderately weaker in this manipulation setting. Since this is a single-run comparison, we interpret the differences as directional rather than as a general ranking of visual encoders. We note that replacing the native encoder may affect the original vision-language alignment of InternVL3.5; as this work focuses on manipulation, we do not separately evaluate general-purpose VLM capabilities.

Observation history. External baselines use different native observation protocols, so their history inputs cannot all be made identical without substantially modifying the original methods. To isolate the contribution of temporal history, we compare single-frame and multi-frame inputs using the same Being-H0.5-style single-branch architecture without pretraining. As shown in Tab. 9 (right),

Table 8: Sensitivity to architectural choices on RoboTwin 2.0. Default settings are underlined.

Latent queries	8 47.8	16 50.1	32 **50.8**	64 48.5
Future frames	2 46.7	4 50.1	8 **50.3**	– –
Aligned layers	4 48.3	9 **50.1**	12 46.6	– –

multi-frame history alone does not provide a consistent improvement, and its effect is substantially smaller than the gap to the full Being-H0.7 model.

Table 9: Left: visual encoder comparison on RoboTwin 2.0. Right: single-frame *vs*. multi-frame input with the single-branch baseline.

Visual Encoder	**RoboTwin 2.0**
V-JEPA2.1	**50.1**
DINOv3	49.7
InternVL3.5 native	48.3

Benchmark	**Single frame**	**Multi-frame**
LIBERO-plus	74.1	73.5
RoboTwin 2.0	42.7	43.6

B.6 Linear Retrieval Probe across Models

To examine whether ordinary VLA representations also contain implicit future information, we apply the linear retrieval probe to the observation tokens and latent-query hidden states of three models trained under the constrained pretraining setting: a Being-H0.5-style single-branch baseline, Being-H0.7 without alignment, and the complete model. The probe is conducted on RoboTwin following the protocol in Section 4.3. As shown in Tab. 10, without alignment, latent queries do not consistently outperform observation tokens. With joint alignment, latent queries improve over the observation tokens of the same model by +5.4%/+4.6%/+4.2% and over the no-alignment latent queries by +4.3%/+6.5%/+7.3% on Top-1/Top-5/Top-10 accuracy. Thus, learnable queries alone do not create the predictive representation; joint alignment makes future information more explicitly organized and linearly accessible at the latent interface used for action generation.

Table 10: Linear retrieval probe across models and representations on RoboTwin.

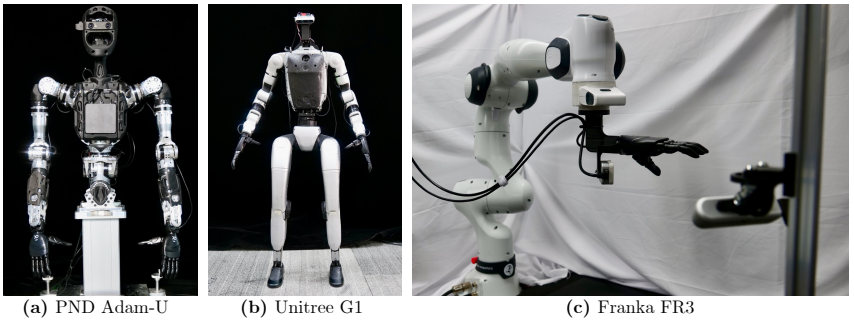
Model	**Representation**	**Top-1 (%)**	**Top-5 (%)**	**Top-10 (%)**
Single-branch	Observation tokens	62.1	83.1	92.3
	Latent queries	60.7	83.5	88.7
W/o alignment	Observation tokens	62.5	85.2	92.1
	Latent queries	61.5	85.7	90.3
Joint alignment	Observation tokens	60.4	87.6	93.4
	Latent queries	**65.8**	**92.2**	**97.6**

C Additional Details of Real Experiments

C.1 Real-robot Task Setup

In this paper, we evaluate real-world manipulation on three robotic platforms: PND Adam-U, Unitree G1, and Franka FR3, all equipped with Linkerbot O6 (6-DoF) dexterous hands. Table 11 summarizes the deployed embodiments. PND Adam-U and Unitree G1 adopt bilateral-hand humanoid configurations, while Franka FR3 provides a single-arm tabletop setup with one external camera and one wrist-mounted camera for visual observation. Unless otherwise specified, all platforms share a unified control interface and online inference infrastructure. This enables the same pretrained policy framework to generalize across heterogeneous embodiments with minimal system-specific modification. For PND Adam-U, the policy controls 19 body DoF together with bilateral Linkerbot

O6 hands. Franka FR3 adopts a simpler tabletop configuration with a 7-DoF arm paired with a single Linkerbot O6 hand, serving as a representative single-arm manipulation platform.



(a) PND Adam-U (b) Unitree G1 (c) Franka FR3

Figure 6: Overview of the real-world embodiments used for evaluation.

Table 11: **Real-robot embodiments for Being-H0.7.** All evaluated platforms are equipped with Linkerbot O6 (6 DoF) hands and share a unified state-action interface.

Platform	Type	Body DoF	Hand	Total DoF	Cameras	Policy Freq.
PND Adam-U	Upper-body humanoid	19	Linkerbot O6 (6 DoF)	31	2 ego-view cameras	20 Hz
Unitree G1	Bimanual humanoid	14	Linkerbot O6 (6 DoF)	26	1 ego-view camera	10 Hz
Franka FR3	Single-arm tabletop	7	Linkerbot O6 (6 DoF)	13	1 external + 1 wrist	20 Hz

For Unitree G1, the policy still exposes the same 26-DoF action interface used by the rest of our deployment stack. To enable stable whole-body execution, we integrate a pretrained AMO controller [94] as a balance-aware low-level control module. AMO operates the 50 Hz Unitree body-control loop and predicts lower-body and waist motions conditioned on the latest dual-arm targets. The resulting lower-body commands are then composed with the policy-driven upper-body motions to form the final executable whole-body command. Meanwhile, the bilateral Linkerbot O6 hands remain controlled through the same unified hand interface used by the other embodiments. This hierarchical integration allows the high-level policy to focus on manipulation-centric upper-body behavior while delegating balance stabilization and locomotion coordination to the low-level controller.

Table 12: **Real-robot task set for Being-H0.7.** Each task is assigned a primary suite and optional overlap tags; while the prompt specifies the instruction provided to the policy during evaluation.

ID	Platform	Primary Suite	Overlap Tags	Prompt
Task01	Franka FR3	Dynamic Scene	Motion Reasoning	Catch the fast rolling ball on the table before it leaves the capture area.
Task02	PND Adam-U	Physical Reasoning	Long Horizon	Use the pipette to transfer the liquid from the source container into the target container accurately.
Task03	Unitree G1	Motion Reasoning	Dynamic Scene, Physical Reasoning	Hit the ball with the racket so that it lands in the marked target area.
Task04	PND Adam-U	Physical Reasoning	Long Horizon	Pour the liquid from the beaker through the funnel into the receiving container.
Task05	Unitree G1	Dynamic Scene	Motion Reasoning, Physical Reasoning	Pour the objects in the cup into the moving target container.
Task06	Franka FR3	Dynamic Scene	Motion Reasoning, Long Horizon, Generalization	Pick the cargo from the moving conveyor and place it on the correct shelf level.
Task07	PND Adam-U	Physical Reasoning	Long Horizon, Generalization	Fold the garment neatly into the target folded shape.
Task08	Unitree G1	Long Horizon	Dynamic Scene, Motion Reasoning, Generalization	Scan the package on the moving conveyor and sort it to the correct destination.
Task09	Unitree G1	Long Horizon	Physical Reasoning, Generalization	Insert the shoe tree into the shoe and place the prepared shoe onto the conveyor.
Task10	PND Adam-U	Long Horizon	Dynamic Scene, Generalization	Pick the shoe from the conveyor and pack it into the shoebox.
Task11	Franka FR3	Generalization	Long Horizon, Physical Reasoning	Pick the tabletop object and place it into the correct drawer level.
Task12	Franka FR3	Physical Reasoning	Motion Reasoning, Long Horizon	Pick up the hammer and drive the nail into the hole.

C.2 Inference Infrastructure for Efficient Deployability.

Figure 7 reports the system-level inference cost under a unified deployment infrastructure. To further improve real-world deployability, we employ the latency-aware **Universal Async Chunking (UAC)** mechanism [76] on the client side, implemented as asynchronous real-time chunked control. As a result, the client maintains a thread-safe action buffer together with an online estimate of how many control steps will be consumed before the next action chunk becomes available. A control

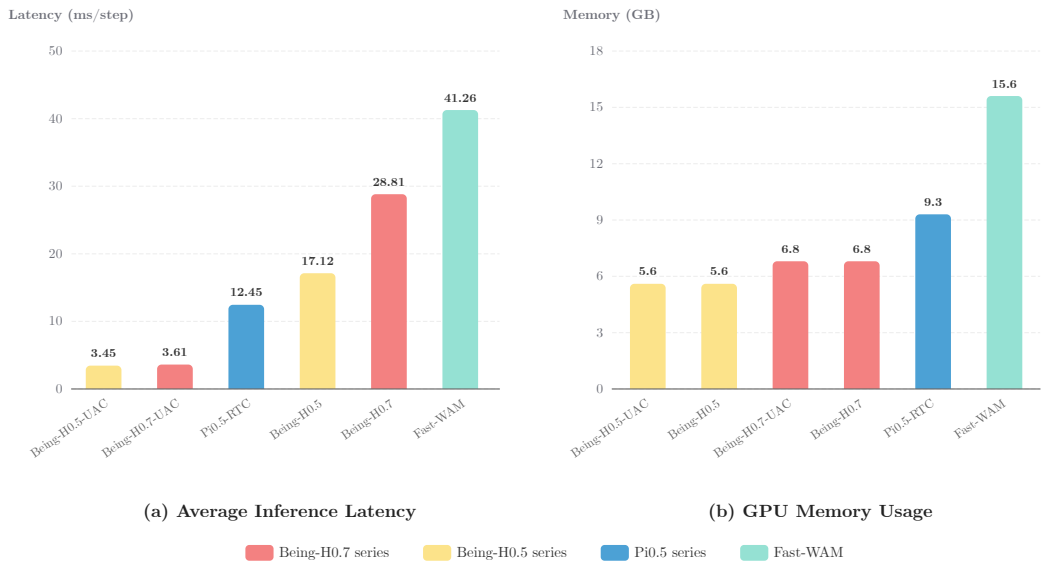


Figure 7: Inference cost measured in the real-world deployment stack. We report it as a system-level view of deployability alongside the main success-rate comparison.

thread continuously executes actions from the committed prefix at the robot control frequency, while a parallel inference thread monitors the remaining buffer length. Once the buffer occupancy falls below a predefined threshold, the inference thread fetches the latest observations and asynchronously requests the next action chunk from the server. A key property of UAC is that it never rewrites the already committed action prefix. Instead, newly predicted actions are stitched only into the future suffix after compensating for the estimated inference delay. This *prefix-lock / suffix-update* mechanism introduces latency by model inference, communication, and system scheduling jitter without modifying the policy interface itself. As a result, the same deployment protocol can be consistently applied across platforms with different control frequencies and embodiment-specific action spaces. In practice, UAC serves as the infrastructure layer that transforms chunked prediction into smooth, continuous control. It preserves temporal consistency, reduces visible control stutter, and stabilizes online execution under network and scheduler jitter. Empirically, UAC-enabled Being-H variants achieve a low-latency regime of approximately 3–4 ms per control step, while maintaining nearly identical GPU memory usage to their non-UAC counterparts. This additional timing margin substantially improves responsiveness in dynamic manipulation scenarios and produces smoother online rollouts at the robot interface. Together with the benchmark results above, the inference-cost analysis demonstrates that the performance gains of Being-H0.7 are realized within an efficient and practically deployable inference loop, rather than relying on an excessively heavyweight deployment stack.

C.3 Real Experiments Rollouts

Here, we provide the rollouts of Being-H0.7 on the 12 real experiment tasks.

C.4 Implementation Details of Linear Retrieval Probe

We study whether the prior latent contains predictive information about future visual observations using a linear-probe retrieval setup. For each query, the target is the DINOv3 [92] image embedding of the future observation. We consider four source-feature baselines: *Latent*, which uses the concatenation of h_ℓ^{prior}; *Noise*, which replaces the latent vector with a random noise vector of the same dimensionality; *Image+Latent*, which concatenates the current-frame image embedding with the latent vector; and *Image+Noise*, which concatenates the current-frame image embedding with a random noise vector.

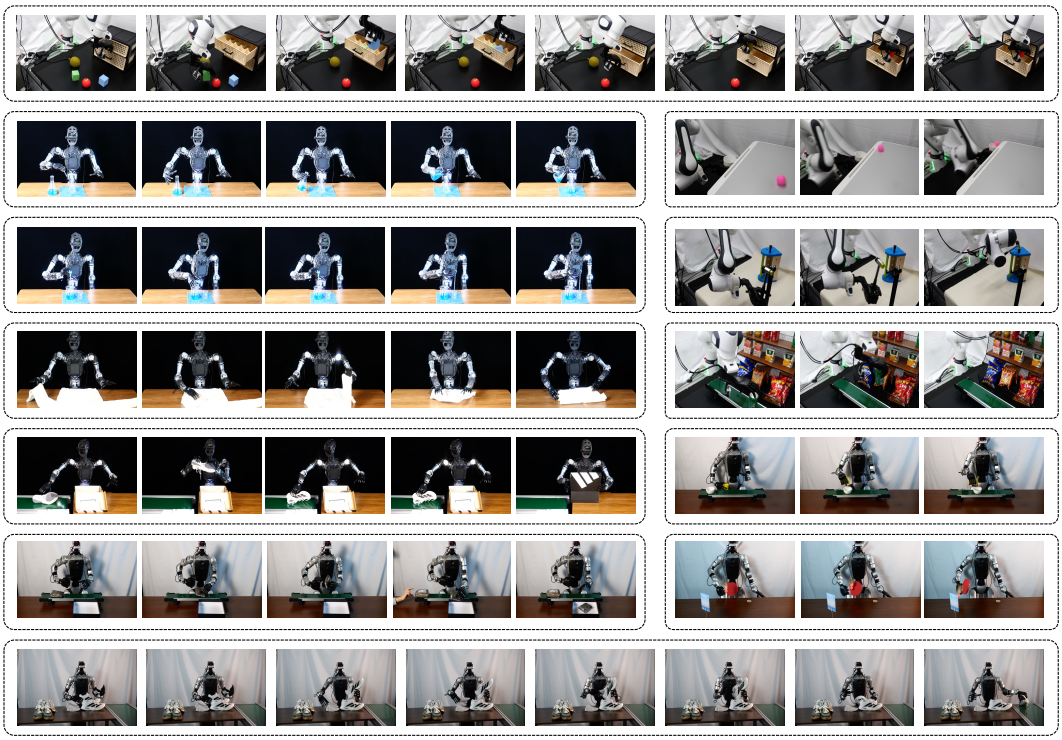


Figure 8: Rollouts of Being-H0.7 on 12 real-world tasks.

Training is conducted on a mixed dataset of real-world robot videos collected from seven tasks (Task 01, Task 03, Task 05, Task 06, Task 08, Task 10, Task 12). For all four baselines, we train a single linear projection from the source feature space to the DINOv3 image embedding space. Training uses the contrastive objective: given a minibatch of projected source features and their corresponding future image embeddings, the matching source-target pair is treated as the positive pair, while all other target embeddings in the same minibatch are treated as negatives. We train with a batch size of 2048 for 50 epochs using AdamW with learning rate 3×10^{-4}, weight decay 10^{-4}, and contrastive temperature 0.07. We linearly warm up the learning rate over the first 5% of optimization steps and then apply cosine decay to zero for the remaining steps. The target frame is the frame one second after the current frame. Image features are extracted using a frozen DINOv3 encoder in `bfloat16` precision.

C.5 Implementation Details of Visualization

For the qualitative visualization experiments, Wan2.2-TI2V-5B [10] is adopted as the video decoder to synthesize the future rollout conditioned on the current observation and the prior latent of Being-H0.7. Specifically, the prior latent is injected into the video generator through a cross-attention conditioning mechanism, enabling the generator to decode a 1-second future video from the current visual state. Training is performed with LoRA [95] on a mixed dataset of real-world robot videos collected from seven tasks (Task 01, Task 03, Task 05, Task 06, Task 08, Task 10, Task 12), using a LoRA rank of 32. The training is conducted for 50 epochs with a learning rate of 1×10^{-4}. The generated video contains 21 frames at a spatial resolution of 256×256.

C.6 Limitation Discussion

One limitation of the current study is that Being-H0.7 has not yet been validated on high-degree-of-freedom dexterous hand manipulation, which involves substantially more complex contact dynamics and control challenges than the settings considered here.

C.7 Computational Cost

Being-H0.7 is pretrained using GPU resources comparable to popular embodied foundation models such as $\pi_{0.5}$, enabling fair comparison under similar compute budgets. Specifically, each checkpoint requires approximately 15K GPU hours on A800-80G GPUs, substantially less than existing world-action models such as Cosmos-Policy.

C.8 Licenses

The simulation evaluations in this work use LIBERO, LIBERO-plus, RoboCasa, and CALVIN. These benchmarks publicly release their codebases under the MIT License and their datasets or assets under CC BY 4.0 licenses. The implementation is built upon the Being-H repository, which is distributed under the Apache-2.0 License. The real-robot demonstrations used in this work consist of internally collected robot-control trajectories for manipulation research. These data do not include personally identifiable information or human-subject annotations, and are used solely for robot policy training and evaluation.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize the paper's key contributions and the scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: In the Appendix.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: No theoretical results are involved in the paper.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Details are clearly included in the paper and in the Appendix.

   Guidelines:

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

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

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

Answer: [No]

Justification: Code and data have not been released so far.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: In the experiments section and the Appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: For the key ablation on joint alignment, we report mean $\pm$ standard deviation over five independent downstream post-training runs under a controlled pretraining setup (Appendix B.1). Repeating large-scale pretraining for the main results is computationally prohibitive, so main benchmark results are from a single run aggregated over a large number of rollouts.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The main resources consumption are included in the Appendix.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conform with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: Not relevant.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: Not relevant.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All properly cited.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [No]

    Justification: Document will be provided after publication.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Not relevant.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: Not relevant.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The core method does not involve LLMs as important components.

Guidelines:

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