UniT: Toward a Unified Physical Language for Human-to-Humanoid Policy Learning and World Modeling

Boyu Chen[1,2,*] **Yi Chen**[1,3,*] **Lu Qiu**[3] **Jerry Bai**[1] **Yuying Ge**[1,†] **Yixiao Ge**[1]
[1]XPENG Robotics [2]Tsinghua University
[3]The University of Hong Kong

Abstract

Scaling humanoid foundation models is bottlenecked by the scarcity of robotic data. While massive egocentric human data offers a scalable alternative, bridging the cross-embodiment chasm remains a fundamental challenge due to kinematic mismatches. We introduce **UniT** (**Uni**fied Latent Action **T**okenizer via Visual Anchoring), a framework that learns a unified physical language for human-to-humanoid manipulation transfer. Grounded in the philosophy that heterogeneous kinematics share consistent visual consequences, UniT employs a tri-branch cross-reconstruction mechanism: actions predict vision to anchor kinematics to physical outcomes, while vision reconstructs actions to filter out irrelevant visual confounders. Concurrently, a fusion branch integrates these purified modalities into a shared discrete latent space of cross-embodiment physical intents. We validate UniT across two paradigms: (1) **Policy Learning (VLA-UniT):** By predicting these unified tokens, VLA-UniT achieves state-of-the-art performance with high data efficiency on the RoboCasa GR1 benchmark. Leveraging diverse human data further improves out-of-distribution (OOD) generalization in simulation and real-world deployment, and enables *zero-shot task transfer* in the real world. (2) **World Modeling (WM-UniT):** By aligning cross-embodiment dynamics via unified tokens as conditions, it supports direct human-to-humanoid action-conditioned generation, translating human knowledge into enhanced action controllability for humanoid video generation. Ultimately, by inducing a more aligned cross-embodiment representation (empirically supported by t-SNE visualizations revealing improved alignment of human and humanoid features), UniT offers a scalable path to distill human priors into humanoid manipulation capabilities.

1 Introduction

Scaling foundation models for humanoids is fundamentally bottlenecked by scarce high-quality robotic data. Massive human motion sequences from low-cost capture offer a scalable alternative rich in physical priors, but leveraging them requires bridging a major cross-embodiment gap[1] caused by mismatched degrees of freedom (DoFs) and control paradigms. Traditional pipelines rely on **motion retargeting**[2, 3], utilizing complex, labor-intensive kinematic solvers that are unscalable and often physically inconsistent. Consequently, there is a critical need for a data-driven **unified physical language** that projects heterogeneous data into a shared latent action space.

Despite recent progress in unified representations, critical limitations persist (Fig. 2). **Action-only** methods [4–8] lack external grounding, causing severe human-robot distribution shifts. Conversely, **vision-only** latent action frameworks [9–12] infer intent directly from videos to bypass kinematic mismatches; however, they often entangle low-level appearance confounders like texture, degrading

*Equal contribution.
†Corresponding author: yyge13@gmail.com.

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

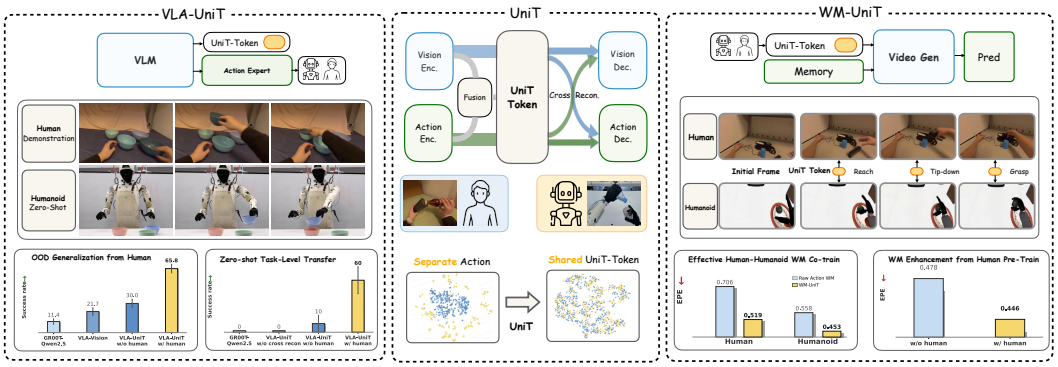


Figure 1: **Overview of UniT.** A visually anchored tokenizer learns a unified physical language by mapping human and humanoid trajectories into shared discrete latent actions through three encoder branches and cross-reconstruction (center). The same representation supports policy learning by predicting latent actions from vision and language (left), and world modeling by conditioning video generation on latent actions (right). Human demonstrations improve policy generalization and enable zero-shot task transfer; human action conditions also transfer to humanoid video generation.

fine-grained execution and underutilizing human pose priors. Furthermore, paradigms incorporating **both vision and action** [13, 14] typically use independent tokenizers, yielding disjoint vocabularies rather than a deeply unified control medium. Empirically, most existing latent-action systems remain confined to simple manipulator setups, leaving scalability to dexterous humanoids largely unexplored.

To address these challenges, we introduce **UniT** (**Uni**fied Latent Action **T**okenizer via Visual Anchoring), as illustrated in Figure 3. Our key insight is that while kinematics differ across embodiments, the visual outcomes of physical intents remain consistent. Thus, visual observations can serve as a universal anchor to align disparate kinematic spaces. UniT distills physical intent by concurrently extracting three coupled features: *temporal-visual* (from consecutive frames), *kinematic* (from inter-frame actions), and *fused visuo-motor*. Instead of treating these as isolated streams, we enforce a rigorous **cross-reconstruction objective**, compelling each feature to independently decode both visual transitions and low-level actions.

This mechanism operationalizes **visual anchoring** via a cross-modal information bottleneck. Forcing kinematics to reconstruct visual changes anchors heterogeneous actions to their actual physical consequences, reducing the memorization of embodiment-specific noise. Conversely, forcing visual features to reconstruct kinematics strips away irrelevant appearance confounders. This bidirectional constraint discards uncorrelated noise from both domains, preserving the physical intent. Consequently, UniT generates deeply integrated visuo-motor tokens that form a unified latent action space. This alignment enhances robustness even with missing modalities during deployment, laying the groundwork for a **unified physical language** in cross-embodiment transfer.

We evaluate this unified language across two embodied-AI paradigms:

First in **Policy Learning (VLA-UniT):** We integrate UniT into Vision-Language-Action architectures [5, 15–20]. VLA-UniT adds UniT token prediction as a structured training objective, while a lightweight flow head generates embodiment-specific actions for execution. Evaluated on the RoboCasa GR1 benchmark and a real-world humanoid, VLA-UniT outperforms state-of-the-art baselines and improves data efficiency. By further leveraging diverse human data, it also achieves stronger out-of-distribution (OOD) generalization and zero-shot task transfer from human with emergent upper-body coordination including waist rotation on unseen tasks.

Second in **World Modeling (WM-UniT):** We utilize UniT as universal conditions for action-conditioned video generation [21–24] instead of raw actions. Having absorbed physical priors during joint dynamics training, these tokens enhance prediction consistency. Rollout validations suggest that pre-training on large-scale human data aligns fine-grained cross-embodiment actions, enabling the transfer of physical dynamics and improving downstream humanoid action controllability.

Beyond downstream performance, UniT provides intrinsic structural benefits. It naturally encourages VLA policies and world models to align their internal cross-embodiment context representations. Furthermore, these visually-anchored tokens maintain higher reconstruction accuracy under noisy action inputs compared to action-only alternatives, demonstrating enhanced robustness to input perturbations.

Our main contributions are summarized as follows:

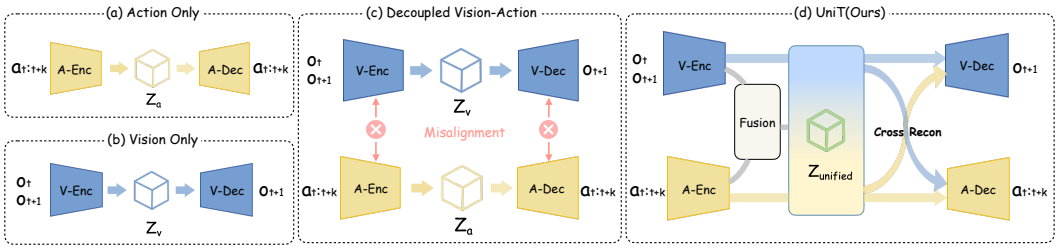


Figure 2: **Comparison of latent action architectures.** UniT differs from action-only, vision-only, and decoupled vision-action designs by learning a shared vision-action space through **cross-reconstruction**.

- **The Unified Tokenizer (UniT):** We propose a visually-anchored tri-branch tokenizer with cross-reconstruction that maps heterogeneous actions into a shared discrete latent space, enabling robust cross-embodiment alignment and action denoising.
- **Generalizable Policy Execution (VLA-UniT):** We integrate UniT into VLA architectures, achieving high data efficiency on the RoboCasa GR1 benchmark. Leveraging human data further improves OOD generalization and enables real-world zero-shot task transfer from human.
- **Unified World Modeling (WM-UniT):** We employ UniT as universal conditions for world models, demonstrating improved action controllability and cross-embodiment transfer in human-to-humanoid action-conditioned generation.

We will release all code, model weights, and benchmark environments upon acceptance to facilitate reproducibility and future research.

2 Related Work

Learning from Human Data. Human data offer a scalable alternative to costly robot demonstrations. Prior work pre-trains visual representations from egocentric videos [25–28], but often underuses fine-grained hand or wrist motion. Other methods co-train policies on human and robot demonstrations [2, 29–32], yet mixed-embodiment training can induce embodiment-specific shortcuts when action distributions differ [1, 33]. Retargeting approaches such as EgoVLA [3] and In-n-On [1] remain case-specific and can misalign with visual observations. UniT instead learns a shared latent space that absorbs embodiment differences without explicit retargeting or action-space unification.

Latent Action Representations. We categorize latent action methods by input modality. **Action-only** methods [4–8] tokenize raw action trajectories, but without visual grounding remain tied to embodiment-specific kinematics. **Vision-only** methods [9–12, 34] infer latent actions from observations, as in Moto [9], LAPA [10], UniVLA [11], and Villa-X [34]; this bypasses action-space mismatch but can entangle appearance with motion and miss fine-grained execution details. Concurrent works such as METIS [13] and XR-1 [14] take **both vision and action as encoder inputs** but have not achieved explicit vision-action alignment. UniT jointly encodes vision and action with a shared codebook and bidirectional cross-reconstruction.

Policy learning and world modeling represent two core paradigms for embodied AI. For policy learning, VLA models [5, 15–20] generate actions for closed-loop control. For world modeling, action-conditioned video generation [21–24] simulates robotic dynamics. Across both paradigms, the heterogeneity of action spaces remains a key bottleneck, as systems typically rely on embodiment-specific raw actions. UniT addresses this by providing a unified token interface that projects heterogeneous actions into a shared latent space. This space serves as a prediction target for VLAs and a conditioning signal for world models, enabling scalable human-to-humanoid transfer across both paradigms.

3 Methodology

3.1 Overview

Our goal is to learn a visually anchored shared latent representation—termed a "physical language"—for cross-embodiment transfer from demonstrations (o_t, s_t, a_t). We first present **UniT** (Sec. 3.2), a visual-anchored tri-branch tokenizer that maps visual transitions, actions, and fused visuo-motor features into discrete tokens of cross-embodiment physical intent. We then deploy these tokens in two paradigms: **VLA-UniT** (Sec. 3.3) predicts UniT tokens as structured policy targets, while **WM-UniT** (Sec. 3.4) uses UniT features as universal conditions for future-observation prediction.

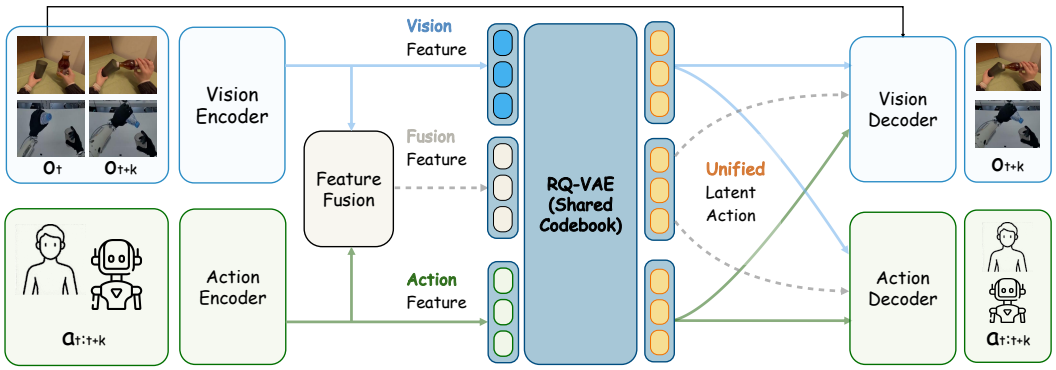


Figure 3: **Architecture of UniT.** Heterogeneous cross-embodiment vision pairs (o_t, o_{t+k}) and action chunks $a_{t:t+k}$ are encoded into vision, action, and fused features via tri-branch encoders. A **shared RQ-VAE codebook** quantizes all three branches into a unified discrete space, yielding cross-embodiment **Unified Latent Action** tokens. Both vision and action decoders reconstruct from these shared tokens, enforcing **cross-reconstruction** alignment.

3.2 UniT: Unified Latent Action Tokenizer via Vision Anchoring

Given an observation transition (o_t, o_{t+k}), current state s_t, and action chunk $a_{t:t+k}$, UniT learns a discrete latent action representation that maps heterogeneous human and humanoid behaviors into a shared token space. UniT is built upon a tri-branch architecture that jointly models visual transitions, actions, and fused visuo-motor features under cross-reconstruction (Fig. 3).

Tri-Branch Encoding. UniT uses three transformer-encoder branches with learnable queries. The **visual branch** E_v follows UniVLA [11], encoding frozen DINOv2 features of (o_t, o_{t+k}) as a domain-stable physical-transition representation. The **action branch** E_a encodes s_t and $a_{t:t+k}$ after embodiment-specific padding and projection. The **fusion branch** E_m combines visual and action features into a compact visuo-motor latent for robust tokenization; details are in Appendix A.

Shared Discrete Quantization. Continuous latents from all three branches are quantized by a shared RQ-VAE [35] codebook, which approximates each latent through a sum of codebook entries that progressively minimize the quantization error. This maps visual, action, and fused features across embodiments into the shared discrete vocabulary of physical intent.

Cross-Reconstruction. The core mechanism of UniT is cross-reconstruction: each quantized token is decoded into both future visual features and action chunks (Appendix A). This bidirectional constraint realizes visual anchoring: visually similar outcomes can be shared even when action spaces differ. Through the shared visual decoder, tokens with similar physical effects are pulled toward a common codebook space, anchoring heterogeneous action representations into a shared latent space.

Training Objective. UniT optimizes visual reconstruction, action reconstruction, and RQ-VAE commitment losses across the visual, action, and fusion branches. VLA-UniT uses fusion-branch tokens, while WM-UniT uses action-branch features; the full objective is given in Appendix A.

3.3 VLA-UniT: Cross-Embodiment Policy Learning via UniT

Given the current observation o_t, state s_t, and instruction ℓ, VLA-UniT predicts a future action chunk $a_{t:t+H}$. Built upon GR00T n1.5 [16] with its backbone replaced by Qwen2.5-VL [36] (Fig. 4a), our model introduces UniT token prediction to facilitate cross-embodiment learning.

Specifically, learnable queries are appended to the VLM sequence to predict discrete UniT codes tokenized from ground-truth transitions. By sharing a physical-intent vocabulary across embodiments, this objective aligns vision-language features into a shared distribution. The flow-matching head then generates embodiment-specific actions conditioned on the vision-language features. Detailed objectives are provided in Appendix A. Future frames and ground-truth actions only provide token supervision during training. At inference, both token prediction and continuous action generation use the current observation, state, and instruction.

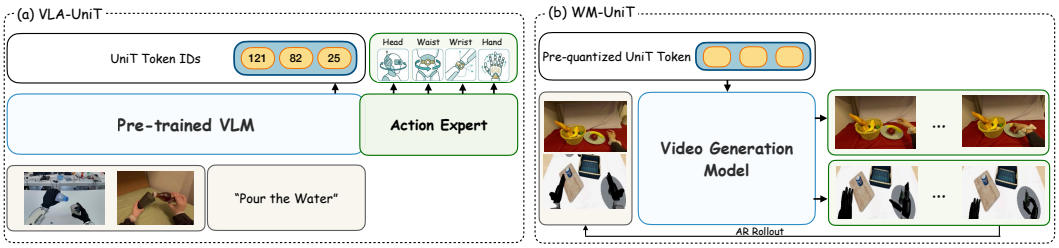


Figure 4: **Downstream applications of UniT. (a)** VLA-UniT predicts UniT tokens as structured policy targets; a flow head generates actions from vision-language features. **(b)** WM-UniT uses UniT features as universal action conditions for controllable video generation.

3.4 WM-UniT: Cross-Embodiment World Modeling via UniT

Given the current observation o_t and an action condition derived from $(s_t, a_{t:t+k})$, WM-UniT generates future visual observations. Built upon Cosmos Predict 2.5 [24] (Fig. 4b), it replaces embodiment-specific raw actions with UniT action features $\tilde{z}_t^a$ to serve as a unified conditioning interface.

Specifically, the UniT action encoder extracts continuous pre-quantization features from the current state and action chunk. Inheriting visual-dynamics priors established during tokenization, these cross-embodiment features are projected through an MLP and injected via cross-attention. By routing through this unified interface, the model facilitates the transfer of visual generation capabilities from human data to humanoid prediction. The generation model is trained with flow matching and supports autoregressive rollout by feeding generated frames back as observations (Appendix A).

4 Experimental Setups

4.1 Benchmarks and Datasets

RoboCasa GR1 Tabletop Simulation. We evaluate on the RoboCasa benchmark with the GR1 humanoid robot [37], covering tabletop pick-and-place and articulated manipulation tasks. *Data configurations:* We use **Full Data** for benchmark comparisons with external baselines, and **Few-Shot** (10% robot demonstrations) to study data efficiency and the benefit of human data under robot-data scarcity. *Learning from human:* To examine human-to-humanoid transfer for both policy learning and world modeling, we incorporate the EgoDex `basic_pick_place` subset [38].[3] The human data is combined with robot data for co-training, followed by fine-tuning exclusively on robot data. *Generalization:* We construct three RoboCasa generalization suites covering unseen appearance, unseen object-container combinations, and unseen object types. Dataset sizes and further details are provided in Appendix D.

Real-World Experiments We validate on a high-DoF real-world humanoid with an action space covering arms, hands, waist, and head, evaluating in-domain tasks: Pick & Place and Pouring. For real-world generalization, we evaluate five OOD axes: geometry, distractors, target placement, background, and instruction-based object disambiguation, covering human-introduced variations absent from robot teleoperation. We evaluate the policy's ability to generalize to these complementary conditions from EgoDex (Appendix D.2), with success rates and standard-error bars reported in Fig. 9.

Embodiment-specific projections map zero-padded action vectors into the shared representation without assuming joint correspondences. Simulation and real-world policies use separate checkpoints containing their corresponding target-humanoid data (Appendix D).

4.2 Evaluated Variants

Policy variants and baselines. In addition to **VLA-UniT**, we use **GR00T-Qwen2.5** as the same-architecture baseline without UniT token prediction, which adopts the GR00T n1.5 [16] architecture but replaces the original backbone with Qwen2.5-VL [36] to match our VLA-UniT setup. We also evaluate tokenizer ablations aligned with Fig. 2: **VLA-Action**, **VLA-Vision**, **VLA-UniT w/o Cross-Recon**, and **VLA-Villa**, a Villa-X-style visual-to-action reconstruction baseline [34]. Other representative diffusion, flow-matching, and action-token baselines are summarized in Appendix E.

[3] EgoDex is used for research purposes only.

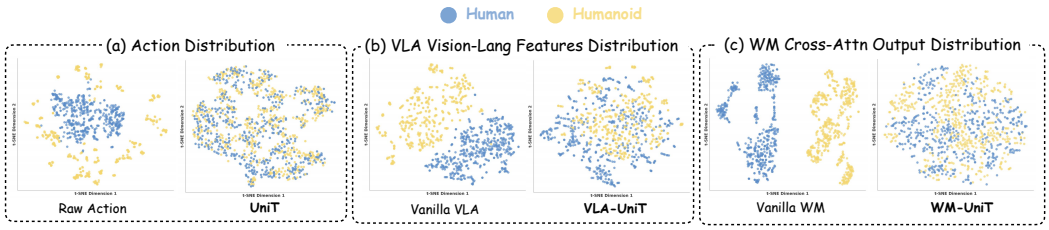


Figure 5: **Cross-embodiment alignment.** Raw actions form distinguishable clusters, whereas UniT overlaps in a unified latent action space, propagating to downstream VLA and WM features.

World modeling variants and metrics. We compare three conditioning paradigms, all built on the same Cosmos Predict 2.5 [24] backbone: **Raw Action**, which directly conditions on embodiment-specific raw actions; **WM-Action**, which replaces raw actions with features from an action-only latent tokenizer; and **WM-UniT** (Sec. 3.4), which uses UniT action-branch features as a unified cross-embodiment conditioning interface. We evaluate generation quality with PSNR, SSIM, LPIPS [39], FVD [40], and EPE, where EPE measures action controllability via optical flow.

5 Experiments

We validate UniT through the following questions:

- **Q1 (Unified Representation):** Does UniT establish a unified physical language that aligns heterogeneous embodiments and remains robust to noise? (Sec. 5.1)
- **Q2 (Efficient Policy Learning):** Does UniT improve data efficiency and enable generalization from human data for humanoid policy learning?(Sec. 5.2.1, 5.2.2, 5.2.3)
- **Q3 (Effective World Modeling):** Does UniT conditioning improve action-conditioned generation and enable cross-embodiment dynamics transfer? (Sec. 5.3.1, 5.3.2)
- **Q4 (Design Soundness):** How does UniT's visual-anchored design compare to alternative tokenizer paradigms? (Sec. 5.4)

5.1 Unified Representation: Alignment and Robustness

We examine whether UniT learns the *unified physical language* claimed in Sec. 3.2 and whether this alignment propagates into downstream models. We perform t-SNE [41] on human and humanoid samples from the RoboCasa GR1 and EgoDex co-training mixture (Fig. 5), comparing UniT-based models with raw-action baselines trained on the same mixture; Sec. 5.3.2 further tests this alignment functionally through cross-embodiment conditioning. Details and quantitative results are in Appendix C.2.

Token and downstream representation alignment. Fig. 5 first shows that raw human and humanoid actions are separated, while UniT token embeddings form shared cross-embodiment space. We then test whether this alignment propagates into downstream model internals by extracting VLA last-layer vision-language hidden states and WM last-layer cross-attention outputs. Vanilla baselines keep human and humanoid features separated, while VLA-UniT and WM-UniT produce more unified distributions, indicating that token-level alignment encourages shared downstream representations. Cross-reconstruction also raises human–humanoid codebook usage-frequency correlation from 0.001 to 0.757 (Table 6), quantitatively supporting distributional alignment.

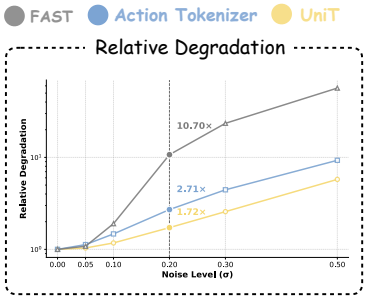


Figure 6: **Action robustness analysis.**

Robustness to action noise. At normalized noise $\sigma/\mathrm{std}(a_{\mathrm{global}}) = 0.2$, reconstruction error rises $1.7\times$ for UniT, versus $2.7\times$ for action-only tokenization and $10.7\times$ for FAST, relative to clean-input errors (Fig. 6). This supports visual anchoring under noisy actions; details are in Appendix C.1.

5.2 Policy Learning

5.2.1 Benchmark Performance and Data Efficiency

As established in Sec. 3.3, UniT token prediction provides the VLM with a compact, visually-anchored learning objective that encodes physical intent. We first compare full-data policy performance on RoboCasa GR1 using the same robot training split and evaluation protocol as the baselines in Sec. 4.2, then study data efficiency under reduced robot data.

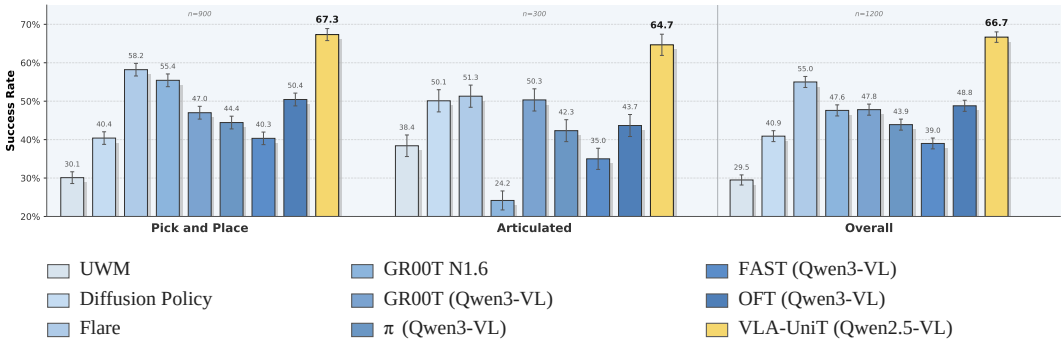


Figure 7: **Full-data RoboCasa GR1 performance.** Baselines: FLARE [42] and StarVLA [43].

Overall performance. As shown in Fig. 7, VLA-UniT achieves the best full-data RoboCasa performance, outperforming the previous best baseline by 11.7%. In the same-backbone comparison, VLA-UniT reaches 66.7% versus 36.8% for GR00T-Qwen2.5 (Fig. 8a) using same backbone, showing that UniT token prediction provides a structured physical-intent target for policy learning.

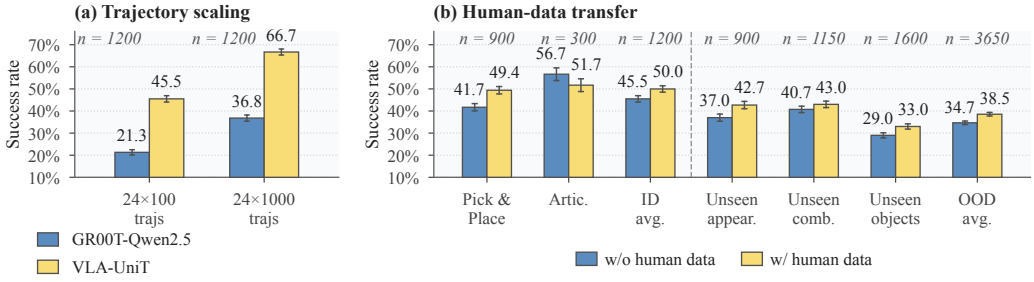


Figure 8: **RoboCasa GR1 data efficiency.** (a) Trajectory scaling. (b) Human co-training.

Data efficiency. Under the few-shot setting (Fig. 8a), VLA-UniT with only 10% of the data outperforms the full-data GR00T-Qwen2.5 baseline. This indicates that UniT tokens help the VLM learn task-relevant physical intent more efficiently than direct raw-action regression.

5.2.2 Human-to-Humanoid Transfer

We investigate whether UniT's shared latent space enables leveraging human demonstrations to improve humanoid policy learning. Under the few-shot regime in simulation, we co-train VLA-UniT on robot data and EgoDex `basic_pick_place` subset, then fine-tune on robot data alone.

Incorporating EgoDex human data improves both in-distribution and OOD policy performance (Fig. 8b). Under the few-shot setting, human co-training brings a 9.8% relative improvement in in-distribution success and an 11.0% relative improvement on OOD scenarios. These gains support the human-to-humanoid transfer enabled by UniT's shared latent space: human demonstrations are mapped into the same token vocabulary, helping the VLM build more aligned cross-embodiment representations (Sec. 5.1) and generalize better beyond the limited robot data.

5.2.3 Real-World Generalization

We deploy VLA-UniT on the real-world humanoid to validate whether the simulation gains carry over to physical execution, with human data from EgoDex `basic_pick_place`, `pour`, and `stack_bowl`.

Real-world performance. We evaluate in-domain tasks and five OOD axes, spanning background, distractors, geometry, target placement, and instruction-based combinational generalization, where

human demonstrations introduce complementary variations beyond robot data (Appendix. D.2). Real-world baselines use the same EgoDex co-training protocol, unless marked "w/o human data,". As shown in Fig. 9, VLA-UniT outperforms GR00T-Qwen2.5 and VLA-Vision, while removing human data reduces both in-domain execution and OOD robustness. The performance gains are clearest on Pouring and OOD cases, where UniT transfers bimanual coordination, visual variations, geometric changes, placement-target changes, and instruction following to humanoid execution. By contrast, VLA-Vision remains less robust, suggesting that vision-only latent actions may entangle appearance with motion and miss fine-grained execution details.

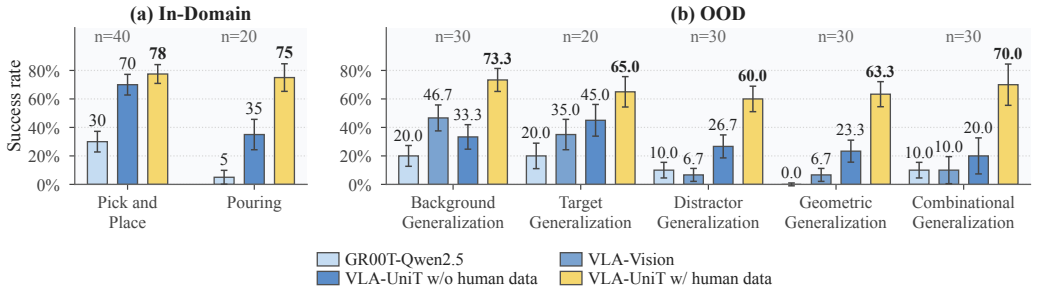


Figure 9: **Real-world humanoid evaluation. (Left)** In-domain tasks. **(Right)** OOD generalization.

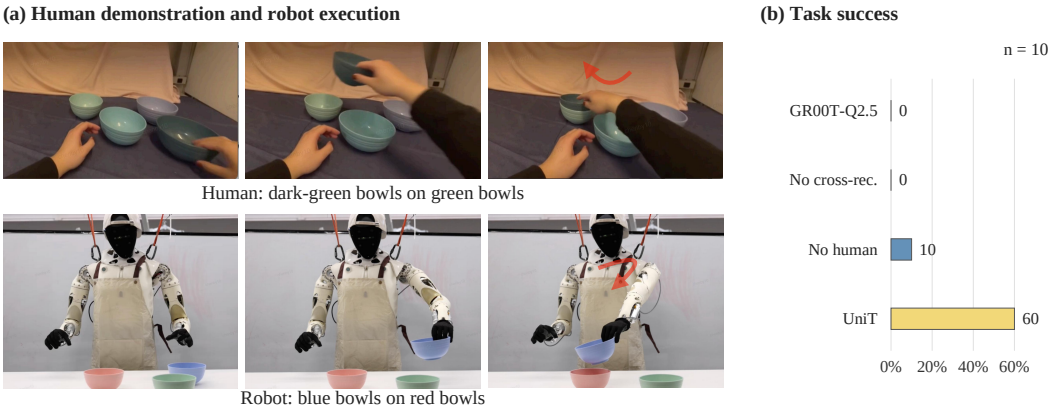


Figure 10: **Zero-shot task transfer.** Human data enables task transfer on unseen bowl stacking.

Zero-shot task-level transfer from human. We evaluate zero-shot transfer on instruction-guided bowl stacking. While robot data covers only placing bowls into a box, this task places bowls outside the original workspace and requires color-matched stacking, demanding torso coordination learned from EgoDex. Baselines and VLA-UniT without human data largely fail (below 10% success), whereas human co-training achieves 60%. Crucially, we observe that the robot exhibits active waist rotation to simultaneously compensate for the human-like visual shift and extend its operational reach. This suggests that UniT's visual-anchored cross-reconstruction transfers not only task semantics but also fine-grained, emergent coordination across embodiments.

5.3 World Modeling

5.3.1 Controllable Generation

Table 1: **Controllable generation** under EgoDex + RoboCasa-GR1 co-training.

Dataset	Method	PSNR ↑	SSIM ↑	LPIPS ↓	FVD ↓	EPE ↓
EgoDex	Raw Action	24.84	0.800	0.164	171.37	0.706
	WM-UniT	**28.06**	**0.858**	**0.086**	**130.87**	**0.519**
RoboCasa-GR1	Raw Action	13.45	0.590	0.259	237.13	0.558
	WM-UniT	**17.66**	**0.718**	**0.142**	**166.50**	**0.453**

As described in Sec. 3.4, WM-UniT conditions video generation on UniT's continuous pre-quantization features. Under EgoDex + RoboCasa-GR1 co-training, Table 1 compares this interface

with raw actions using the same video generation backbone and 10-step autoregressive rollouts. WM-UniT improves generation quality and reduces EPE on both human and humanoid subsets, showing that UniT provides a more controllable conditioning signal under human-humanoid co-training. Single-embodiment DROID results in Appendix F.1 show the same trend on real-world robot data.

5.3.2 Human-Humanoid Transfer

Table 2: **World-model pretraining and cross-embodiment conditioning.** (Left) EgoDex pre-training for RoboCasa-GR1. (Right) Human-to-robot MLLM scores.

Pretraining setting	PSNR ↑	SSIM ↑	LPIPS ↓	FVD ↓	EPE ↓
No PT	16.34	0.678	0.168	180.51	0.478
EgoDex PT	**18.06**	**0.713**	**0.135**	**153.31**	**0.446**

World model	Semantic ↑	Temporal ↑	Geometry ↑	Overall ↑
Raw Action	2.98	3.16	2.72	2.95
WM-UniT	**3.28**	**3.43**	**3.09**	**3.27**

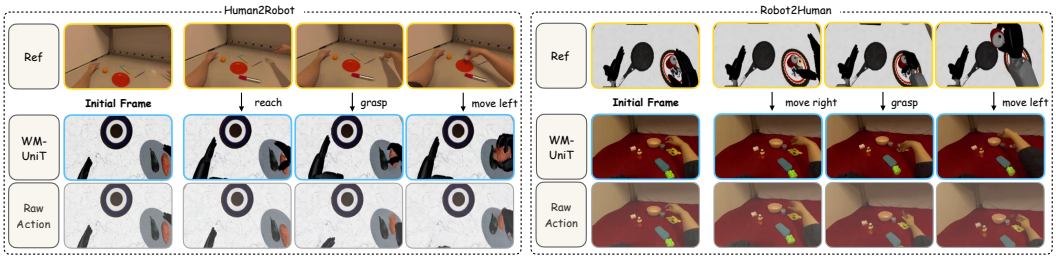


Figure 11: **Cross-embodiment conditioning. Left**: human-to-robot. **Right**: robot-to-human.

Human pre-training. We first test whether human dynamics can improve humanoid world modeling by pre-training WM-UniT on EgoDex `basic_pick_place` and fine-tuning on RoboCasa-GR1 pick-and-place subset. The gains in Table 2, together with the aligned WM representations in Fig. 5(c), show that the humanoid world model can inherit dynamics knowledge from human data through UniT.

Cross-embodiment conditioning. We further test whether UniT action conditions from one embodiment can drive generation in another. We co-train WM-UniT and a Raw Action WM baseline on human and humanoid data, then pair source-domain actions with a target-domain initial frame; WM-UniT conditions on the UniT-encoded action features, while the baseline conditions on raw actions. As shown in Fig. 11, in both human-to-robot (left) and robot-to-human (right) cases, WM-UniT rollouts remain consistent with the source action, while the Raw Action WM fails to. Table 2 quantifies the human-to-robot direction with an MLLM judge; additional examples and protocol details are in Appendix F.2 and Appendix F.3. Together, these results show that UniT transfers both dynamics knowledge and fine-grained action intent across embodiments in world modeling.

5.4 Tokenizer Design Ablation

We ablate UniT's two core claims, vision-action synergy and bidirectional cross-reconstruction (Sec. 3.2), in the VLA setting under full-data human-humanoid co-training and robot fine-tuning. All tokenizer ablations use the same policy architecture and training protocol; details are in Appendix E.

Vision-Action Synergy Enables Transfer. As shown in Fig. 12a, VLA-UniT (49.9%) outperforms VLA-Vision (45.2%) and VLA-Action (42.1%) across OOD scenarios. This suggests that UniT benefits from visual effects and precise motion details.

Cross-Reconstruction Produces Aligned Representations. VLA-UniT w/o Cross-Recon (30.3%) falls below even the single-modality variants, showing that multi-modal input alone does not guarantee alignment. The 19.6% gain from cross-reconstruction suggests that explicit cross-modal alignment is the key enabler for human-to-humanoid transfer.

Bidirectional vs. Unidirectional Reconstruction. On in-domain performance (Fig. 12b), VLA-UniT (66.8%) outperforms VLA-Villa (63.8%), which uses unidirectional visual-to-action reconstruction. Bidirectional cross-reconstruction better aligns cross-embodiment tokens by grounding motor representations in visual effects while preserving fine-grained control.

Direct Action Decoding Preserves the Gain. UniT achieves 66.5% with direct decoding and 66.7% with the flow head, versus FAST's 39.0% and 56.9%, respectively (Fig. 12c). Its gain persists across action interfaces, supporting token-level motor detail.

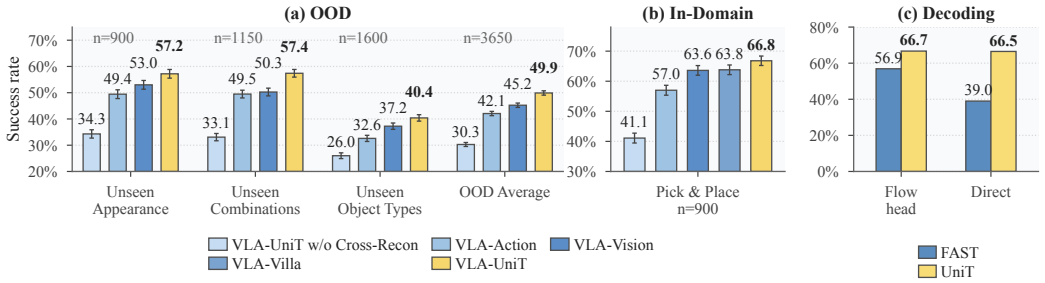


Figure 12: **Tokenizer ablations on RoboCasa GR1.** (a) OOD generalization. (b) In-domain Pick & Place, with full-data human co-training. (c) Direct versus flow-head decoding.

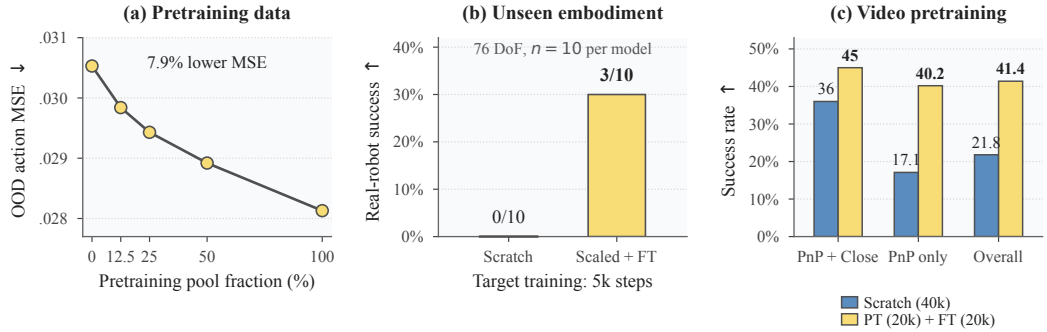


Figure 13: **Additional transfer studies.** (a) Pretraining-pool size. (b) Adaptation to a pretraining-unseen 76-DoF humanoid ($n = 10$ per model). (c) Action-unlabeled human-video pretraining.

5.5 Scaling Across Data, Embodiments, and Supervision

Cross-Embodiment Data. With target-humanoid anchors, model, and budget fixed, scaling the human/hand-robot/gripper-robot pool to 4.8k hours across over 12 datasets/configurations lowers OOD MSE from 0.03053 to 0.02813 (7.9%) and raises real-robot success from 50.9% to 65.9% (Fig. 13a).

New Embodiment. On a pretraining-unseen 76-DoF humanoid (32 body, 22 per hand), the task requires bending to grasp an object from a table, turning, and placing it on a side shelf from non-fixed initial configurations. With identical target data and 5k-step fine-tuning, scaled initialization lowers in-domain/whole-body-generalization FK-space MSE by 6.3%/4.8%. Success rises from 0/10 to 3/10: preliminary adaptation evidence (Fig. 13b).

Unlabeled Videos. After paired vision–action anchoring, three hours of self-collected, action-unlabeled tabletop video support 20k pretraining steps with a frozen action head plus 20k RoboCasa-GR1 100-shot fine-tuning steps. Against 40k steps from scratch, success rises from 21.8% to 41.4% (Fig. 13c; 50 episodes/task).

6 Conclusion and Discussion

We presented UniT, a visual-anchored latent action tokenizer that learns a unified physical language for human-to-humanoid transfer through cross-reconstruction. Across policy learning and world modeling, UniT improves data efficiency, zero-shot cross-embodiment transfer, and cross-embodiment dynamics modeling, with ablations suggesting the importance of vision-action synergy and bidirectional cross-reconstruction. Our current experiments focus on humanoid tabletop manipulation. Looking forward, scaling UniT to internet videos and diverse human demonstrations may provide broader physical priors for whole-body and dexterous control.

References

[1] Xiongyi Cai, Ri-Zhao Qiu, Geng Chen, Lai Wei, Isabella Liu, Tianshu Huang, Xuxin Cheng, and Xiaolong Wang. In-n-on: Scaling egocentric manipulation with in-the-wild and on-task

data. *arXiv preprint arXiv:2511.15704*, 2025.

[2] Tony Tao, Mohan Kumar Srirama, Jason Jingzhou Liu, Kenneth Shaw, and Deepak Pathak. Dexwild: Dexterous human interactions for in-the-wild robot policies. *arXiv preprint arXiv:2505.07813*, 2025.

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

[4] Seungjae Lee, Yibin Wang, Haritheja Etukuru, H Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions. *arXiv preprint arXiv:2403.03181*, 2024.

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

[6] An Dinh Vuong, Minh Nhat Vu, Dong An, and Ian Reid. Action tokenizer matters in in-context imitation learning. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13490–13496. IEEE, 2025.

[7] Atharva Mete, Haotian Xue, Albert Wilcox, Yongxin Chen, and Animesh Garg. Quest: Self-supervised skill abstractions for learning continuous control, 2024. *URL https://arxiv.org/abs/2407.15840*, 2024.

[8] Yating Wang, Haoyi Zhu, Mingyu Liu, Jiange Yang, Hao-Shu Fang, and Tong He. Vq-vla: Improving vision-language-action models via scaling vector-quantized action tokenizers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11089–11099, 2025.

[9] Yi Chen, Yuying Ge, Weiliang Tang, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for learning robot manipulation from videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19752–19763, 2025.

[10] 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.

[11] 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.

[12] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Xindong He, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025.

[13] Yankai Fu, Ning Chen, Junkai Zhao, Shaozhe Shan, Guocai Yao, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. Metis: Multi-source egocentric training for integrated dexterous vision-language-action model. *arXiv preprint arXiv:2511.17366*, 2025.

[14] Shichao Fan, Kun Wu, Zhengping Che, Xinhua Wang, Di Wu, Fei Liao, Ning Liu, Yixue Zhang, Zhen Zhao, Zhiyuan Xu, et al. Xr-1: Towards versatile vision-language-action models via learning unified vision-motion representations. *arXiv preprint arXiv:2511.02776*, 2025.

[15] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[16] NVIDIA, Johan Bjorck, Nikita Cherniadev Fernando Castañeda, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. In *ArXiv Preprint*, March 2025.

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

[18] 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.

[19] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[20] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[21] Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9834–9844, 2025.

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

[23] Julian Quevedo, Percy Liang, and Sherry Yang. Evaluating robot policies in a world model. *arXiv e-prints*, pages arXiv–2506, 2025.

[24] NVIDIA, Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, Prithvijit Chattopadhyay, Mike Chen, Yongxin Chen, Yu Chen, Shuai Cheng, Yin Cui, Jenna Diamond, Yifan Ding, Jiaojiao Fan, Linxi Fan, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Ruiyuan Gao, Yunhao Ge, Jinwei Gu, Aryaman Gupta, Siddharth Gururani, Imad El Hanafi, Ali Hassani, Zekun Hao, Jacob Huffman, Joel Jang, Pooya Jannaty, Jan Kautz, Grace Lam, Xuan Li, Zhaoshuo Li, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Yen-Chen Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Yifan Lu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Seungjun Nah, Yashraj Narang, Abhijeet Panaskar, Lindsey Pavao, Trung Pham, Morteza Ramezanali, Fitsum Reda, Scott Reed, Xuanchi Ren, Haonan Shao, Yue Shen, Stella Shi, Shuran Song, Bartosz Stefaniak, Shangkun Sun, Shitao Tang, Sameena Tasmeen, Lyne Tchapmi, Wei-Cheng Tseng, Jibin Varghese, Andrew Z. Wang, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Jiashu Xu, Dinghao Yang, Xiaodong Yang, Haotian Ye, Seonghyeon Ye, Xiaohui Zeng, Jing Zhang, Qinsheng Zhang, Kaiwen Zheng, Andrew Zhu, and Yuke Zhu. World simulation with video foundation models for physical ai, 2025.

[25] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[26] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

[27] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2023.

[28] Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. *Advances in neural information processing systems*, 37:124420–124450, 2024.

[29] Ri-Zhao Qiu, Shiqi Yang, Xuxin Cheng, Chaitanya Chawla, Jialong Li, Tairan He, Ge Yan, David J Yoon, Ryan Hoque, Lars Paulsen, et al. Humanoid policy˜ human policy. *arXiv preprint arXiv:2503.13441*, 2025.

[30] Chengbo Yuan, Rui Zhou, Mengzhen Liu, Yingdong Hu, Shengjie Wang, Li Yi, Chuan Wen, Shanghang Zhang, and Yang Gao. Motiontrans: Human vr data enable motion-level learning for robotic manipulation policies. *arXiv preprint arXiv:2509.17759*, 2025.

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

[32] Hongzhe Bi, Lingxuan Wu, Tianwei Lin, Hengkai Tan, Zhizhong Su, Hang Su, and Jun Zhu. H-rdt: Human manipulation enhanced bimanual robotic manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 18135–18143, 2026.

[33] Simar Kareer, Karl Pertsch, James Darpinian, Judy Hoffman, Danfei Xu, Sergey Levine, Chelsea Finn, and Suraj Nair. Emergence of human to robot transfer in vision-language-action models. *arXiv preprint arXiv:2512.22414*, 2025.

[34] Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, et al. Villa-x: enhancing latent action modeling in vision-language-action models. *arXiv preprint arXiv:2507.23682*, 2025.

[35] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11523–11532, 2022.

[36] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[37] NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025.

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

[39] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[40] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. 2019.

[41] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[42] 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.

[43] starVLA Contributors. Starvla: A lego-like codebase for vision-language-action model developing. GitHub repository, 1 2025.

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

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

[46] NVIDIA GEAR Team, Allison Azzolini, Johan Bjorck, Valts Blukis, et al. Gr00t n1.6: An improved open foundation model for generalist humanoid robots. `https://research.nvidia.com/labs/gear/gr00t-n1_6/`, December 2025.

[47] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos, 2024.

[48] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset. *RSS*, 2024.

A Implementation and Objective Details

Tri-branch encoder details. UniT uses three parallel transformer-encoder branches with learnable queries to summarize modality-specific inputs before quantization:

- **Visual branch** E_v: operates as an inverse dynamics model (IDM), taking frozen DINOv2 [44] features of (o_t, o_{t+k}) as input, following UniVLA [11]. By encoding both the current and future observations, it extracts a compact representation of the visual transition, with DINOv2 providing a stable visual anchor across embodiments.
- **Action branch** E_a: encodes current state s_t and action chunk $a_{t:t+k}$. Because embodiments differ in control mode and degrees of freedom, raw actions are zero-padded to a fixed maximum dimension and projected by embodiment-specific MLPs before being summarized. This simple padding strategy avoids manual cross-embodiment action alignment and accommodates heterogeneous action formats, making it easy to scale to new embodiments.
- **Fusion branch** E_m: combines visual and action features into a fused visuo-motor representation that captures complementary cross-modal structure for compact and robust tokens.

Fusion operator and exact downstream interfaces. The fusion branch uses a Q-Former over vision and action feature tokens to produce eight fused tokens. VLA-UniT predicts only the fusion-branch codes: eight positions with two residual-level indices per position; its flow head separately predicts embodiment-specific continuous actions. WM-UniT instead uses continuous, pre-quantization action-branch features as its conditioning signal.

Action formats and temporal preprocessing. Padding standardizes tensor shape and does not impose a correspondence between physical functions of action dimensions. Trajectories are interpolated to a common 30 Hz sampling rate before fixed-length chunking, without motion-specific temporal warping or event-level alignment. An embodiment with an already covered action format can reuse the existing codebook. For an unseen action format, an embodiment-specific action projection is added and fine-tuned while the shared codebook remains unchanged; action outputs use embodiment-specific decoders.

Missing action annotations. Cross-reconstruction requires paired vision–action samples. For video-only samples, visual reconstruction and quantization can be retained while unavailable action/fusion losses are masked. After paired anchoring, the trained visual branch can tokenize additional videos without action estimates. The label-free-video policy experiment uses this pretrained visual branch, rather than learning the full tokenizer from videos alone.

Cross-reconstruction decoder details. For each quantized branch token $\hat{z}_i$, UniT decodes both future visual features and action chunks:

$$\hat{f}^{(i)}_{t+k} = D_v(\hat{z}_i,\, f_t), \qquad \hat{a}^{(i)}_{t:t+k} = D_a(\hat{z}_i,\, s_t). \tag{1}$$

The shared visual decoder D_v operates as a forward dynamics model conditioned on the current DINOv2 feature f_t and is supervised by cosine similarity against the future feature f_{t+k}. The embodiment-specific action decoder D_a is conditioned on the current state s_t and reconstructs the action chunk $a_{t:t+k}$. Both decoders model relative physical change rather than absolute configurations, which helps different embodiments share analogous transition patterns despite different state and action spaces. The shared visual decoder is the key visual anchor: tokens that produce similar physical outcomes are encouraged to occupy nearby regions of the shared codebook.

Branch usage. VLA-UniT uses fusion-branch tokens because policy learning observes paired visual transitions and actions during training, so the fused visuo-motor representation provides the compact target for physical intent prediction. WM-UniT instead uses action-branch features as conditions because future frames must be generated by the world model and are unavailable at inference time; using action-branch features avoids leaking target visual information while still carrying the visual-dynamics priors learned through cross-reconstruction. Thus, UniT plays different downstream roles: a supervised target for VLA training, and an inference-time conditioning interface for WM generation.

UniT training objective. The total tokenizer loss aggregates visual reconstruction, action reconstruction, and quantization terms across all three branches:

$$\mathcal{L} = \sum_{i\in\{v,a,m\}} \left[\lambda_v \mathcal{L}^{(i)}_{\text{vis}} + \lambda_a \mathcal{L}^{(i)}_{\text{act}}\right] + \mathcal{L}_{\text{RQ}}, \tag{2}$$

where $\mathcal{L}_{\text{vis}}$ is cosine visual-feature reconstruction, $\mathcal{L}_{\text{act}}$ is action reconstruction, and $\mathcal{L}_{\text{RQ}}$ is the RQ-VAE commitment loss.

Residual quantization. Continuous latents from the visual, action, and fusion branches are quantized with a shared residual codebook $\mathcal{C}$:

$$\hat{z}_i = \text{RQ}(z_i; \mathcal{C}), \quad i \in \{v, a, m\}. \tag{3}$$

The residual levels refine the token representation from coarse physical intent to finer motion detail. Table 3 summarizes the key architectural and quantization hyperparameters of UniT.

VLA-UniT token prediction. Learnable action queries q_t are appended to the VLM sequence. Conditioned on the current observation o_t and language instruction ℓ, the VLM predicts logits over UniT code indices. The target codes are obtained by encoding the ground-truth observation pair, state, and action chunk through the pre-trained UniT tokenizer:

$$c_t = \text{UniT}(o_t, o_{t+k}, a_{t:t+k}, s_t), \qquad \hat{p}_t = f_{\text{VLM}}(o_t, \ell, q_t), \qquad \mathcal{L}_{\text{token}} = \text{CE}(\hat{p}_t, c_t). \tag{4}$$

VLA-UniT flow matching. Continuous embodiment-specific actions are generated by a lightweight flow-matching head. Given action chunk $A_t = [a_t, \ldots, a_{t+H-1}]$, time variable $\tau \sim \mathcal{U}[0, 1]$, and noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the interpolated path is $A^\tau_t = \tau A_t + (1-\tau)\epsilon$. We set the policy horizon to match the tokenizer chunk size, i.e., $H = k$; $k = 16$ in simulation and $k = 50$ in real-world experiments to account for the slower pace of teleoperated demonstrations. The flow head learns a velocity field conditioned on the VLM features x_t, the current visual encoding $\text{Enc}(o_t)$, and τ:

$$\mathcal{L}_{\text{fm}} = \mathbb{E}_{\tau,\epsilon}\left[\left\|V_\theta(A^\tau_t \mid x_t,\, \text{Enc}(o_t),\, \tau) - (A_t - \epsilon)\right\|^2_2\right], \tag{5}$$

Table 3: **UniT tokenizer hyperparameters.**

Component	Hyperparameter	Value
RQ-VAE	Num. tokens per sample	8
	Residual levels	2
	Codebook size (per level)	128
	Codebook dimension	32
	Commitment loss β (level 1 / level 2)	0.5 / 0.1
Tri-Branch Encoders	Hidden dimension	1024
	Attention heads	16
	Transformer layers	4
Action Input	Action chunk size k	16 (sim.) / 50 (real)
	Max action / state dimension	128
Loss Weights	Visual / action reconstruction λ_v, λ_a	1.0 / 1.0
	RQ-VAE commitment	1.0
Training	Training steps	80k
	Batch size	256
	Learning rate	1e-4
	LR schedule	Cosine decay, warmup 5%

Table 4: **VLA-UniT architectural hyperparameters.**

Component	Hyperparameter	Value
VLM Backbone	Architecture	Qwen2.5-VL-3B
	Hidden dimension	2048
Action Expert (Flow Head)	Transformer layers	16
	Attention heads	32
	Hidden dimension	1024
	Inference timesteps	4
Token Prediction	Num. prediction tokens	8
	CE loss weight (level 1 / level 2)	1.0 / 0.5
	Label smoothing	0.1
Training (Full Data)	Pre-train steps	80k
	Fine-tune steps	80k
	Learning rate (both stages)	5e-5

with total policy objective $\mathcal{L}_{\text{VLA}} = \lambda_{\text{ce}}\mathcal{L}_{\text{token}} + \mathcal{L}_{\text{fm}}$, where $\lambda_{\text{ce}} = 0.1$.

Table 4 summarizes the key architectural hyperparameters of VLA-UniT.

WM action condition injection. We consider two action conditioning modes for WM-UniT. The raw-action baseline injects min-max normalized state and action vectors, while WM-UniT uses continuous pre-quantization features produced by the UniT action encoder. In both cases, the conditioning signal is projected through a MLP and injected into the DiT via cross-attention (instead of AdaLN). This choice was made partly because cross-attention facilitates visualization and comparison of action feature distributions across embodiments. Text embeddings are disabled in all experiments.

WM-UniT flow matching. For world modeling, the UniT action-branch encoder produces continuous features $\tilde{z}_t^a = E_a(s_t, a_{t:t+k})$, which are projected through an MLP and injected as action conditions. Let X_t denote the latent future-frame representation. The world model is trained with:

$$\mathcal{L}_{\text{WM}} = \mathbb{E}_{\tau,\epsilon}\left[\left\|V_\phi(X_t^\tau \mid o_t,\, \text{MLP}(\tilde{z}_t^a),\, \tau) - (X_t - \epsilon)\right\|_2^2\right]. \tag{6}$$

Table 5 summarizes the key architectural hyperparameters of WM-UniT.

B Computational Resources

All main training runs were conducted on a server with 8 high-performance enterprise GPUs. Under the full-data setting, UniT tokenizer training takes roughly 0.5 day, VLA-UniT policy training roughly 1.5 days, and WM-UniT training roughly 1 day.

Table 5: **WM-UniT architectural hyperparameters.**

Component	Hyperparameter	Value
Video DiT Backbone	Base model	Cosmos Predict 2.5
	Total parameters	2.1B
	Transformer layers	28
	Hidden dimension	2048
	Attention heads	16
Action Condition Projector	Input dimension	32 (codebook dim)
	Projected dimension	1024

For real-world inference, VLA-UniT is deployed on an NVIDIA RTX 4090 workstation. Pure model inference takes roughly 0.15 s per policy update; publishing each 50-step action chunk takes 0.27 s on average, corresponding to an effective action command frequency of about 180 Hz.

Longer-Training Control. We test whether extending policy optimization closes the gap to VLA-UniT by training the tokenizer-free GR00T-Qwen2.5 baseline for 24k steps instead of 16k. Full-shot RoboCasa-GR1 success increases from 36.8% to 38.3%, compared with 66.7% for VLA-UniT at 16k steps. The modest improvement from additional baseline training indicates that the observed gap is not explained by this increase in policy-training steps. This experiment controls the policy-optimization schedule; it does not match total training FLOPs, since UniT additionally requires tokenizer training. We therefore interpret it as a longer-training baseline rather than an equal-compute comparison.

C Representation and Robustness Details

C.1 Noise Robustness Details

In-the-wild human motion capture data inevitably contains noise from sensor jitter and annotation artifacts. We evaluate whether the cross-reconstruction mechanism in UniT (Sec. 3.2), which grounds actions in visual transitions, provides implicit denoising. We inject additive Gaussian noise into EgoDex action trajectories at varying levels σ, where σ is normalized by the global action standard deviation computed across the entire dataset (i.e., the x-axis in Fig. 6 shows $\sigma/\mathrm{std}(a_{\text{global}})$). Specifically, $\mathrm{std}(a_{\text{global}})$ estimates the average dynamic range of the task. We compare three tokenizers: FAST [5], a frequency-based BPE action tokenizer; an action-only RQ-VAE tokenizer [4] that uses the same architecture as UniT but is trained on action data alone; and UniT, which jointly leverages visual and action information through cross-reconstruction.

As shown in Fig. 6, UniT exhibits the lowest relative degradation. At $\sigma = 0.2$, FAST degrades by $10.7\times$ and the action-only tokenizer by $2.7\times$ relative to clean input, while UniT degrades by only $1.7\times$. The gap between UniT and the action-only tokenizer suggests that visual grounding provides complementary information that regularizes the latent space, filtering out kinematic noise that lacks visual correspondence and yielding more robust representations for downstream deployment.

C.2 Tokenizer Codebook Analysis

We analyze whether UniT's shared codebook induces consistent distributions across three sources of heterogeneity: embodiments, modalities, and control modes. We use a consistent set of distribution metrics throughout this section: codebook usage-frequency correlation, JS divergence of pre-quantization latents, and linear probing accuracy on quantized tokens. JS divergence lies in $[0, \log 2]$ for two distributions, where 0 indicates identical distributions and larger values indicate greater separation. When analyzing vision-action branch alignment, we additionally report codebook utilization to verify that the shared vocabulary is broadly used rather than collapsed to a small subset.

Table 6 first reports cross-embodiment alignment between EgoDex human and GR1 humanoid samples. Cross-reconstruction substantially increases codebook usage correlation and reduces distributional divergence, suggesting that it is the key mechanism for pulling heterogeneous embodiments into a shared discrete space.

We next evaluate cross-modal alignment within each data source, where paired visual transitions and action trajectories are available. Table 7 uses the same distribution metrics as Table 6 and additionally reports utilization for the vision branch, action branch, and their union. Both EgoDex and GR1 show high usage-frequency correlation, low JS divergence, and near-complete codebook utilization,

Table 6: **Tokenizer-level cross-embodiment codebook alignment** between EgoDex human and GR1 humanoid samples.

Method	Usage Freq. Corr. ↑	Pre-Quant. JS Div. ↓	Linear Probe Acc. ↓
UniT w/o Cross-Recon.	0.001	0.024	0.645
UniT	**0.757**	**0.019**	**0.575**

indicating that visual effects and motion trajectories are mapped into a shared token distribution rather than separate modality-specific regions.

Table 7: **Vision-action branch alignment** within each data source.

Data Source	Usage Freq. Corr. ↑	Pre-Quant. JS Div. ↓	Linear Probe Acc. ↓	Vision Util. ↑	Action Util. ↑	Union Util. ↑
EgoDex	0.721	0.0197	0.560	94.53%	100.00%	100.00%
GR1	0.858	0.0110	0.628	97.27%	94.92%	98.83%

Beyond cross-embodiment alignment, we further examine whether UniT can unify different control modes within the same embodiment. The GR1 humanoid dataset contains trajectories in both end-effector (EEF) and joint-space control formats, which exhibit substantially different action distributions. Table 8 compares codebook alignment between EEF and joint-space samples. UniT achieves high correlation and low JS divergence across control modes, while an action-only tokenizer produces negatively correlated usage patterns with large distributional gaps. This demonstrates that visual anchoring enables UniT to map heterogeneous control representations into a consistent latent space regardless of the underlying action parameterization.

Table 8: **Cross-control-mode codebook alignment** between GR1 EEF and joint-space action formats.

Method	Usage Freq. Corr. ↑	Pre-Quant. JS Div. ↓	Linear Probe Acc. ↓
Action-Only Tokenizer	−0.021	0.285	0.990
UniT (Action Branch)	**0.857**	**0.028**	**0.760**

C.3 Downstream Representation Analysis Details

For downstream VLA representation analysis, the raw-action baseline is GR00T-Qwen2.5. For downstream world modeling, the raw-action baseline is Cosmos Predict 2.5 with raw action conditioning. Both baselines are trained on the same RoboCasa GR1 and EgoDex human-humanoid mixture as the corresponding UniT-based models.

D Benchmark and Experimental Setup

Stage-wise data usage and checkpoint scope. Simulation and real-world policy learning follow the same three-stage recipe, with setting-specific target-humanoid data (Table 9).

Human and target-humanoid data are co-trained during tokenizer training and VLA pretraining; fine-tuning uses only the corresponding target-humanoid data. The reported simulation and real-world results use setting-specific UniT checkpoints containing the corresponding humanoid data, rather than one frozen checkpoint that universally covers every embodiment. The original simulation and real-world humanoids have distinct 29-DoF and 32-DoF action spaces. The broader 4.8k-hour mixture and pretraining-unseen 76-DoF humanoid are additional scaling and adaptation studies.

Evaluation reporting. All error bars in success-rate plots denote one standard error.

D.1 RoboCasa GR1 Simulation Setup

As illustrated in Figure 14, the RoboCasa GR1 benchmark contains 24 tabletop tasks: 18 pick-and-place rearrangement tasks where the robot follows language instructions to move objects between containers, and 6 articulated tasks involving more complex interactions such as placing objects inside and subsequently closing cabinets, drawers, or microwaves.

Table 9: **Stage-wise data usage.** Simulation and real-world policy training use setting-specific target-humanoid data.

Stage	RoboCasa simulation	Real-world humanoid
Tokenizer training	EgoDex human + GR1	EgoDex human + real humanoid
VLA pretraining	EgoDex human + GR1	EgoDex human + real humanoid
VLA fine-tuning	GR1 only	Real humanoid only



Figure 14: Examples from the 24 RoboCasa GR1 Tabletop Tasks, including object rearrangement (e.g., Croissant to Box) and interaction with articulated fixtures (e.g., Bottle to Cabinet).

GR1 action space. GR1 is a dual-arm humanoid with dexterous hands, whose 29 joint-space DoFs (14 for the arms, 12 for the hands, and 3 for the waist) lead to a high-dimensional action space that makes precise bimanual dexterous manipulation challenging.

Data configurations The **Full Data** setting uses 24,000 robot trajectories (1,000 per task), while the **Few-Shot** setting uses a 10% subset with 2,400 trajectories (100 per task).

OOD evaluation. For simulation OOD evaluation, we construct three generalization suites from RoboCasa assets, as illustrated in Fig. 15: *Unseen Appearance* (18 tasks) with novel visual textures on familiar scenes; *Unseen Combinations* (23 tasks) with seen assets in novel object-container or articulated-scene configurations; and *Unseen Object Types* (32 tasks) with novel object categories.

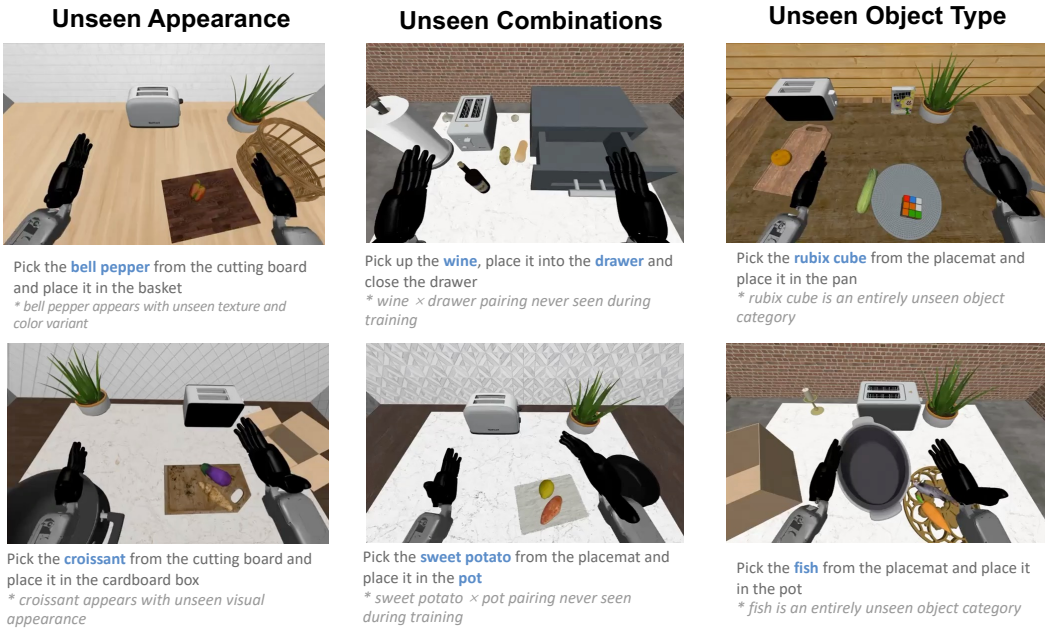


Figure 15: **RoboCasa GR1 OOD generalization suites.** Representative examples of unseen appearance, unseen object-container or articulated-scene combinations, and unseen object types used for simulation evaluation.

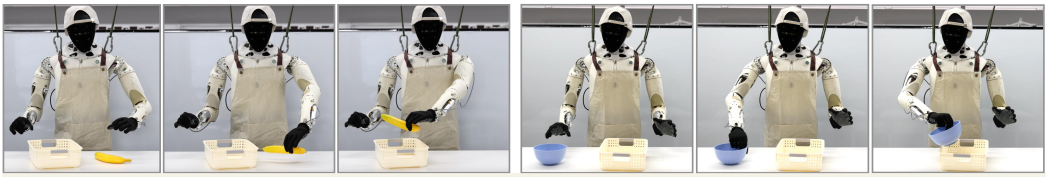
Simulation evaluation protocol. All simulation success rates are computed over independent rollouts with randomized initial object and container states. In-domain RoboCasa GR1 evaluation

covers 24 tasks with 50 episodes per task, resulting in 1,200 total evaluation episodes. The OOD suites follow the same 50-episode-per-task protocol: Unseen Appearance contains 18 tasks (900 episodes), Unseen Combinations contains 23 tasks (1,150 episodes), and Unseen Object Types contains 32 tasks (1,600 episodes). Reported category averages are computed over all episodes in the corresponding suite.

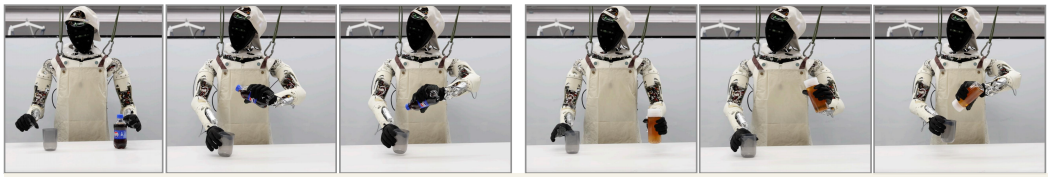
For human-to-humanoid transfer, we use the EgoDex `basic_pick_place` subset [38], which contains 27,419 pick-and-place trajectories.

D.2 Real-World Experimental Setup

Fig. 16 illustrates the real-world in-domain tasks used in Sec. 5.2.3. **Pick & Place** is analogous to the EgoDex `basic_pick_place` subset and requires placing an object into a box. **Pouring** is analogous to the EgoDex `pour` subset and requires coordinated bimanual manipulation. We collect 120 robot trajectories per task. Unless marked "w/o human data," models are pre-trained on a mixture of collected robot trajectories and EgoDex human demonstrations, then fine-tuned on task-specific robot data.



Pick and Place



Pour Water (Bimanual Cooperation)

Figure 16: **Real-world in-domain tasks.** Pick & Place and Pouring are designed to match the corresponding EgoDex task families while requiring execution on the humanoid robot.

Real-world humanoid action space The real-world humanoid extends the simulation configuration (arms, hands, waist) by incorporating an additional 3-DoF head, yielding a total of 32 joint-space degrees of freedom.

OOD evaluations. Fig. 17 shows the five real-world OOD axes. For the first four axes, robot teleoperation covers part of the variation, while EgoDex human demonstrations provide complementary conditions not sufficiently covered in robot data. **Geometry** introduces new object shapes with the same affordance; **Distractor** adds irrelevant objects around the target; **Target** changes the placement destination; and **Background** varies table or scene appearance. Beyond transfer, **Combinational** evaluates instruction-based disambiguation among multiple seen objects.

Real-world evaluation protocol. Real-world success rates are measured by physical execution trials on the humanoid robot. A trial is counted as successful only when the robot completes the instructed manipulation without human intervention and places or uses the target object according to the language instruction. The real-world evaluation uses the following category totals: 40 trials for in-domain Pick & Place, 20 trials for in-domain Pouring, 30 trials each for Background, Distractor, Geometry, and Combinational generalization, 20 trials for Target generalization, and 10 trials for Task Level Stacking Bowl generalization. The OOD settings evaluate human-introduced variations that are absent in robot teleoperation data. The zero-shot stacking task follows the same physical-trial protocol but uses no robot demonstrations of stacking during training.

E Policy Learning Details

E.1 Policy Baselines

For the policy benchmark in Sec. 5.2.1, we compare VLA-UniT with representative baselines spanning diffusion policies, flow-matching policies, VLM-based action generation, and action-token prediction.

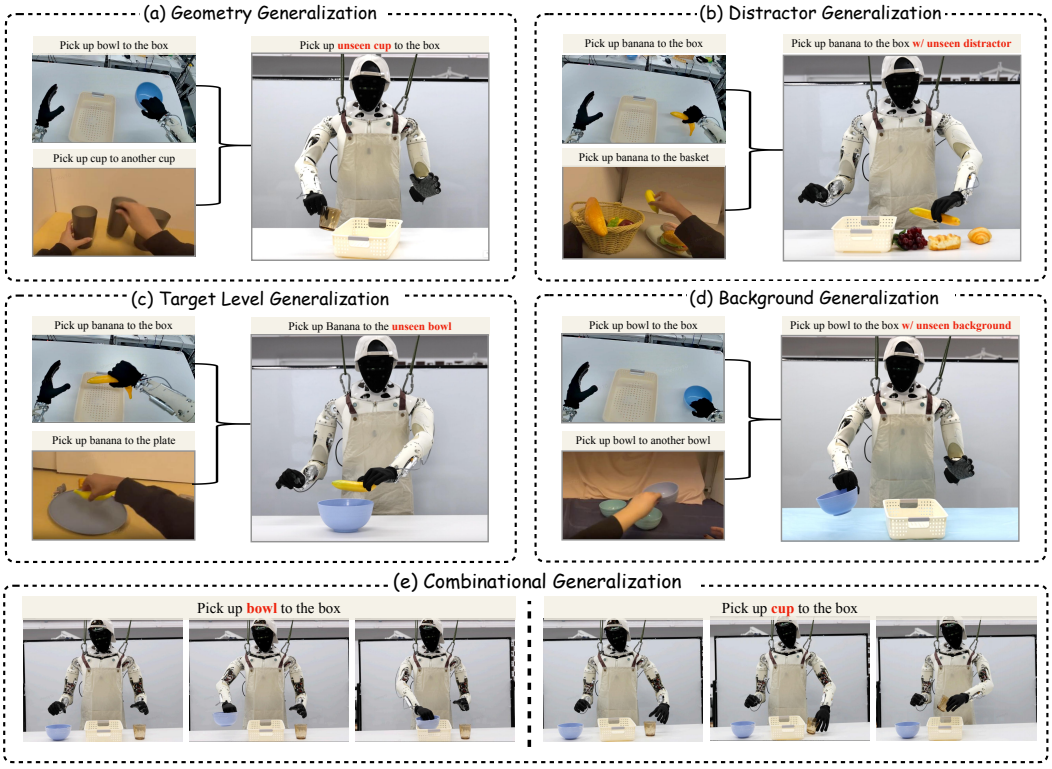


Figure 17: **Real-world OOD generalization scenarios.** Human demonstrations introduce complementary variations absent from robot data, and the humanoid is evaluated on the human-introduced conditions.

Diffusion Policy [15] models actions with U-Net denoising. **UWM** [45] unifies action and video diffusion in a transformer. **FLARE** [42] uses flow matching with future latent alignment. **GR00T-N1.6** [46], **GR00T-Qwen3** [16], and **π-Qwen3** [18] represent VLM-based action generation variants. **FAST-Qwen3** [5] predicts frequency-tokenized actions autoregressively, and **OFT-Qwen3** [17] applies an optimized fine-tuning recipe with parallel action-chunk decoding.

All RoboCasa GR1 benchmark comparisons use the same robot training split and the same evaluation protocol described in Appendix D.1: 24 tasks are evaluated with 50 independent rollouts per task, and success rates are averaged over the corresponding evaluation episodes. External baselines follow the FLARE/StarVLA benchmark protocol and are used for benchmark-level comparison.

We use two comparison types. **GR00T-Qwen2.5** is the controlled same-architecture baseline used to isolate the effect of UniT token prediction under the same backbone and action head. The remaining methods are external strong baselines following the FLARE/StarVLA protocols, used to contextualize overall performance rather than isolate a single architectural factor.

E.2 Tokenizer Ablation Variants

For tokenizer ablations in Sec. 5.4, all VLA variants share the same policy architecture, observation inputs, language inputs, action head, and training protocol. They differ only in the tokenizer used to produce the auxiliary token prediction target. This isolates whether the benefit comes from action-only tokenization, vision-only tokenization, vision-action tokenization without alignment, unidirectional visual-to-action reconstruction, or UniT's bidirectional cross-reconstruction.

VLA-Action. Following [4, 8], this variant predicts tokens from an action-only tokenizer. The tokenizer observes raw action trajectories but does not use visual transitions during token learning. It tests whether compact action discretization alone is sufficient for policy learning and cross-embodiment transfer.

VLA-Vision. Following [9, 11, 12, 47], this variant predicts tokens from a vision-only latent-action tokenizer. The tokenizer infers latent actions from visual observations without raw motor inputs. It tests whether visually inferred motion tokens provide a transferable signal, and whether the absence of fine-grained motor supervision limits control.

VLA-UniT w/o Cross-Recon. Similar to [13, 14], this variant uses a tokenizer that encodes both vision and action but removes UniT's cross-reconstruction objective. Vision and action are therefore available to the tokenizer, but they are not explicitly forced to reconstruct each other through a shared codebook. It tests whether multi-modal input alone can align human and humanoid action semantics.

VLA-Villa. This variant re-implements a Villa-X-style visual-to-action reconstruction objective [34] in our codebase. It uses unidirectional visual-to-action reconstruction rather than UniT's bidirectional cross-reconstruction. It tests whether enforcing only visual-to-action prediction is sufficient, or whether the reverse action-to-visual reconstruction is needed to ground motor tokens in visual effects.

F World Modeling Additional Results

F.1 Single-Embodiment DROID World Modeling

Table 10 reports the single-embodiment DROID [48] controllable generation results. WM-UniT keeps the full metric set and achieves the best EPE while also improving PSNR, SSIM, and LPIPS over raw-action and action-only latent conditioning.

Table 10: **Single-embodiment DROID controllable generation.** Full metrics for the DROID world-modeling evaluation.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FVD ↓	EPE ↓
Raw Action	<u>21.02</u>	<u>0.820</u>	<u>0.097</u>	**76.38**	0.2662
WM-Action	20.86	0.819	0.102	80.30	<u>0.2593</u>
WM-UniT	**21.32**	**0.823**	**0.095**	<u>76.44</u>	**0.2588**

F.2 Additional Cross-Embodiment Conditioning Results

Fig. 18 and Fig. 19 provide additional qualitative examples for the cross-embodiment world-model transfer discussed in Sec. 5.3.2. In both directions, UniT better preserves the source action semantics and temporal structure than raw-action conditioning.

Table 11: **Robot-to-human conditioning consistency.** MLLM evaluation shows that WM-UniT better preserves action semantics, temporal order, and geometric motion when robot actions condition human video generation.

Method	Semantic ↑	Temporal ↑	Geometric ↑	Overall ↑
Raw Action	2.96	3.12	2.74	2.92
WM-UniT	**3.91**	**3.98**	**3.66**	**3.84**

F.3 MLLM Evaluation Protocol

For Table 2 and Table 11, we use Gemini-3-Pro(Thinking) as an MLLM judge to evaluate each generated transfer video against its reference. Two transfer results from different methods are shown alongside the reference in a four-quadrant video layout, and the judge produces per-method scores. We evaluate 256 videos in total, with scores averaged over three independent judge runs per video. The judge scores three dimensions on a 1–5 scale:

- *Semantic* consistency: whether the transfer agent replicates the same action primitive and logical intent as the reference within its local coordinate system (e.g., reaching, pushing, grasping).
- *Temporal* consistency: whether acceleration, deceleration, and contact moments occur at matching relative timestamps, capturing the velocity profile and rhythm of the reference motion.
- *Geometric* consistency: whether the spatial trajectory and path style (e.g., arched vs. linear approach, wrist orientation) match the reference motion manifold.

The overall score averages these three dimensions. The judge is instructed to evaluate **topological correspondence**—whether the transfer agent replicates the relative interaction intent of the reference—rather than requiring identical objects or environments. The full evaluation prompt (with minor formatting edits) is provided below.

Listing 1: MLLM judge system prompt for cross-embodiment transfer evaluation.

```
# Role
You are a senior Robotics Expert AI specializing in evaluating
"Embodiment-Agnostic Action Transfer."
```

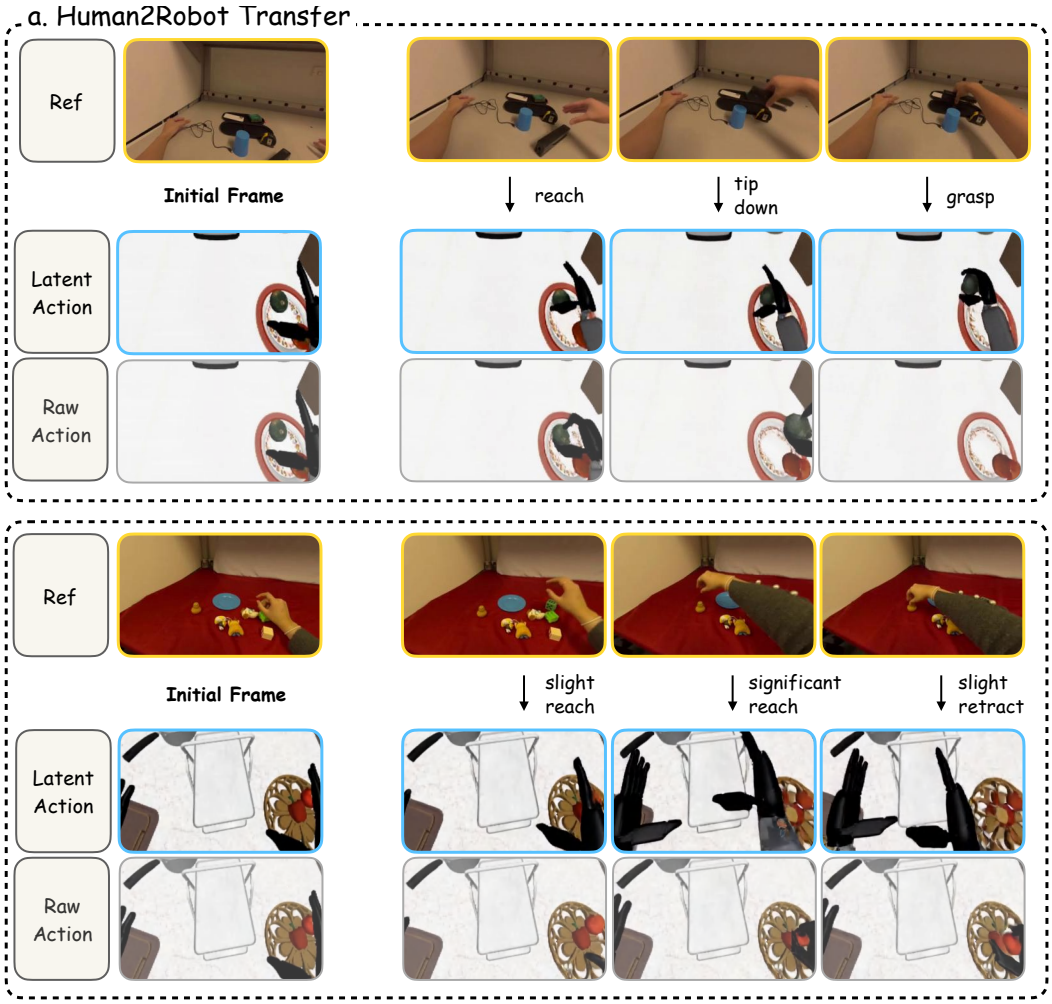


Figure 18: **Human-to-robot conditioning.** Human reference actions condition robot video generation through UniT vs. Raw Action.

```
# Core Evaluation Principle
The "ref" and "transfer" scenes may feature DIFFERENT objects and
environments. Do NOT look for identical objects. Instead, evaluate
the Topological Correspondence: Does the agent in the transfer
video replicate the Relative Interaction Intent of the reference
within its own local coordinate system?

# Advanced Physical & Kinematic Audit
1. Kinematic State Saturation: If the Reference agent performs a
   state change (e.g., closing a gripper), but the Transfer agent
   is ALREADY in that terminal state, maintaining that state is
   semantically correct. Do NOT penalize as "static."
2. Multi-Effector Tracking & Implicit Force: Pay attention to ALL
   active parts of the agent. If the environment changes in the
   Reference (e.g., a door slowly closes), identify which arm
   applied the force and look for the corresponding subtle
   movement in the Transfer methods.
3. Displacement vs. Intent: Only penalize heavily if the Reference
   shows a large, unconstrained spatial displacement AND the
   Transfer agent genuinely exhibits zero corresponding intent.
   Subtle, high-precision movements or maintaining a force/pose
   are valid actions.

# Scoring Criteria (1-5 Scale)
1. Semantic (1-5): Evaluates "Action Primitive" alignment. Does
   the agent perform the same logical intent relative to its local
   target? (e.g., if Ref is "reaching", is Transfer "reaching"?)
2. Temporal (1-5): Evaluates "Rhythm and Synchronization." Do the
```

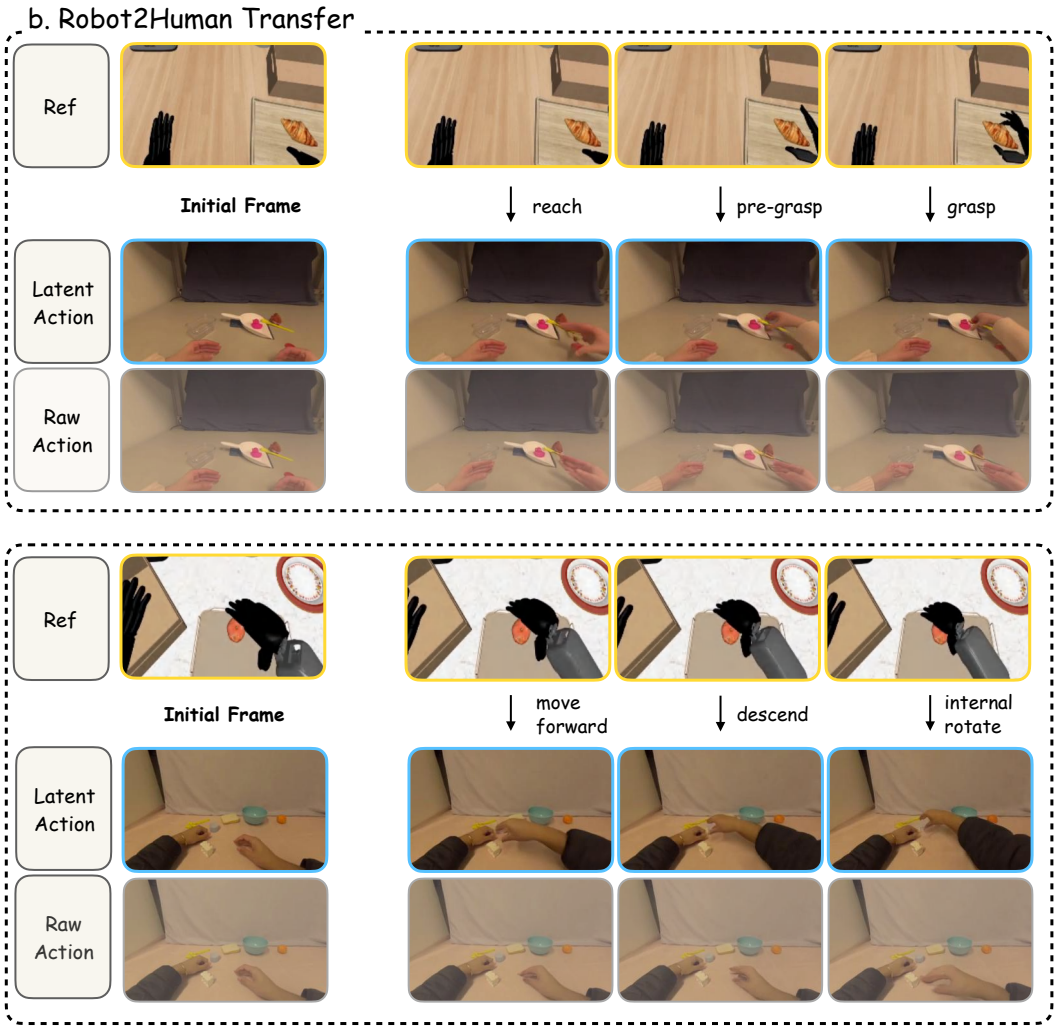


Figure 19: **Robot-to-human conditioning.** Robot reference actions condition human video generation through UniT vs. Raw Action.

```
   acceleration, deceleration, and contact moments occur at the
   same relative time? Judge if the velocity profile matches the
   reference's tempo.
3. Geometric (1-5): Evaluates "Motion Manifold and Path Style."
   Does the spatial trajectory match the Ref? Look for nuances:
   wrist orientation, non-standard paths, style details beyond
   default robotic habits.

# Execution Guidelines
- Step 1 (Fact Audit): Note what the REF is doing at key timestamps.
- Step 2 (Comparison): Contrast how Method A and Method B respond
  at those exact timestamps.
- Step 3 (Pattern-Break): Judge if the method is merely "repeating
  a dataset habit" or actually "transferring the demonstrated style."
```

G Limitations and Future Directions

Interpretation of the Shared Token Space. We use "physical language" to describe a visually anchored shared latent representation that supports cross-embodiment transfer. Codebook statistics and embedding visualizations diagnose representation alignment; they do not establish interpretable semantics for individual tokens. Cross-embodiment generation provides functional evidence of sequence-level transfer, while free composition of tokens into unseen skills has not been demonstrated.

Task and Embodiment Scope. Our main evaluation focuses on humanoid tabletop manipulation. The 76-DoF experiment extends this scope to a whole-body task, but uses target-embodiment data and fine-tuning; its ten trials per model provide preliminary, task-specific evidence rather than zero-shot deployment to an unseen robot. The zero-shot results in Sec. 5.2.3 concern unseen tasks on a trained target embodiment. Broader whole-body, in-hand, and long-horizon capabilities remain to be established.

Data Scale and Embodiment Diversity. The 4.8k-hour mixture adds human, dexterous-hand robot, and gripper-robot data from over 12 datasets/configurations. The observed gains support scaling within the evaluated mixture and target tasks, but do not establish a general scaling law or universal transfer across morphologies. The main simulation and real-world results also use setting-specific checkpoints containing the corresponding target-humanoid data, rather than one frozen checkpoint for every embodiment.

Video-Only Human Data. Our unlabeled-video experiment uses a visual branch trained through paired vision–action anchoring, followed by three hours of self-collected tabletop videos and target-robot fine-tuning. It therefore demonstrates policy pretraining with additional action-unlabeled videos, rather than learning the full tokenizer from videos alone. Scaling to unconstrained internet or third-person videos remains untested.

Training Cost. UniT requires tokenizer training in addition to policy optimization. The longer-training control in Appendix B extends the baseline policy schedule, but does not match total training FLOPs. The resulting comparison should therefore be interpreted as a control for additional policy-training steps rather than an equal-compute comparison.

Closed-Loop Policy and World Modeling. Our policy and world-model evaluations are separate: predictive video quality and cross-embodiment conditioning do not establish closed-loop planning or rollout-based policy improvement. Future work could use policies to propose latent actions, world models to simulate their visual consequences, and imagined rollouts to support planning or reinforcement learning within the shared latent action space.

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 UniT's contributions in cross-embodiment policy learning and world modeling, and the experiments evaluate these claims in simulation and real-world humanoid settings. The scope and remaining limitations are further discussed in Appendix G.

   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: Appendix G discusses the current scope of UniT, including its focus on humanoid tabletop manipulation, moderate-scale paired demonstrations, limited robot morphology diversity, and unexplored use of large-scale third-person human videos.

   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 does not present formal theoretical results, theorems, or proofs; its contributions are methodological and empirical.

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 paper provides implementation objectives, architecture and training hyperparameters, benchmark splits, evaluation protocols, and baseline details in Appendices A, D, and E. We also state that code, model weights, and benchmark environments will be released upon acceptance.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No]

   Justification: The submission does not include an anonymized public code release at review time. We provide detailed implementation and experimental protocols in the appendix, and will release code, model weights, and benchmark environments upon acceptance.

   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 main text describes the benchmark settings, evaluated variants, and metrics, while Appendices A, D, and E provide training objectives, hyperparameters, data splits, evaluation protocols, and baseline details.

   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: The paper reports error bars for success-rate plots, with one standard error defined in Appendix D. For real-world evaluation, Appendix D.2 also reports the number of physical trials per category used to compute the success rates.

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: Appendix B reports the hardware used for the main experiments and approximate wall-clock training times for UniT tokenizer, VLA-UniT policy, and WM-UniT world-model training, along with the corresponding full-pipeline GPU-day estimate and the real-world inference time.

   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: We have reviewed the NeurIPS Code of Ethics and believe the research conforms to it. The work uses cited datasets, simulation benchmarks, and controlled robot experiments without deceptive data collection or deployment to end users.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: The work is a foundational study of cross-embodiment representation learning for robot policy learning and world modeling, and is not tied to a specific public deployment or user-facing application.

    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 paper does not release a scraped dataset or a high-risk generative model for public deployment. The work focuses on embodied policy learning and world-modeling research under benchmark settings.

    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: The paper credits the existing datasets, models, and benchmarks used in the experiments through citations, including RoboCasa, EgoDex, GR00T, Qwen2.5-VL, and Cosmos Predict. These assets are publicly available under open licenses, and we follow their usage policies.

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 assets as part of the submission.

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 conduct crowdsourcing experiments or new human-subject studies. Human demonstrations are drawn from existing cited datasets or controlled robot-data collection protocols rather than crowdworker studies.

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 report new human-subject experiments requiring IRB approval. Existing human demonstration data are used through cited datasets, and the study focuses on robot learning and evaluation.

    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: [Yes]

    Justification: The paper describes the use of an MLLM judge for world-model transfer evaluation in Appendix F.3. No LLM is used as an original or non-standard component of the UniT training or inference 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.