DiT4DiT: Jointly Modeling Video Dynamics and Actions for Generalizable Robot Control

Teli Ma[1,2] **Jia Zheng**[1,2] **Zifan Wang**[1,2] **Chunli Jiang**[1]
Andy Cui[1] **Junwei Liang**[2,3,*] **Shuo Yang**[1,*]

[1]Mondo Robotics
[2]The Hong Kong University of Science and Technology (Guangzhou)
[3]The Hong Kong University of Science and Technology
[*]Corresponding authors, co-advising
Project page: https://dit4dit.github.io/

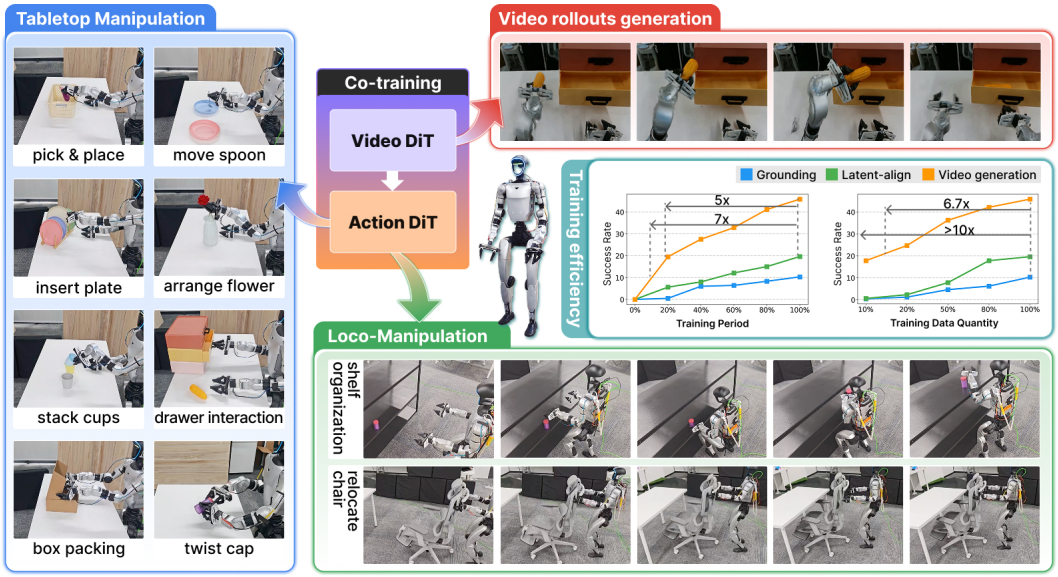

Figure 1: **DiT4DiT: a unified Video-Action Model for generalist robot control.** A Video DiT and an Action DiT are *co-trained* under a dual flow-matching objective. The resulting policy drives diverse tabletop manipulation (**left**) and real-time humanoid whole-body loco-manipulation on a Unitree G1 (**bottom**). The Video DiT also produces physically plausible future rollouts (**top right**). A controlled study (**middle right**) shows that the video-generation proxy yields up to $7\times$ faster convergence and $>10\times$ higher data efficiency than grounding [1] and latent-alignment [2] baselines on the 24-task RoboCasa-GR1 [3] suite.

Abstract

Vision-Language-Action (VLA) models inherit their visual backbone from static image-text pretraining, leaving physical dynamics to be learned from scarce action data. Generative video models, by contrast, already encode motion, contact, and implicit physics at internet scale. We introduce **DiT4DiT**, a Video-Action Model (VAM) that couples a Video Diffusion Transformer with an Action Diffusion Transformer and co-trains them under a dual flow-matching objective with a tri-timestep scheme for video supervision, action supervision, and feature extraction.

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

The key insight is that the policy does not need to *generate* a future video: we intercept the Video DiT's hidden state at a single fixed flow timestep, turning a multi-step video rollout into a one-shot feature extraction consumed by the Action DiT. Across benchmarks, DiT4DiT sets new SOTA averages on LIBERO (98.6%) and the 24-task RoboCasa-GR1 suite (56.7%), outperforming strong VLA and video-based baselines. On a real Unitree G1, DiT4DiT reaches 73.8% on eight tabletop tasks and 72.2% on three whole-body loco-manipulation tasks, running up to 12× faster than prior VAM baselines while also exceeding them in accuracy. To our knowledge, this is the *first* video-generation-based policy to run humanoid whole-body control at real-time rates. It further generalizes zero-shot to unseen object categories and scene variations. Together, these results indicate that video-dynamics priors, accessed through a single forward pass, are a practical and scalable foundation for generalist robot policies.

1 Introduction

Building generalist robot policies requires backbones that encode not only *what* is in a scene, but also *how* it evolves under action. Vision-Language-Action (VLA) models [4, 5, 6, 7, 8, 1, 9] inherit the semantic priors of large Vision-Language Models [10, 11, 12, 13] and generalize well across instructions and object categories. But their backbones are pretrained almost exclusively on *static* image-text pairs: physical dynamics, contact events, and temporal state transitions must be learned entirely from scarce robot trajectories. Video Generation Models (VGMs) [14, 15, 16, 17], in contrast, are trained at internet scale to synthesize temporally coherent futures, so their latent space already encodes motion, causality, and physical structure.

This prompts a first question: *can video dynamics serve as a proxy objective for learning action?* If a video backbone already encodes motion and contact, then co-training it with an action head should turn the video-prediction loss into a dense, physics-aware supervisory signal—one that shapes the same latent space the policy reads from. Prior attempts to enrich visual supervision for action like object-level grounding [1] or VLM-centric latent alignment such as FLARE [2], instead rely on semantic proxies, and struggle to capture continuous pixel-level dynamics. A controlled study on the 24-task RoboCasa-GR1 [3] suite (Fig. 1; details in Appendix A.1) confirms the hypothesis: under matched trainable-parameter budgets, our video-dynamics signal yields up to 7× faster convergence and up to 10× higher data efficiency than both grounding and latent-alignment baselines, with consistent gains as data scales. Learning to predict future pixels teaches the backbone more about control per demonstration than any semantic proxy we tested.

Having established that video supervision helps, two questions remain: how to adapt the video backbone for control during training, and how to use it efficiently at test time. A natural design is the *imagine-then-execute* recipe adopted by recent work [18, 19, 20, 21, 22, 23, 24, 25, 26]: the video model denoises future frames over many steps, and actions are predicted from the rollout. Two distinct issues need to be addressed. *Cost 1 (training-time representation):* the video backbone is typically frozen or only loosely coupled, so the action objective cannot reshape its latent space for control. *Cost 2 (inference-time latency):* full multi-step video denoising sits on the control loop—existing VAM/WAM policies run at 0.5–1.9 Hz closed-loop, ruling out the high control rates required by highly dynamic tasks such as humanoid whole-body manipulation, where the robot must close the loop across tens of joints. We argue that *while pixel prediction is a powerful training signal, actually imagining future frames at test time is unnecessary*: what the policy needs is the *representation* the video model acquires during training, not the pixels it would synthesize at inference. We therefore ask: *can a single, deterministic forward pass of a video backbone—shaped jointly by video and action losses—replace the multi-step rollout?*

DiT4DiT answers yes. It is a unified Video-Action Model (VAM) with a **dual-DiT** architecture: a bidirectional Video Diffusion Transformer [27] initialized from Cosmos-Predict2.5-2B [16] and an Action DiT. The two are trained end-to-end under a **dual flow-matching** objective, so video synthesis and action generation share one optimization. Actions are *not* conditioned on a sampled future video; instead, a forward hook intercepts the Video DiT's hidden state at a *single, fixed* flow timestep τ_f, producing deterministic features that the Action DiT consumes. To make joint training work, we introduce a **tri-timestep scheme** that decouples three diffusion processes that prior joint-training works entangle: a uniform τ_v for the video loss, a Beta-distributed τ_a for the action decoder, and a

fixed τ_f for feature extraction. The result is a policy that keeps the full physical priors of the video backbone, yet runs closed-loop control without ever rolling out a video at test time.

This combination of co-trained representation and single-pass inference unlocks a regime prior video-based policies could not reach: **real-time humanoid whole-body control**. On a Unitree G1 driven from a single egocentric camera, DiT4DiT controls the 31-DoF robot through a hierarchical controller, predicting 23-D actions comprising 16 upper-body targets and 7 high-level lower-body commands. To our knowledge, this is the first video-generation-based policy to run whole-body humanoid control at real-time rates. In simulation, DiT4DiT sets a new state of the art on both LIBERO [28] (98.6% average success) and the 24-task RoboCasa-GR1 [3] Tabletop suite (56.7%), outperforming strong VLA and VAM baselines; among VAM/WAM methods, DiT4DiT simultaneously attains the highest success rate and the highest closed-loop speed (6.0 Hz vs. 0.5–1.9 Hz for prior video-based policies). In real-world G1 deployments, it surpasses both pre-trained $\pi_{0.5}$ [29] and GR00T-N1.5 [30] on precision-critical tasks (*Arrange Flower*, *Stack Cup*, *Twist Cap etc.*) and shows robust zero-shot generalization to unseen objects, category changes, and quantity variations.

Our contributions are:

- **A jointly trained Video-Action Model.** DiT4DiT unifies a pre-trained Video DiT and an Action DiT under a dual flow-matching objective, with a tri-timestep scheme that decouples τ_v, τ_a, and τ_f to reconcile the video loss's need for the full noise schedule with the action policy's need for a stable, deterministic feature signal.
- **Single-pass control from a video backbone.** By reading hidden states at a fixed τ_f rather than denoising future frames at test time, DiT4DiT removes multi-step video rollout from the control loop, enabling video-model-driven real-time whole-body humanoid control for the first time.
- **State-of-the-art performance and strong generalization capability.** By establishing video generation as a proxy task that substantially improves data and training efficiency, DiT4DiT achieves new state-of-the-art average success rates on the LIBERO benchmark, the RoboCasa-GR1 tabletop suite, and real Unitree G1 tabletop and whole-body control tasks, and shows robust zero-shot generalization to unseen objects, category changes, and quantity variations.

2 Related Works

Prior work on robot foundation models splits into two distinct families that differ in *what* the backbone was pretrained to predict: VLA models, built on static image-text backbones, and video-based policies, built on generative video models.

Vision-Language-Action (VLA) models: RT-2 [5], OpenVLA [6], UniVLA [31], CogVLA [32], GR00T [1, 9], and the π family [7, 29] inherit semantic priors from VLM pretraining, but their backbones see only *static* image-text pairs, so all physical and temporal structure must be learned during robot fine-tuning. DiT4DiT instead starts from a pre-trained video diffusion model [33, 34, 16, 15, 14], whose next-frame objective already encodes spatiotemporal dynamics.

Video-Generation-Based policies works in this family differ along two axes: **(A1)** Is the video backbone updated jointly with the action part (*frozen* vs. *co-trained*)? **(A2)** Does control require multi-step video denoising at test time (*full rollout* vs. *single-pass hook*)?

- **Frozen backbone, full rollout.** Visual-foresight [35, 36, 37, 38] and VPP [39] denoise future frames under a frozen video model and derive actions from the rollout, paying for full denoising at every step.
- **Frozen backbone, single-pass.** Mimic-video [26] conditions a separate flow-matching decoder on the hidden state at a single intermediate flow step of a frozen video model, avoiding multi-step denoising at deployment, but leaving the backbone's latent space untouched by the action objective.
- **Co-trained backbone, full rollout.** World-Action Model (WAM) methods [40, 23, 24, 41, 42, 19, 20, 21, 22, 43, 44] denoise visual dynamics and actions jointly in a shared latent space. Cosmos Policy [25] fine-tunes a video diffusion model to emit actions as native latent

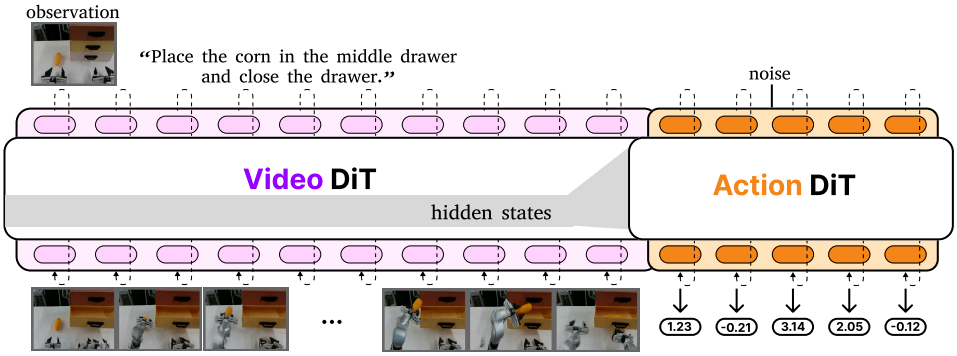

Figure 2: **Overview of DiT4DiT.** A Video DiT predicts future dynamics from the current observation and language goal; its intermediate activations, extracted at a fixed flow timestep τ_f, condition an Action DiT that denoises the control trajectory. Both DiTs are jointly optimized under a dual flow-matching objective.

frames. These methods retain full gradient coupling but still run multi-step denoising at test time.

- **Co-trained backbone, single-pass (ours).** DiT4DiT occupies the remaining cell: a dual flow-matching objective updates both the video and action models, while actions are inferred from a single hook at a *fixed* flow timestep τ_f of video generation. The tri-timestep scheme (τ_v, τ_f, τ_a decoupled) makes this feasible, giving each module its own schedule without entangling them.

3 DiT4DiT: Jointly Modeling Video Dynamics and Actions

DiT4DiT is a Video-Action Model (VAM) that couples a generative Video DiT and an Action DiT under a joint flow-matching objective. Rather than sampling full future videos, we treat the Video DiT as a stateful feature extractor: actions are decoded from intermediate generative features, grounding control in the same distribution that drives video synthesis.

3.1 Preliminaries

Flow matching. Flow Matching (FM) [45] regresses a time-dependent velocity field $v_\theta(x, \tau)$ that transports samples along a probability path between noise $p_1 = \mathcal{N}(0, I)$ and data p_0. We adopt the optimal-transport linear interpolant

$$x_\tau = (1-\tau)\, x_0 + \tau\, z, \quad \tau \in [0,1], \quad x_0 \sim p_{\text{data}},\ z \sim \mathcal{N}(0, I), \tag{1}$$

so τ=0 returns clean data and τ=1 returns noise. The target velocity is $v^*(x_\tau, \tau) = dx_\tau/d\tau = z - x_0$, and training minimizes

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{x_0,\, z,\, \tau \sim \mathcal{U}[0,1]} \left[\| v_\theta(x_\tau, \tau) - (z - x_0) \|^2 \right]. \tag{2}$$

At inference we integrate the learned ODE $dx/d\tau = v_\theta(x, \tau)$ from $x_1 \sim \mathcal{N}(0, I)$ back to τ=0 via a first-order Euler step with $\Delta\tau = 1/N$:

$$x_{\tau - \Delta\tau} = x_\tau - \Delta\tau \cdot v_\theta(x_\tau, \tau). \tag{3}$$

3.2 Problem Formulation

At decision time t, let $\mathbf{O}_{\text{cond}} \in \mathbb{R}^{T_{\text{cond}} \times 3 \times H \times W}$ be the conditioning frame sequence and $\mathbf{O}_{\text{future}} \in \mathbb{R}^{T_v \times 3 \times H \times W}$ the ground-truth future frame sequence. Here T_{cond} and T_v denote the respective numbers of frames, and H and W are the image height and width. The action chunk $\mathbf{a}_t \in \mathbb{R}^{T_a \times d_a}$ has horizon T_a and per-step action dimension d_a; l denotes the language goal.

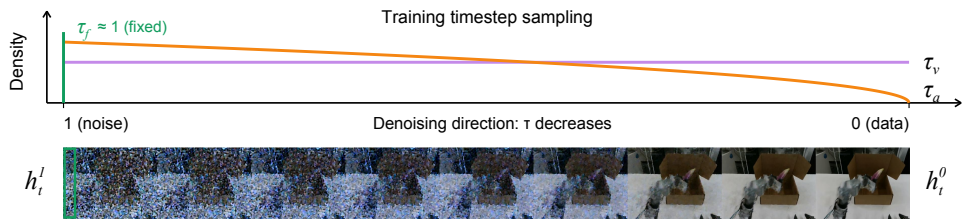

Figure 3: **Asymmetric tri-timestep design.** Training sampling densities are shown over the nominal interval $[0, 1]$: video timesteps τ_v are uniform, while Beta-based action timestep sampling favors higher noise levels. Video features are read once at a fixed high-noise timestep $\tau_f \approx 1$. At inference, these features condition all action denoising steps from noise ($\tau_a = 1$) to data ($\tau_a = 0$).

Unlike VLA policies that learn a direct mapping $\pi_\theta(\mathbf{a}_t \mid \mathbf{O}_{\text{cond}}, l)$, DiT4DiT follows a *predict-video-dynamics–then-invert* paradigm:

$$\mathbf{O}_{\text{future}} \sim p_v(\cdot \mid \mathbf{O}_{\text{cond}}, l), \tag{4}$$

$$\mathbf{a}_t \sim p_a(\cdot \mid \mathbf{O}_{\text{cond}}, \mathcal{H}(\mathbf{O}_{\text{future}}{}^{\tau_v})), \quad \text{where } \mathbf{O}_{\text{future}}{}^{\tau_v} \xrightarrow{\tau_v \to 0} \mathbf{O}_{\text{future}}, \tag{5}$$

where $\mathbf{O}_{\text{future}}{}^{\tau_v}$ is the intermediate state of the future frame sequence at flow step τ_v, and $\mathcal{H}$ extracts hidden states from this generative process. Training jointly models the video–action distribution $p_{va}(\mathbf{O}_{\text{future}}, \mathbf{a}_t \mid \mathbf{O}_{\text{cond}}, l)$.

3.3 Dual-DiT Architecture

Video DiT. We initialize the video backbone from Cosmos-Predict2.5-2B [16], comprising a causal spatio-temporal VAE and a flow-matching diffusion transformer. The VAE compresses $\mathbf{O}_{\text{cond}}, \mathbf{O}_{\text{future}}$ into latents $\mathbf{z}_t^0, \mathbf{z}_{t+1}^0$; the DiT is conditioned on language embeddings from Cosmos-Reason1 [46]. Instead of consuming the fully denoised output [27], we install a forward hook on a single transformer block to intercept its activations at a fixed flow timestep τ_f, producing visual tokens for control:

$$\mathbf{h}_t^{\tau_f} = \mathcal{H}\big[v_\theta^{\text{video}}\big]\big(\mathbf{z}_{t+1}^{\tau_f}, \tau_f \mid \mathbf{z}_t^0, l\big), \quad \mathbf{z}_{t+1}^{\tau_f}\big|_{\tau_f \to 1} \sim \mathcal{N}(0, I), \tag{6}$$

where $\mathcal{H}[\cdot]$ reads the hidden state of the velocity network v_θ^{video} on a single forward pass. We fix τ_f at the first step of the flow-matching schedule ($\tau_f \approx 1$, i.e., the pure-noise end), so the future-frame latent is (near-)pure Gaussian noise and the Video DiT is queried in its *one-shot imagination* regime: given the clean observation $\mathbf{z}_t^0$ as conditioning and near-pure noise for the future, a single forward pass produces activations that capture where the scene is *about to go*, without ever denoising those frames (see Sec. 4.6 for the layer-choice ablation).

Action DiT. The control head is a 16-block flow-matching transformer with Adaptive Layer Normalization [27] for diffusion-timestep injection and interleaved self/cross-attention to $\mathbf{h}_t^{\tau_f}$. Its input sequence concatenates embedded proprioceptive state s and noisy action tokens; a linear head outputs the action-velocity field, which is integrated during inference to recover the trajectory.

3.4 Joint Training via Dual Flow Matching

Tri-timestep scheme. Jointly training a generative video model and a feature-conditioned policy requires reconciling two conflicting demands: the video loss wants exposure to *all* noise levels, while the action policy wants a *stable*, consistent visual signal. We resolve this with three independent timesteps (Fig. 3):

- **Video (τ_v)** is sampled from $\mathcal{U}[0, 1]$ at every step, giving the backbone the full denoising trajectory it needs to learn future-frame synthesis.
- **Feature extraction (τ_f)** is fixed at the first step of the flow schedule ($\tau_f \approx 1$, the pure-noise end), giving the action head a deterministic, high-variance operating point in which the backbone is forced to synthesize plausible futures from scratch at every query.

- **Action (τ_a)** uses $\sigma \sim \mathrm{Beta}(1.5,\, 1.0)$ and $\tau_a = \sigma / c_a$, with $c_a = 0.999$, during training, favoring higher-noise action states. This scaling permits training timesteps slightly above 1 (up to $1/c_a$). At inference, the action flow integrates from noise to data over all N_a steps, conditioned on the same features extracted at τ_f.

Decoupling these timesteps lets the action decoder learn inverse dynamics independently of the video backbone's noise schedule, while still consuming stable features.

Objective. The text encoder and VAE are frozen; only the two DiTs are updated (Algorithm 1). Training minimizes a joint flow-matching loss:

$$\mathcal{L}_t^{\text{total}} = \underbrace{\mathbb{E}_{\tau_a,\epsilon}\Big[\big\| v_\phi^{\text{action}}\big(\mathbf{a}_t^{\tau_a}, \tau_a \mid \mathbf{h}_t^{\tau_f}, s\big) - (\epsilon - \mathbf{a}_t^0)\big\|^2\Big]}_{\text{Action Flow Matching}} + \lambda \underbrace{\mathbb{E}_{\tau_v,z}\Big[\big\| v_\theta^{\text{video}}\big(\mathbf{z}_{t+1}^{\tau_v}, \tau_v \mid \mathbf{z}_t^0, l\big) - (z - \mathbf{z}_{t+1}^0)\big\|^2\Big]}_{\text{Video Flow Matching}}, \quad (7)$$

with λ balancing the two signals (default $\lambda = 1$). The action branch is conditioned on proprioception s and the hook features $\mathbf{h}_t^{\tau_f}$ from Eqn. 6, so minimizing both terms jointly forces the backbone to embed control-relevant dynamics rather than pixel-only cues.

3.5 Inference

At deployment, DiT4DiT supports three modes: video prediction, action prediction, or both concurrently (Algorithm 2).

Video DiT sampling. $\mathbf{O}_{\text{cond}}$ is VAE-encoded to $\mathbf{z}_t^0$; starting from $\hat{\mathbf{z}}_{t+1} \sim \mathcal{N}(0, I)$, the DiT integrates its velocity field over N_v Euler steps conditioned on $\mathbf{z}_t^0$ and l, and the VAE decoder projects the final latent back to pixels.

Action DiT sampling. Crucially, control does *not* require a full video rollout. We run a *single* forward pass of the video backbone at τ_f to produce $\mathbf{h}_t^{\tau_f}$ via the hook, then integrate the action velocity field from $\hat{\mathbf{a}}_t \sim \mathcal{N}(0, I)$ over N_a steps, conditioned on $\mathbf{h}_t^{\tau_f}$ and s. This single-pass conditioning removes the multi-step video-generation bottleneck, enabling high-frequency closed-loop control.

4 Experiments

We evaluate DiT4DiT on three questions: (1) how it compares with VLA baselines in simulation and on a real humanoid; (2) whether a VAM needs to roll out the full video-generation process before predicting actions, or whether intermediate denoising features already suffice; and (3) how well it generalizes under distribution shifts. We answer these through simulation benchmarks, real-world deployment on a Unitree G1, and ablation studies.

4.1 Experiment Setup

LIBERO benchmark. The LIBERO benchmark [28] evaluates manipulation on a Franka Emika Panda robot across four suites: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long. These suites test generalization to new spatial layouts, unseen objects, language instructions, and long-horizon behaviors, respectively. Each suite contains 500 demonstrations evenly distributed over 10 tasks, and each task is evaluated over 50 rollouts.

RoboCasa-GR1 tabletop benchmark. To evaluate a more complex embodiment, we use the RoboCasa-GR1 benchmark [1] on the Fourier GR1 humanoid (29-DoF action space), which covers 24 tabletop tasks (6 articulated-object interactions, e.g., *opening microwaves*, and 18 pick-and-place tasks with novel objects) with 1,000 teleoperated demonstrations per task. We report category-wise and overall average success rates; full embodiment and evaluation protocol are in Appendix A.3.

Real-world tabletop evaluation. To validate real-world performance, we deploy our policy on a Unitree G1 humanoid, predicting a continuous 16-DoF action space consisting of two 7-DoF arms and ALOHA2 grippers. The policy uses only ego-view camera observations. We evaluate it on

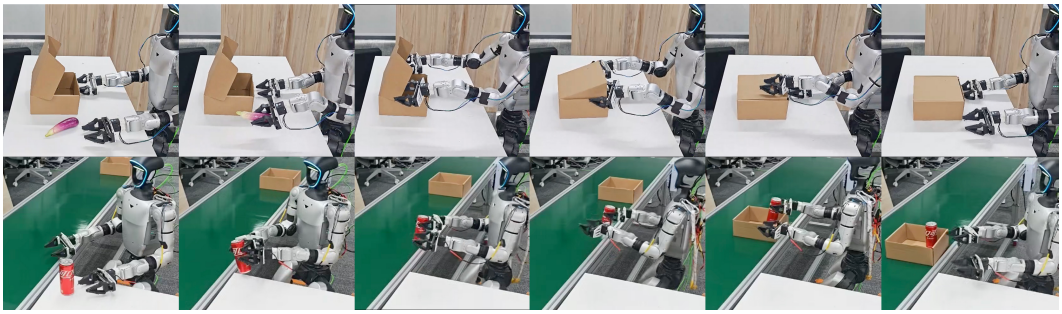

Figure 4: **Representative real-world tasks on the Unitree G1.** *Box packing* (**tabletop**) stresses long-horizon task planning across a multi-step folding sequence and fine-grained spatial reasoning for aligning and inserting the cardboard flaps and tabs to securely seal the box. *Assembly line work* (**whole-body**) further require turning and locomoting to adjust standing distance for reachable grasps, and reasoning about the timing of items arriving on the conveyor so that the policy must run in closed loop at real-time rates. Together, these tasks evaluate spatial understanding, long-horizon planning, locomotion–manipulation coordination, and reactive real-time control.

eight household manipulation tasks (Fig. 8): *pick and place, arrange the flower, stack up the cups, insert plate into the rack, box packaging, move the spoon*, *drawer interaction* and *twist cap*. Each task contains 200 human demonstration episodes and is evaluated over 20 real-world rollouts, with success rate as the main metric.

Real-world whole-body evaluation. We further deploy DiT4DiT on the Unitree G1 for whole-body loco-manipulation using an upper/lower decoupled controller: DiT4DiT directly predicts 16 upper-body joint targets and a compact 7-D high-level command for the lower body (torso RPY, base height, and base velocity v_x, v_y, ω_z), while a separate RL policy converts this high-level command into 15-DoF leg and waist joint targets at a higher rate, handing those off to the 500 Hz motor PD loop. We evaluate three loco-manipulation tasks (example rollouts in Fig. 1&4): *shelf organization*, *relocate chair*, and *assembly line work* (200 demonstrations and 10 rollouts per task; see Appendix A.6 for full setup).

Baselines. We compare DiT4DiT against both state-of-the-art VLA baselines, including the π [7, 29] and GR00T [1, 9] series, and VAM/WAM baselines, including mimic-video [26], Motus [24], and Cosmos Policy [25]. We also build a parameter-matched baseline, Qwen3DiT, by combining Qwen3-VL [13] 2B with the same Action DiT used in DiT4DiT. On both LIBERO and RoboCasa-GR1, DiT4DiT and Qwen3DiT are trained *from scratch* on the target dataset without action-data pre-training for a controlled comparison. More details are in Appendix A.7.

4.2 Main Results

Table 1: **Success rates (%) on the four evaluation suites of the LIBERO simulation benchmark.** Bold numbers indicate the highest performance in each category.

	Spatial SR (%)	Object SR (%)	Goal SR (%)	Long SR (%)	Average SR (%)
VLA					
π_0 [7]	96.8	98.8	95.8	85.2	94.2
UniVLA [31]	96.5	96.8	95.6	92.0	95.2
$\pi_{0.5}$ [29]	**98.8**	98.2	98.0	92.4	96.9
OpenVLA-OFT [47]	97.6	98.4	97.9	94.5	97.1
CogVLA [32]	98.6	98.8	96.6	95.4	97.4
GR00T-N1.5 [30]	96.2	94.0	96.0	90.0	94.1
Qwen3DiT	98.0	98.8	96.0	93.6	96.6
VAM / WAM					
mimic-video [26]	96.6	97.2	96.2	92.0	95.5
Motus [24]	96.8	99.8	96.6	**97.6**	97.7
Cosmos Policy [25]	98.1	**100.0**	98.2	**97.6**	98.5
DiT4DiT	98.4	99.6	**98.6**	**97.6**	**98.6**

Table 2: **RoboCasa-GR1 tabletop tasks evaluation results** (success rate (%)). Bold numbers indicate the highest performance in each category.

	Articulated Obj SR (%)	Novel Obj SR (%)	Average SR (%)	Params (B)	Speed (Hz)
		VLA			
π_0 [7]	43.7	42.3	42.7	3.3	12.0
$\pi_{0.5}$ [29]	43.3	43.0	43.1	2.3	13.0
GR00T-N1.5 [30]	45.0	42.2	42.9	2.7	13.0
GR00T-N1.6 [9]	46.3	48.3	47.8	3.3	10.0
Qwen3DiT	37.3	35.8	36.2	2.3	9.0
		VAM / WAM			
mimic-video [26]	45.3	44.2	44.5	2.5	1.9
Motus [24]	49.7	47.1	47.8	5.9	0.5
Cosmos Policy [25]	46.0	46.7	46.5	2.0	0.7
DiT4DiT	**56.3**	**56.8**	**56.7**	2.2	6.0

Table 3: **Real-world evaluation on the Unitree G1 robot** (success rate (%), reported as mean $\pm$ std over three independent evaluation sessions under distinct scene arrangements and lighting; each session uses 20 rollouts/task for tabletop and 10 rollouts/task for whole-body).

	Tabletop Manipulation									*Loco-Manipulation*			
	Insert Plate	Drawer Interaction	Arrange Flower	Move Spoon	Pick & Place	Box Packing	Stack Cup	Twist Cap	Avg	Shelf Org	Relocate Chair	Assembly Line Work	Avg
$\pi_{0.5}$ [29]	$41.7_{\pm 2.9}$	$18.3_{\pm 2.9}$	$18.3_{\pm 7.6}$	$18.3_{\pm 2.9}$	$25.0_{\pm 5.0}$	$10.0_{\pm 0.0}$	$16.7_{\pm 2.9}$	$18.3_{\pm 2.9}$	$20.8_{\pm 1.0}$	-	-	-	-
GR00T-N1.5 [30]	$71.7_{\pm 7.6}$	$76.7_{\pm 5.8}$	$28.3_{\pm 2.9}$	$18.3_{\pm 5.8}$	$83.3_{\pm 5.8}$	$28.3_{\pm 10.4}$	$23.3_{\pm 2.9}$	$56.7_{\pm 7.6}$	$48.3_{\pm 4.1}$	$43.3_{\pm 5.8}$	$23.3_{\pm 5.8}$	$36.7_{\pm 5.8}$	$34.4_{\pm 1.9}$
Qwen3DiT	$13.3_{\pm 2.9}$	$0.0_{\pm 0.0}$	$3.3_{\pm 2.9}$	$10.0_{\pm 0.0}$	$10.0_{\pm 5.0}$	$0.0_{\pm 0.0}$	$1.7_{\pm 2.9}$	$0.0_{\pm 0.0}$	$4.8_{\pm 0.7}$	-	-	-	-
DiT4DiT	$\mathbf{86.7}_{\pm 2.9}$	$\mathbf{83.3}_{\pm 5.8}$	$\mathbf{76.7}_{\pm 2.9}$	$\mathbf{48.3}_{\pm 7.6}$	$\mathbf{98.3}_{\pm 2.9}$	$\mathbf{53.3}_{\pm 5.8}$	$\mathbf{60.0}_{\pm 5.0}$	$\mathbf{83.3}_{\pm 5.8}$	$\mathbf{73.8}_{\pm 0.0}$	$\mathbf{73.3}_{\pm 5.8}$	$\mathbf{66.7}_{\pm 5.8}$	$\mathbf{76.7}_{\pm 5.8}$	$\mathbf{72.2}_{\pm 1.9}$

Simulation results. DiT4DiT tops both benchmarks (Tables 1, 2): **98.6%** average on LIBERO [28] (98.4/99.6/98.6/97.6% on Spatial/Object/Goal/Long) and **56.7%** on RoboCasa-GR1 [3] (56.3/56.8% on Articulated-/Novel-Object), outpacing GR00T-N1.5 and GR00T-N1.6 by 13.8 and 8.9 points.

DiT4DiT also surpasses recent video- and world-model policies. On LIBERO, it ties the strongest baseline Cosmos Policy [25] (98.6 vs. 98.5) and leads Motus [24] / mimic-video [26] by 0.9 / 3.1 points. The gap widens on RoboCasa-GR1 (+8.9 / +10.2 / +12.2 over Motus / Cosmos Policy / mimic-video) at 2.2B parameters vs. 5.9B for Motus. These baselines condition the action head on reconstructed future frames or handcrafted world-model latents; DiT4DiT instead routes intermediate denoising features directly to the Action DiT, so the gain scales with task complexity.

Against the parameter-matched Qwen3DiT, swapping its VLM backbone for our Video DiT lifts performance from 96.6% $\rightarrow$ 98.6% on LIBERO and 36.2% $\rightarrow$ 56.7% on RoboCasa-GR1 on tasks that demand long-horizon reasoning, precise spatial coordination, or complex contact.

Real-world G1 task results. Table 3 reports success rates on eleven real-world tasks (eight tabletop, three whole-body), averaged over three independent evaluation sessions per task. DiT4DiT wins every task, averaging **73.8%** on tabletop, 25.5 points over GR00T-N1.5, 53.0 over $\pi_{0.5}$, and 69.0 over Qwen3DiT. Qwen3DiT effectively collapses (at most 13.3%, with zero success on three tasks), showing how hard it is to ground static image–text priors into continuous 3D control. DiT4DiT's lead is widest on precision tasks (*Arrange Flower* 76.7 vs. 28.3, *Stack Cup* 60.0 vs. 23.3, *Move Spoon* 48.3 vs. 18.3) and long-horizon tasks (83.3/83.3/98.3% on *Drawer*, *Twist Cap*, *Pick & Place*), where fine visual structure and temporal consistency matter most.

The same trend carries over to whole-body loco-manipulation: DiT4DiT reaches **72.2%** average (73.3/66.7/76.7% on *Shelf Organization*, *Relocate Chair*, *Assembly Line Work*) against GR00T-N1.5's 34.4% (43.3/23.3/36.7%)—a margin of 30.0, 43.4, and 40.0 points respectively. Routing Video DiT features through the 7-D high-level interface to the decoupled whole-body controller thus transfers the same physics-aware advantage even once locomotion and base-height control enter the loop.

4.3 Efficiency Comparison

Table 2 reports inference speed and model size. Among VAM/WAM baselines, which run multi-step video denoising at test time, DiT4DiT (2.2B, 6.0 Hz) is 3.2$\times$ faster than mimic-video (1.9 Hz), 8.6$\times$ faster than Cosmos Policy (0.7 Hz), and 12$\times$ faster than Motus (0.5 Hz, 5.9B), yet attains the highest success rate. Compared with VLA policies, which forgo video prediction and run at

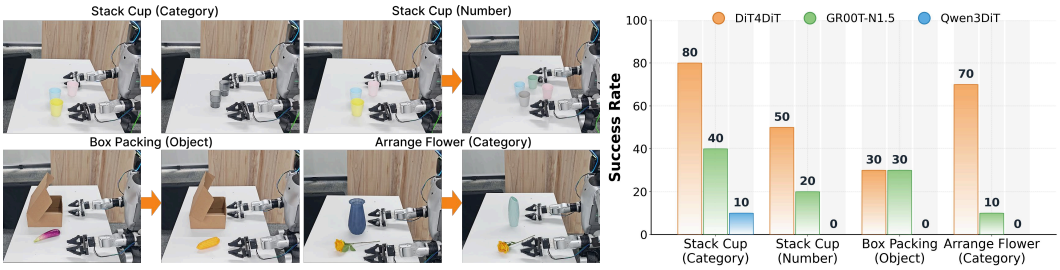

Figure 5: **Zero-shot generalization under out-of-distribution variations.** (**Left**) Qualitative rollouts across three OOD axes: *Category* (unseen cups and vases), *Object* (corn substituted for eggplant), and *Number* (four stacked cups instead of three). (**Right**) Success rates on four OOD scenarios; DiT4DiT outperforms the parameter-matched Qwen3DiT baseline and pre-trained GR00T-N1.5 [30].

9-13 Hz, DiT4DiT trades a modest speed gap for a 8.9-14.0-point gain in average success. This places DiT4DiT on a new point of the speed-accuracy frontier: video-quality priors at close to VLA-level throughput.

4.4 Do We Need a Full Video Rollout?

We claim that a policy does not need to complete the video-generation process: a short rollout that exposes the Video DiT hidden state at the noisy end of the schedule ($\tau_f \approx 1$) suffices. We test this on five representative RoboCasa-GR1 tasks (*CanToDrawerClose, FromCuttingboardToBasket, FromPlacematToBowl, FromPlateToCardboardbox, FromTrayToPot*). We sweep $N_v \in \{1, 2, 4, 8, 16, 32\}$, the number of video-denoising steps executed before the Action DiT reads the hidden state; larger N_v simultaneously means a longer rollout and a feature-extraction timestep τ_f pushed toward the clean-data end of the schedule. Success rate *peaks at* $N_v = 1$ *and decreases monotonically* with N_v (Fig. 6(a)): committing more compute to pixel-level denoising, and reading from a cleaner τ_f, both erode the physics-aware structure the action head relies on. This indicates that imagining a full video at test time is not only unnecessary but harmful. Complementary ablations on the sampling distributions of τ_v and τ_a are provided in Appendix A.2.

4.5 Generalization Capability

We introduce four challenging zero-shot generalization scenarios on the real-world Unitree G1 robot (visualized in Fig. 5 Left) to demonstrate the generalization capability. These tasks evaluate three distinct dimensions of generalization:

- **Category generalization**: In *Stack Cup (Category)* and *Arrange Flower (Category)*, we drastically alter the material, shape, and visual appearance of the interactive objects (e.g., swapping standard plastic cups for metallic/glass variants, and changing both the vase and the flower).
- **Object substitution**: In *Box Packing (Object)*, the target item is completely replaced with a novel, out-of-distribution object (e.g., swapping an eggplant for corn).
- **Quantity variation**: In *Stack Cup (Number)*, we test whether the policy can handle a different number of objects than seen during training, evaluating its resistance to distractors and novel scene clutter.

As shown in the right panel of Fig. 5, DiT4DiT shows strong zero-shot transfer in the real world. The parameter-matched Qwen3DiT baseline collapses under visual shifts, scoring 0% on three of four tasks. In contrast, DiT4DiT captures the underlying physical interactions more robustly. On *Arrange Flower (Category)*, it reaches 70% success, far above Qwen3DiT (0%) and the pre-trained GR00T-N1.5 (10%). It also maintains 50% success on *Stack Cup (Number)*, suggesting that video-generative representations are more robust to visual changes and distractors.

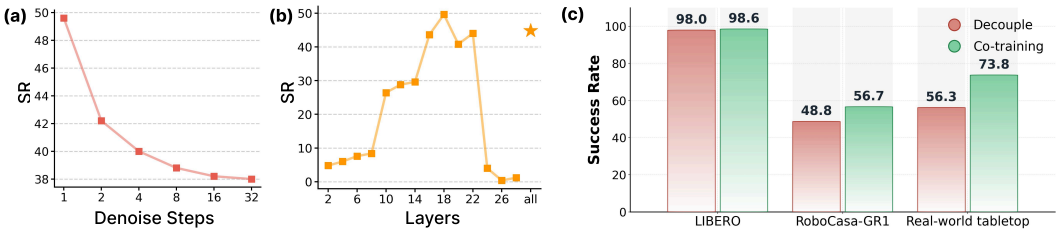

Figure 6: (a) **Denoising steps:** Impact of the number of iterative denoising steps used for action conditioning. A single forward step yields the highest success rate, preventing over-commitment to pixel-level reconstruction. (b) **Feature extraction layer:** Success rate across different hidden layers of the video backbone, with performance peaking at layer 18. (c) **Joint vs. decoupled training:** success rate when the Video DiT is jointly trained with the Action DiT (co-training) or frozen (decouple). Co-training consistently wins, with the margin growing from $+0.6$ on LIBERO to $+7.9$ on RoboCasa-GR1 and $+17.5$ on real-world tabletop tasks.

4.6 Ablations

Choice of feature extraction layer. Using the same five-task subset as Sec. 4.4, we probe hidden states from different Video DiT blocks. As shown in Fig. 6(b), early layers (2–8) underperform—likely encoding low-level textures without actionable semantics—while the final layers (24–28) collapse, having over-specialized to pixel-level reconstruction. Performance peaks at layer 18, where middle-to-deep blocks balance spatiotemporal physics with high-level scene understanding. Averaging across all layers is competitive but trails the best single layer, so we adopt layer 18.

Joint *vs.* decoupled training. To test whether the video and action branches should share gradients, we compare our co-training objective to a decoupled variant that freezes the Video DiT while training only the Action DiT on its features. Fig. 6(c) reports success rate on three settings. Co-training is at worst neutral and at best decisive: the gap is $+0.6$ on LIBERO (98.6 vs. 98.0), widens to $+7.9$ on RoboCasa-GR1 (56.7 vs. 48.8), and reaches $+17.5$ on the real-world tabletop tasks (73.8 vs. 56.3). Crucially, the advantage grows with task difficulty and embodiment complexity, where action-relevant priors differ most from pure pixel reconstruction, supporting the claim that sharing gradients across the two branches reshapes the video latent space into a more control-relevant one, rather than leaving it at its pretraining optimum.

5 Conclusion

DiT4DiT co-trains a Video DiT and an Action DiT under a dual flow-matching objective, and replaces multi-step video rollout with a single forward pass at a fixed flow timestep, turning the video model into a stateful feature extractor rather than a frame generator. The resulting policy sets new state-of-the-art averages on LIBERO (98.6%) and RoboCasa-GR1 (56.7%), attains 73.8% / 72.2% on real-world Unitree G1 tabletop and whole-body tasks, and generalizes zero-shot to unseen object categories and scene variations—at 2.2B parameters and $\sim$6 Hz closed-loop control. These results suggest that video-dynamics priors, accessed in a single pass, are a more effective scaling proxy for embodied policy learning than static image-text features. Limitations are discussed in Appendix A.12.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (No. 62306257).

References

[1] 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. 1, 2, 3, 6, 7, 14, 15, 18, 20, 21

[2] Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, et al. FLARE: Robot learning with implicit world modeling. *arXiv preprint arXiv:2505.15659*, 2025. 1, 2, 14

[3] 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. 1, 2, 3, 8, 14, 15

[4] Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as I can, not as I say: Grounding language in robotic affordances. In *Conference on Robot Learning*, pages 287–318. PMLR, 2023. 2

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 2, 3

[6] 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. 2, 3

[7] 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. 2, 3, 7, 8, 14, 17

[8] Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, et al. $\pi^*_{0.6}$: a VLA that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025. 2

[9] NVIDIA GEAR Team. GR00T N1.6: An improved open foundation model for generalist humanoid robots. NVIDIA Research, December 2025. URL https://research.nvidia.com/labs/gear/gr00t-n1_6/. 2, 3, 7, 8, 17

[10] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[11] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 2

[12] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic VLMs: Investigating the design space of visually-conditioned language models. *arXiv preprint arXiv:2402.07865*, 2024. 2

[13] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. 2, 7, 14, 17

[14] 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. 2, 3, 18

[15] NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, et al. Cosmos world foundation model platform for physical AI, 2025. URL https://arxiv.org/abs/2501.03575. 2, 3

[16] Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical AI. *arXiv preprint arXiv:2511.00062*, 2025. 2, 3, 5, 14, 19

[17] Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Xin Jin, Liangchen Li, et al. Z-Image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025. 2

[18] Unitree. UnifoLM-WMA-0: A world-model-action (WMA) framework under UnifoLM family, 2025. URL https://github.com/unitreerobotics/unifolm-world-model-action. 2

[19] 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. 2, 3

[20] Yao Feng, Hengkai Tan, Xinyi Mao, Guodong Liu, Shuhe Huang, Chendong Xiang, Hang Su, and Jun Zhu. Vidar: Embodied video diffusion model for generalist bimanual manipulation. *arXiv preprint arXiv:2507.12898*, 2025. 2, 3

[21] 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. 2, 3

[22] Yiqi Wang, Mrinal Verghese, and Jeff Schneider. Latent policy steering with embodiment-agnostic pretrained world models. *arXiv preprint arXiv:2507.13340*, 2025. 2, 3

[23] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. *arXiv preprint arXiv:2503.00200*, 2025. 2, 3

[24] 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. 2, 3, 7, 8, 17

[25] 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. 2, 3, 7, 8, 17

[26] 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. 2, 3, 7, 8, 17

[27] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 2, 5

[28] 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, 2023. 3, 6, 8, 21

[29] 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. 3, 7, 8, 17

[30] Johan Bjorck, Valts Blukis, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, et al. GR00T N1.5: An improved open foundation model for generalist humanoid robots. NVIDIA Research, June 2025. URL https://research.nvidia.com/labs/gear/gr00t-n1_5/. 3, 7, 8, 9, 17

[31] 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. 3, 7, 17

[32] Wei Li, Renshan Zhang, Rui Shao, Jie He, and Liqiang Nie. CogVLA: Cognition-aligned vision-language-action model via instruction-driven routing & sparsification. *arXiv preprint arXiv:2508.21046*, 2025. 3, 7, 17

[33] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. 3

[34] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-Sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024. 3

[35] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017. 3

[36] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018. 3

[37] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. UniSim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1389–1399, 2023. 3

[38] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in Neural Information Processing Systems*, 36:9156–9172, 2023. 3

[39] 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. 3

[40] 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. 3

[41] 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. 3

[42] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024. 3

[43] Zhide Zhong, Haodong Yan, Junfeng Li, Xiangchen Liu, Xin Gong, Wenxuan Song, Jiayi Chen, and Haoang Li. FlowVLA: Thinking in motion with a visual chain of thought. *arXiv preprint arXiv:2508.18269*, 2025. 3

[44] Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. WorldVLA: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025. 3

[45] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 4

[46] Alisson Azzolini, Junjie Bai, Hannah Brandon, Jiaxin Cao, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, et al. Cosmos-Reason1: From physical common sense to embodied reasoning. *arXiv preprint arXiv:2503.15558*, 2025. 5

[47] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025. 7, 17

[48] Jorge Aldaco, Travis Armstrong, Robert Baruch, Jeff Bingham, Sanky Chan, Kenneth Draper, Debidatta Dwibedi, Chelsea Finn, Pete Florence, Spencer Goodrich, et al. ALOHA 2: An enhanced low-cost hardware for bimanual teleoperation. *arXiv preprint arXiv:2405.02292*, 2024. 15

[49] Zhigen Zhao, Liuchuan Yu, Ke Jing, and Ning Yang. XRoboToolkit: A cross-platform framework for robot teleoperation. *arXiv preprint arXiv:2508.00097*, 2025. 15, 16

[50] Jinhui Ye, Ning Gao, Senqiao Yang, Jinliang Zheng, Zixuan Wang, Yuxin Chen, Pengguang Chen, Yilun Chen, Shu Liu, and Jiaya Jia. StarVLA-α: Reducing complexity in vision-language-action systems. *arXiv preprint arXiv:2604.11757*, 2026. 17

A Appendix

A.1 Validation of Video Generation as a Scaling Proxy

A core hypothesis of this work is that video generation is an effective proxy task for robot control. We therefore compare it against two widely used semantic-centric pre-training paradigms: (i) *object-level grounding* [1], which attaches an auxiliary detection head to a VLM so that it learns "what" and "where" objects are; and (ii) *FLARE-style latent alignment* [2], which uses learnable queries over VLM features aligned to the latents of future observations. For this action-free proxy-objective study, we adapt FLARE's future-latent alignment objective as a backbone pretraining task while omitting its action flow-matching branch, since no action annotations are used. For a fair trainable-parameter budget, the VLM baselines use Qwen3-2B [13] and our video backbone uses Cosmos-Predict2.5-2B [16].

All three settings share the same protocol: the VLM or video backbone is pre-trained on the target dataset in a self-supervised manner (grounding uses pre-annotated bounding boxes) and then frozen while the action expert is fine-tuned, which isolates the contribution of the proxy task. We evaluate on the 24 tabletop manipulation tasks of the RoboCasa-GR1 suite [3, 1]. As shown in Fig. 1, the video-generation objective reaches the same success level with up to 7$\times$ fewer training steps and 10$\times$ less data than either semantic-centric baseline, and its advantage *grows* with data scale—exactly the pattern one would expect from a proxy task that continues to benefit from more pixels rather than saturating. We take this as evidence that video generation is a viable scaling proxy for generalizable robot control.

A.2 Ablation on Tri-timestep Scheme

Our tri-timestep scheme decouples three diffusion processes: the video timestep τ_v that supervises video generation, the action timestep τ_a that drives the Action DiT, and the feature timestep τ_f used for one-shot feature extraction. The feature timestep τ_f is studied jointly with the video rollout length in the main text (Sec. 4.4): increasing N_v lengthens the rollout and shifts τ_f toward the clean-data end, and we observe a monotone drop in success rate as N_v grows. Here we complete the picture by ablating the remaining two axes—the sampling distributions of τ_v and τ_a. We use uniform sampling for video supervision and a Beta-based action timestep distribution biased toward higher noise levels, consistent with the high-noise training emphasis in π_0 [7]. Table 4 reports RoboCasa-GR1 average success rate under the four combinations.

The uniform-video/Beta-action configuration achieves the highest success rate (56.7%) among the four settings. Using Beta sampling for both branches yields 49.8%, while uniform sampling for both yields 51.3%. Reversing the assignment—Beta for video and uniform for action—yields 46.7%. These results support the asymmetric sampling configuration used in DiT4DiT on this benchmark.

Table 4: **Ablation on the tri-timestep sampling scheme.** Each row varies the sampling distribution of the video timestep τ_v and the action timestep τ_a while keeping the feature timestep τ_f fixed; we report average success rate (%) over the 24 tasks of the RoboCasa-GR1 tabletop suite. Pairing uniform τ_v with Beta-based τ_a sampling biased toward higher noise levels during training is best; using the same sampling family for both branches performs worse.

τ_v	τ_a	Success rate (%)
Beta	Uniform	46.7
Uniform	Uniform	51.3
Uniform	Beta	**56.7**
Beta	Beta	49.8

A.3 RoboCasa-GR1 Tabletop Benchmark

The RoboCasa-GR1 benchmark [1] is built on the RoboCasa simulator [3] and uses the Fourier GR1 humanoid, which has two 7-DoF arms, two 6-DoF dexterous hands, and a 3-DoF waist, for a 29-dimensional action space. The policy takes only ego-view camera observations. Each of the 24 tasks is evaluated over 50 rollouts with a maximum horizon of 720 steps; we report category-wise and overall average success rates (%).

In the simulator, we designed a targeted object-substitution experiment within the RoboCasa [3] environment. Specifically, we restricted the training distribution to three tasks involving only a single object category: *BottleToDrawerClose, BottleToCabinetClose,* and *BottleToMicrowaveClose*. During evaluation, we completely removed the bottle and tested the policies zero-shot on four unseen objects: *Can, Cup, Milk,* and *Wine*. DiT4DiT exhibits a remarkably stronger ability to generalize to novel objects compared to the parameter-matched Qwen3DiT baseline, as shown in Table 5. Across all three tasks, our approach maintains robust performance. Most notably, in the *ToDrawerClose* tasks, DiT4DiT achieves an impressive 54.5% success rate with the unseen objects, exceeding Qwen3DiT by a massive 22.5%. We observe similarly substantial margins in *ToCabinetClose* (34.0% vs. 24.5%) and *ToMicrowaveClose* (30.5% vs. 17.0%).

Table 5: **Zero-shot generalization to unseen objects in RoboCasa-GR1.** Success rates (%) when policies trained on bottle-based task variants are evaluated with four unseen objects: can, cup, milk, and wine.

Task	Qwen3DiT	DiT4DiT
ToDrawerClose	32.0	**54.5**
ToCabinetClose	24.5	**34.0**
ToMicrowaveClose	17.0	**30.5**

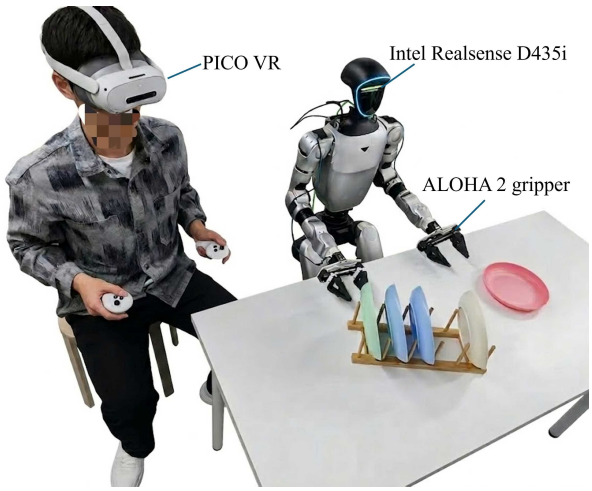

Figure 7: **Real-robot system setup.** A Unitree G1 humanoid equipped with two 7-DoF arms ending in ALOHA 2 [48] grippers and a head-mounted Intel RealSense D435i camera. Demonstrations are collected with a PICO VR headset and handheld controllers through the XRoboToolkit [49] SDK: wrist poses drive dual-arm end-effector IK for the upper body, controller triggers drive continuous ALOHA 2 gripper commands, and—for whole-body tasks—the joysticks and headset orientation feed a lower-body RL policy running concurrently with the VR-teleoperated upper body.

A.4 Teleoperation Setup

Our real-world system (Fig. 7) is a Unitree G1 humanoid with two 7-DoF arms ending in ALOHA 2 [48] grippers, a head-mounted Intel RealSense D435i providing ego-centric RGB at 640×480@30 Hz, and a single NVIDIA RTX 4090 workstation for closed-loop inference. Demonstrations are collected with a PICO VR headset and handheld controllers streamed through the

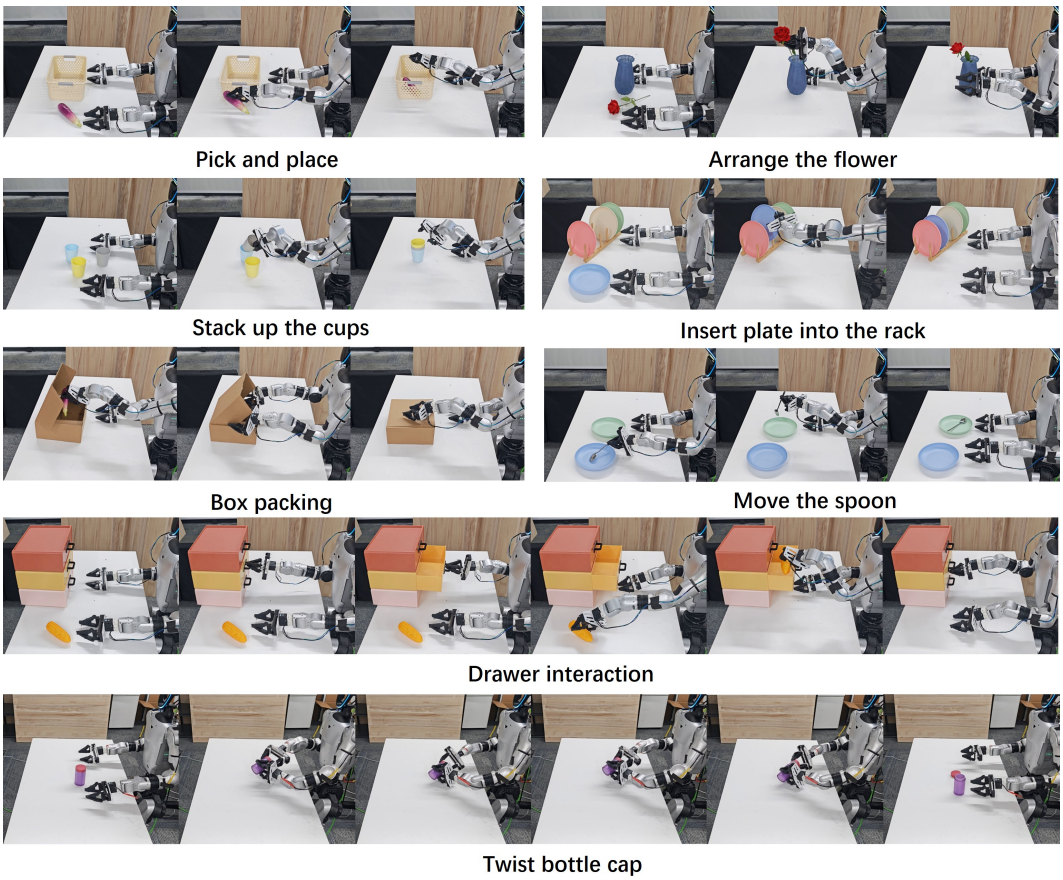

Figure 8: **Real-world evaluation suite on the Unitree G1 humanoid robot.** The selected tasks evaluate distinct dimensions of robotic proficiency, ranging from high-precision spatial manipulation (e.g., *stack up the cups, insert plate into the rack, arrange the flower, twist cap*) to complex, extended-horizon execution (e.g., *box packing, drawer interaction)*.

XRoboToolkit [49] SDK. The operator's 6-DoF wrist poses, aligned to the robot's end-effector frame via a one-time calibration, are solved into dual-arm joint targets by a Pink-based whole-body IK solver that respects all joint limits; controller triggers drive the ALOHA 2 grippers as continuous $[0, 1]$ commands. For whole-body tasks, a separate lower-body RL controller runs concurrently (see below): the left joystick commands base linear velocity and the right joystick commands yaw rate, while the headset pitch/roll/yaw—clipped to $\pm 30^\circ$—drives torso orientation. The upper-body teleop loop runs at 30 Hz and the low-level controller at 50 Hz. All streams—robot state, gripper state, RGB images, and teleop action targets—are logged with monotonic timestamps at 30 Hz into LeRobot-format parquet episodes, with a 0.2 s time-delta watchdog flagging any desynchronized frames.

Decoupled whole-body controller. To decouple VR-teleoperated manipulation from locomotion, the lower body runs a decoupled whole-body policy that consumes upper-body teleop commands in parallel with the IK solver. It executes two ONNX models—a *balance* policy for standing and a *walk* policy for locomotion—and gates between them by a commanded-velocity threshold $\|(v_x, v_y, \omega_z)\| < 0.05$ m/s (no hysteresis). Each step the policy receives an 86-dimensional observation (commanded $\{v_x, v_y, \omega_z\}$, target base height, torso RPY, IMU angular velocity and projected gravity, all 29 joint positions and velocities, and the previous 15-D action), stacked over a 6-frame history (~ 120 ms) for temporal context, and outputs a 15-DoF joint-target vector for the two legs (hip pitch/roll/yaw, knee, ankle pitch/roll) and the 3-DoF waist. The network output a is mapped to motor targets as $q_{\text{target}} = 0.25\, a + q_{\text{default}}$, where q_{default} is a nominal standing pose, and the underlying PD loop runs at 500 Hz on the robot. Commanded velocities are clipped to $v_x \in [-0.1, 0.3]$ m/s, $v_y \in [-0.1, 0.1]$ m/s, $\omega_z \in [-0.8, 0.8]$ rad/s, and are zeroed unless the commanded base height is within ± 1 cm of 0.74 m. If no teleop update arrives within 1 s, the policy reverts to a safe standing goal.

A.5 Real-world Tabletop Evaluation

To validate the real-world applicability of our approach, we deploy our policy on a Unitree G1 humanoid robot. The robotic system features a continuous 16-DoF action space, driven by two 7-DoF arms and ALOHA2 grippers, and relies exclusively on the robot's egocentric (ego-view) camera for visual observations. To comprehensively assess the model's robustness across diverse physical interactions and spatial reasoning challenges, we construct a benchmark suite comprising eight distinct household manipulation tasks. As illustrated in Fig. 8, these include: *pick and place, arrange the flower, stack up the cups, insert plate into the rack, box packaging, move the spoon*, *drawer interaction* and *twist cap*. For each task, we collected a dataset of exactly 200 human demonstration episodes. During the evaluation phase, performance is measured over 20 independent real-world rollouts per task, with the success rate reported as the primary metric.

Additional real-world robustness evaluation. We conduct a separate robustness evaluation of DiT4DiT on the Stack Cups task under three controlled perturbations: (i) increased clutter, by randomly placing 5–10 irrelevant objects around the task objects on the tabletop; (ii) changed lighting, by illuminating the tabletop with a red light; and (iii) dynamic distractors, by manually moving a hand-held irrelevant object around the periphery of the camera view. We evaluate 50 trials per condition, including the original setting as a reference. DiT4DiT achieves success rates of 64% in the original setting, 58% with increased clutter, 52% under changed lighting, and 60% with dynamic distractors. Changed lighting produces the largest observed decrease of 12 percentage points, while dynamic distractors produce the smallest decrease of 4 percentage points.

A.6 Real-world Whole-body Evaluation

We extend the real-world evaluation to whole-body loco-manipulation on the same Unitree G1 platform, now with the lower body in the control loop. Although the platform has a 31-DoF actuator space (two 7-DoF arms, two ALOHA 2 grippers, 3-DoF waist, and two 6-DoF legs), DiT4DiT only outputs a 23-dimensional action per step: the 16-dimensional upper-body target (14 arm joints plus two continuous gripper commands), and a 7-dimensional *high-level command* for the lower body consisting of torso orientation (roll, pitch, yaw), target base height, and base velocity (v_x, v_y, ω_z). This high-level command is then passed to the decoupled whole-body controller described above, which produces the 15-DoF leg and waist joint targets internally. Mirroring the training-time interface, the model runs at 50 Hz, predicting a 50-step action chunk and executing the first 32 steps before re-querying; observations consist of an ego-view RGB stream at 640×480 and proprioception (joint positions and velocities, IMU, and base pose). We evaluate three long-horizon tasks: *shelf organization* (reaching multiple shelf heights to place household items), *relocate chair* (locomoting to a chair and repositioning it to a target location), and *assembly line work* (sequential pick-and-place along a tabletop workflow). For each task we collect 200 teleoperated demonstrations and report success rate over 10 real-world rollouts.

A.7 Policy Setup and Baselines

We group the baselines into three families.

(i) VLA baselines inherit priors from large Vision-Language Models and emit actions without a generative video branch: the π series (π_0 [7] and $\pi_{0.5}$ [29]), OpenVLA-OFT [47], UniVLA [31], CogVLA [32], and the GR00T series (GR00T-N1.5 [30] and GR00T-N1.6 [9]).

(ii) VAM / WAM baselines are video-based policies that couple a video model to the action head in various ways: mimic-video [26] (frozen video backbone, single-pass hidden-state read), Motus [24] (co-trained video + action with multi-step denoising at test time), and Cosmos Policy [25] (a video diffusion model fine-tuned to emit actions as native latent frames, also with full rollout at test time).

(iii) A parameter-matched baseline, Qwen3DiT, replaces our bidirectional Video DiT backbone with a Qwen3-VL [13] 2B VLM while keeping the same Action DiT; this isolates the contribution of the video-dynamics backbone from the size and architecture of the action head.

Matched-scale baseline verification. StarVLA-α [50] uses a Qwen3-VL-4B backbone in its main RoboCasa-GR1 evaluation. For a matched-scale comparison with Qwen3DiT, we additionally evaluate its Qwen3-VL-2B configuration under our target-only RoboCasa-GR1 training protocol,

without external action-labeled pretraining data. This configuration achieves approximately 37% average success, close to the 36.2% obtained by Qwen3DiT in Table 2.

For simulation experiments (LIBERO and RoboCasa-GR1), DiT4DiT and Qwen3DiT are both trained *from scratch* on the target dataset to give a controlled, pre-training-free comparison; all other baselines are evaluated from their official open-sourced weights. For the real-world experiments, we adopt a two-stage pipeline: DiT4DiT is first pre-trained on a 241,450-episode subset of the simulated GR1 dataset [1] to acquire spatiotemporal priors, and then fine-tuned on the teleoperated G1 demonstrations. Qwen3DiT follows the exact same pipeline for a strictly matched comparison, while GR00T-N1.5 is initialized from its official checkpoint—benefiting from a substantially larger pre-training corpus—before being fine-tuned on the same real-world tasks; our pre-training volume is only $\sim$**15%** of the scale that GR00T-N1.5 trained on.

A.8 Algorithms

We summarize the full training procedure—including the tri-timestep sampling, hook-based feature extraction, and the joint video–action flow-matching objective—in Algorithm 1, and the closed-loop inference pipeline, which replaces the video rollout with a single forward pass at τ_f, in Algorithm 2.

Algorithm 1 Joint Training of Video and Action DiT

Require: Conditioning frames $\mathbf{O}_{\text{cond}}$, future frames $\mathbf{O}_{\text{future}}$, action $\mathbf{a}_0$, state s, language goal l, action mask M

Ensure: Updated parameters θ (Video DiT), ϕ (Action DiT)

// ===== Video DiT Forward =====
$\mathbf{z}_t^0 \leftarrow \text{VAE}_{\text{enc}}(\mathbf{O}_{\text{cond}})$ ▷ Encode conditioning frames
$\mathbf{z}_{t+1}^0 \leftarrow \text{VAE}_{\text{enc}}(\mathbf{O}_{\text{future}})$ ▷ Encode future frames
$\tau_v \sim \mathcal{U}[0,1]$ ▷ Sample video timestep
$z \sim \mathcal{N}(0, I)$ ▷ Sample video noise
$\mathbf{z}_{t+1}^{\tau_v} \leftarrow (1-\tau_v)\cdot \mathbf{z}_{t+1}^0 + \tau_v \cdot z$ ▷ Noisy future latent
$\hat{v}_{\text{video}} \leftarrow v_\theta^{\text{video}}(\mathbf{z}_{t+1}^{\tau_v}, \tau_v \mid \mathbf{z}_t^0, l)$ ▷ Predict velocity
$v_{\text{video}}^* \leftarrow z - \mathbf{z}_{t+1}^0$ ▷ Target velocity
$\mathcal{L}_{\text{video}} \leftarrow \|\hat{v}_{\text{video}} - v_{\text{video}}^*\|^2$ ▷ Video loss
// ===== Extract Hidden States at a Fixed Timestep =====
$\tau_f \leftarrow 1$ ▷ Fixed, first step of the flow schedule
$\hat{\mathbf{z}}_{t+1} \sim \mathcal{N}(0, I)$ ▷ Pure-noise future latent
$\mathbf{h}_t^{\tau_f} \leftarrow \mathcal{H}(\theta, \hat{\mathbf{z}}_{t+1}, \tau_f, \mathbf{z}_t^0, l)$ ▷ Single forward pass, capture block-k hidden
// ===== Action DiT Forward =====
$\sigma \sim \text{Beta}(1.5, 1.0)$; $\tau_a \leftarrow \sigma / c_a$, $c_a = 0.999$ ▷ Sample training timestep
$\epsilon \sim \mathcal{N}(0, I)$ ▷ Sample action noise
$\mathbf{a}_t^{\tau_a} \leftarrow (1-\tau_a)\cdot \mathbf{a}_t^0 + \tau_a \cdot \epsilon$ ▷ Noisy action
$\hat{v}_{\text{action}} \leftarrow v_\phi^{\text{action}}(\mathbf{a}_t^{\tau_a}, \tau_a \mid \mathbf{h}_t^{\tau_f}, s)$ ▷ Predict velocity
$v_{\text{action}}^* \leftarrow \epsilon - \mathbf{a}_t^0$ ▷ Target velocity
$\mathcal{L}_{\text{action}} \leftarrow \|(\hat{v}_{\text{action}} - v_{\text{action}}^*) \odot M\|^2 / \|M\|_1$ ▷ Masked action loss
// ===== Backward =====
$\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{action}} + \lambda \cdot \mathcal{L}_{\text{video}}$
Update θ, ϕ via $\nabla \mathcal{L}_{\text{total}}$

A.9 Model & Training Configurations

See Table 6.

A.10 Backbone Scaling

We additionally evaluate DiT4DiT with Wan2.2-TI2V-5B [14] as the video backbone while keeping the Action DiT unchanged. For the Wan2.2 variant, we average features extracted from all transformer layers without tuning the feature-extraction layer. On the 24-task RoboCasa-GR1 benchmark, this variant achieves a 60.2% average success rate, compared with 56.7% for Cosmos-Predict2.5-2B,

Table 6: **DiT4DiT model and training configurations.** Parameters without an embodiment prefix are shared across all three settings; action-space parameters differ by robot (LIBERO: 7-DoF Franka with gripper; RoboCasa-GR1: 29-DoF humanoid; real-world G1: 16-DoF upper body plus 7-D high-level lower-body command, see Appendix A.6).

Parameter	**Value**
Video DiT	
Base VGM	Cosmos-Predict2.5-2B [16]
Attention implementation	flash_attention_2
Hidden feature dim.	2048
Extract layer k	18
Conditioning-frame timestep	1×10^{-4}
Future-frame timestep τ_f	first step of flow schedule (≈ 1)
Future inference steps	1
Action DiT	
Variant	DiT-B, 16 layers
Hidden size / output dim.	2560
Cross-attention dim.	2048
Max sequence length	1024
Positional embedding	learned, interleaved self-attention
Norm type	AdaLN
Dropout / final dropout	0.2 / applied
Past action window	0
Num. visual tokens fed to Action DiT	32
Action timestep $\tau_a = \sigma / c_a$, $\sigma \sim \mathrm{Beta}(\alpha, \beta)$	α=1.5, β=1.0, c_a=0.999
Timestep buckets	1000
Inference flow steps	4
Repeated diffusion steps (train)	4
Per-embodiment Action dimensions	
LIBERO: $\lvert a \rvert$ / $\lvert s \rvert$ / future-action window	8 / 16 / 7 (horizon 8)
RoboCasa-GR1: $\lvert a \rvert$ / $\lvert s \rvert$ / future-action window	32 / 64 / 15 (horizon 16)
Real-world G1: $\lvert a \rvert$ / $\lvert s \rvert$ / future-action window	32 / 64 / 15 (horizon 16)
Training	
Optimizer	AdamW, (β_1, β_2)=$(0.9, 0.95)$, ϵ=10^{-8}
Weight decay	1×10^{-8}
Gradient clip / accum. steps	1.0 / 1
Video DiT LR	1×10^{-5}
Action DiT LR	1×10^{-4}
LR schedule	cosine with min LR, $\eta_{\min}$=5×10^{-7}
Max train steps	1×10^{5}
Warmup steps	5×10^{3}
Per-device batch size	16
Number of GPUs	32

Algorithm 2 DiT4DiT Inference. The Action DiT only requires the hidden state $\mathbf{h}_t^{\tau_f}$ from a *single* forward pass of the Video DiT at $\tau_f = 1$ (lines after the optional block). The optional video-sampling block is used only for future-frame visualization and is **not** in the control loop.

Require: Conditioning frames $\mathbf{O}_{\text{cond}}$, state s, language goal l, action sampling steps N_a
Require: (*optional, visualization only*) Video sampling steps N_v
Ensure: Predicted action $\hat{\mathbf{a}}_t$ (and, optionally, predicted future frames $\hat{\mathbf{O}}_{\text{future}}$)

$\mathbf{z}_t^0 \leftarrow \text{VAE}_{\text{enc}}(\mathbf{O}_{\text{cond}})$ ▷ Encode conditioning frames
// ===== (Optional) Video DiT Rollout for Visualization =====
$\hat{\mathbf{z}}_{t+1}^{\text{vis}} \sim \mathcal{N}(0, I)$ ▷ Initialize from noise
$\Delta\tau_v \leftarrow 1/N_v$
for $i = 0, 1, \ldots, N_v - 1$ **do**
  $\tau_v \leftarrow 1 - i \cdot \Delta\tau_v$
  $\hat{v} \leftarrow v_\theta^{\text{video}}(\hat{\mathbf{z}}_{t+1}^{\text{vis}}, \tau_v \mid \mathbf{z}_t^0, l)$
  $\hat{\mathbf{z}}_{t+1}^{\text{vis}} \leftarrow \hat{\mathbf{z}}_{t+1}^{\text{vis}} - \Delta\tau_v \cdot \hat{v}$ ▷ Euler step backward
end for
$\hat{\mathbf{O}}_{\text{future}} \leftarrow \text{VAE}_{\text{dec}}(\hat{\mathbf{z}}_{t+1}^{\text{vis}})$ ▷ Decoded future frames, for visualization only
// ===== Feature Extraction at $\tau_f = 1$ (one forward pass) =====
$\tau_f \leftarrow 1$ ▷ Fixed, pure-noise end of the flow
$\hat{\mathbf{z}}_{t+1} \sim \mathcal{N}(0, I)$ ▷ Pure-noise future latent
$\mathbf{h}_t^{\tau_f} \leftarrow \mathcal{H}(\theta, \hat{\mathbf{z}}_{t+1}, \tau_f, \mathbf{z}_t^0, l)$ ▷ Single forward pass; capture block-k hidden
// ===== Action DiT Sampling =====
$\hat{\mathbf{a}}_t \sim \mathcal{N}(0, I)$ ▷ Initialize from noise
$\Delta\tau_a \leftarrow 1/N_a$
for $i = 0, 1, \ldots, N_a - 1$ **do**
  $\tau_a \leftarrow 1 - i \cdot \Delta\tau_a$
  $\hat{v} \leftarrow v_\phi^{\text{action}}(\hat{\mathbf{a}}_t, \tau_a \mid \mathbf{h}_t^{\tau_f}, s)$
  $\hat{\mathbf{a}}_t \leftarrow \hat{\mathbf{a}}_t - \Delta\tau_a \cdot \hat{v}$ ▷ Euler step backward
end for
return $\hat{\mathbf{a}}_t$, $\hat{\mathbf{O}}_{\text{future}}$ ▷ $\hat{\mathbf{O}}_{\text{future}}$ optional

an improvement of 3.5 percentage points (Table 7). These results demonstrate that DiT4DiT can accommodate a different, larger video backbone.

Table 7: **Backbone scaling on RoboCasa-GR1.** The Action DiT is unchanged across the two configurations.

Video backbone	Feature extraction	Success rate (%)
Cosmos-Predict2.5-2B	Single intermediate layer	56.7
Wan2.2-TI2V-5B	Average over all layers	60.2

A.11 Dataset Configuration

We use five datasets, split across simulation and real-world pipelines (Table 8).

Simulation (train from scratch). For RoboCasa-GR1, we train on the Fourier_GR1_Unified_1K dataset [1], 24,000 demonstrations collected on the 29-DoF Fourier GR1 humanoid. For LIBERO, we use the official dataset of 2,000 demonstrations on the 7-DoF Franka Emika Panda arm (500 demonstrations per suite $\times$ 4 suites). Training from scratch on the target dataset provides a controlled comparison, since the official pre-training data used by some baselines is not publicly released.

Real-world pre-training. For real-robot deployment, DiT4DiT is first pre-trained on the Fourier_GR1_Pretrain_10K dataset [1] (241,450 episodes, same 29-DoF GR1 embodiment), which acquires spatiotemporal priors at scale. This corpus is only $\sim$**15%** of the pre-training volume used by GR00T-N1.5, supporting the data-efficiency claim in the main text.

Real-world fine-tuning on the Unitree G1. We collect two teleoperated datasets on a Unitree G1 using the setup of Fig. 7: a *tabletop* set covering the eight tasks of Appendix A.5 with 200

demonstrations each (1,600 in total, 16-DoF action), and a *whole-body* set covering the three loco-manipulation tasks of Appendix A.6 with 200 demonstrations each (600 in total, 23-D policy output: 16 upper-body joint targets plus 7 high-level lower-body commands). Both sets share the pre-training checkpoint and are fine-tuned independently per configuration.

Table 8: **Datasets used in DiT4DiT.** #Episodes are teleoperated demonstration counts. Action dimension is the policy output dimension per step; for the G1 whole-body configuration this is 16 upper-body joint targets plus a 7-D high-level lower-body command (torso RPY, base height, v_x, v_y, ω_z) consumed by the decoupled controller (Fig. 7).

Stage	Dataset	#Episodes	Embodiment	Action dim.
Sim. train	LIBERO [28]	2,000	Franka Emika Panda	7
Sim. train	Fourier_GR1_Unified_1K [1]	24,000	Fourier GR1 humanoid	29
Real pre-train	Fourier_GR1_Pretrain_10K [1]	241,450	Fourier GR1 humanoid	29
Real fine-tune (tabletop)	G1 Tabletop (ours, 8 tasks $\times$ 200)	1,600	Unitree G1	16
Real fine-tune (whole-body)	G1 WBC (ours, 3 tasks $\times$ 200)	600	Unitree G1	23

A.12 Limitations and Discussion

While DiT4DiT demonstrates exceptional capabilities in bridging generative video priors and continuous robotic control, limitations present exciting avenues for future research.

Our current physical deployment relies exclusively on a single egocentric (ego-view) camera. While this highlights the remarkable spatial reasoning capabilities of our video representations, single-view setups are fundamentally susceptible to severe visual occlusions. In complex bimanual tasks, the robot's own arms or larger objects can temporarily block the camera's line of sight, potentially disrupting the temporal continuity of the visual features. Future iterations could explore the integration of auxiliary sensory inputs, such as wrist-mounted cameras or tactile feedback, fusing these modalities with the Video DiT backbone to maintain robust state estimation during severe occlusions.

Our real-world experiments achieved state-of-the-art zero-shot generalization using a pre-training corpus that represents merely 15% of the data volume utilized by contemporary large-scale models like GR00T. A natural and promising next step is to drastically scale the pre-training data across diverse robotic embodiments (e.g., varying kinematics, grippers, and camera parameters). Given the data-efficient nature of our dual flow-matching objective, scaling DiT4DiT with massive, cross-embodiment datasets could yield a highly generalized robotic foundation model, further solidifying video generation as the optimal scaling proxy for embodied intelligence.

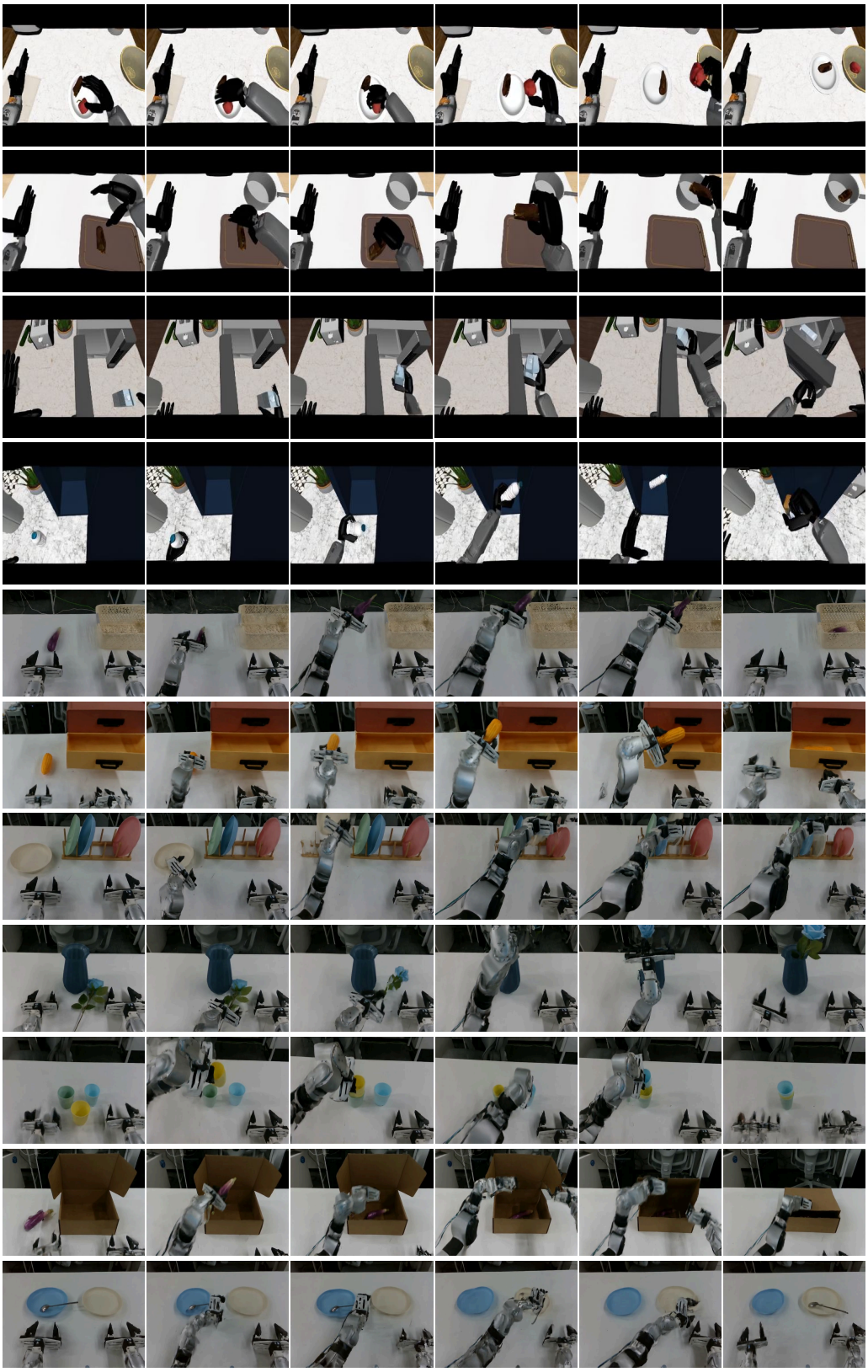

Figure 9: Future video rollouts generated by DiT4DiT.

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 state three concrete contributions—a jointly trained Video–Action Model, single-pass inference via a fixed feature timestep, and state-of-the-art results on LIBERO, RoboCasa-GR1, and real Unitree G1 tabletop/whole-body tasks—each of which is substantiated in the method (Sec. 3) and experiments (Sec. 4).

   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: Limitations are discussed in the conclusion and appendix: the Video DiT backbone adds inference cost ($\sim$6 Hz closed-loop rate), experiments focus on tabletop and whole-body loco-manipulation rather than contact-rich or dexterous tasks, and real-world evaluation is limited to a single humanoid platform (Unitree G1).

   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: The paper is empirical; it does not claim formal theoretical results requiring proofs.

   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: The method section fully specifies the dual-DiT architecture, the dual flow-matching objective, and the tri-timestep scheme; the experiments and appendix describe benchmarks (LIBERO, RoboCasa-GR1), real-robot setup (Unitree G1 with ALOHA 2 grippers, Intel RealSense D435i), teleoperation pipeline, decoupled whole-body controller, dataset sizes, and training/evaluation protocols in enough detail to reproduce the main results.

   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: [Yes]

   Justification: Anonymized code is provided as supplementary material at submission time, with instructions for reproducing the main LIBERO and RoboCasa-GR1 results. Public assets we build on (Cosmos-Predict2.5-2B, Qwen3-VL, LIBERO, RoboCasa) are referenced in the paper.

   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: The experimental setup section and the appendix report benchmarks, dataset sizes, evaluation rollouts per task, model and training configurations, and baseline specifications in tabular form.

   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: Real-world success rates (Table 3) are reported as mean $\pm$ std over three independent evaluation sessions per task, each consisting of 20 rollouts/task (tabletop) or 10 rollouts/task (whole-body) under distinct scene arrangements and lighting; per-row averages are the mean $\pm$ std of the three session-level averages. Simulation averages (LIBERO, RoboCasa-GR1) are reported without explicit error bars, following standard practice in comparable VLA and VAM papers, because per-task rollout counts (50) and the number of tasks (24 for RoboCasa-GR1, 40 tasks across 4 suites with 10 tasks per suite for LIBERO) yield stable aggregate estimates.

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 appendix specifies training compute (GPU type and count, training steps) and real-world inference hardware (a single NVIDIA RTX 4090 workstation driving the Unitree G1 at $\sim$6 Hz closed-loop).

   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 conforms to the NeurIPS Code of Ethics. No human-subjects data is collected beyond the authors' own teleoperation demonstrations, and public assets are used under their respective licenses.

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: [Yes]

    Justification: Positive impacts include faster, more data-efficient generalist robot policies for household and assistive manipulation. Potential negative impacts include labor displacement and misuse of autonomous humanoid robots; our contribution—a more sample-efficient policy architecture—does not introduce risks beyond those of existing VLA/VAM systems, and real-world deployment remains gated by hardware safety envelopes and human supervision.

    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: The released artifact is a robot-control policy trained on teleoperation demonstrations; it does not pose the kinds of misuse risks associated with unconstrained language models, image generators, or scraped web datasets.

    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 external assets we build on—Cosmos-Predict2.5-2B, Qwen3-VL, LIBERO, RoboCasa, the Unitree G1 platform, ALOHA 2 gripper design, and the XRoboToolkit teleoperation SDK—are cited at first use and used in accordance with their respective licenses.

    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: [N/A]

    Justification: The paper does not release new datasets or pretrained models at submission time beyond the supplementary code.

    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: The paper does not involve crowdsourcing or research with human subjects; teleoperation demonstrations were collected by the authors themselves.

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: The paper does not involve research with human subjects.

    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: LLMs are not a component of the core method. The Qwen3-VL baseline used for controlled comparison is a Vision-Language Model whose role is cited and clearly described; its use does not constitute an original or non-standard LLM component of our method.

    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.