VIOLA: LEARNING GENERALIST HUMANOID CONTROL POLICIES FROM HUMAN DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Teaching a humanoid to follow instructions with its whole body runs into two obstacles. Its action space is large and tightly coupled: legs, arms, and fingers must move together while the robot keeps its balance, which makes joint-level actions hard to learn. And humanoid demonstrations are scarce, so current humanoid generalist policies do not follow new instructions out of the box and are fine-tuned on teleoperated demonstrations of each task before deployment. Human demonstrations exist in far larger numbers, but a person's motion is not a robot command. We remove both obstacles by changing what the generalist policy predicts. We introduce VioLA, a generalist humanoid policy that predicts body and hand motion latents instead of joint commands. A pretrained body controller and hand decoder execute these latents on the robot. Their corresponding motion encoders map human and robot motion into the same latent spaces. A human recording is therefore labeled in the policy's action space, and the converted demonstration pool contains 140.6 million frames, 93.2% of them human. As a result, VioLA follows locomotion instructions on the real robot zero-shot, without task-specific fine-tuning, reaching 100% success where GR00T N1.7 and Ψ_0 reach 16.7% and 0%, respectively. It also reaches 88.6% manipulation success without task-specific fine-tuning. The same approach works across two VLA and one world-action model backbones. A generalist policy trained on human demonstrations alone performs locomotion tasks on the real robot zero-shot. Code, data pipelines, and checkpoints will be released.

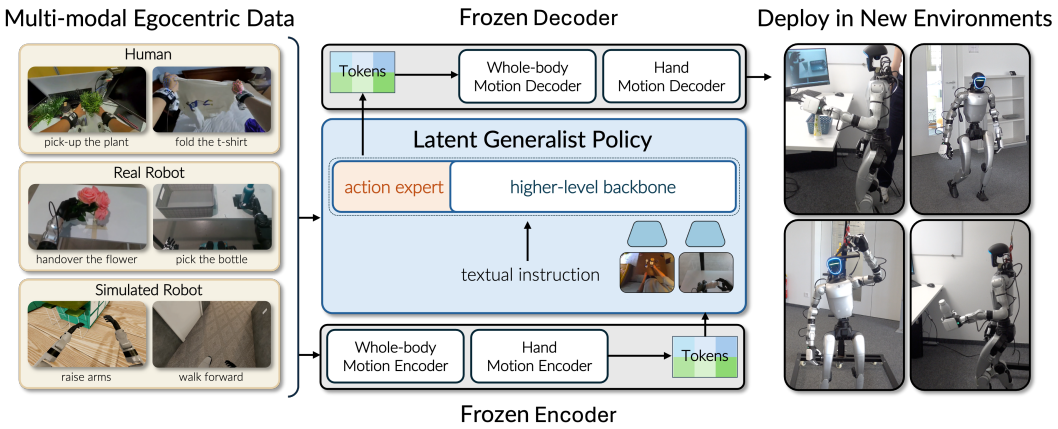


Figure 1: **VioLA.** Human and robot motion encoders produce targets in shared body and hand latent spaces. The generalist policy predicts motion latents from vision, language, and proprioception; frozen robot decoders execute its predictions. Annotated human motion therefore supervises the policy in the same action representation as robot motion.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Consider telling a humanoid robot to walk to a table and pick up a bottle. The robot must find the table, walk to it, reach for the bottle, and close its fingers around it without losing its balance. Vision-language-action models (VLAs) and world-action models (WAMs) learn to follow instructions from demonstrations, with substantial progress on robot arms (Zitkovich et al., 2023; Kim et al., 2025; Black et al., 2025a; Ye et al., 2026). Learning these behaviors on a humanoid presents two difficulties: coordinating the whole body and obtaining enough demonstrations to learn from.

Whole-body control requires the legs, torso, arms, and fingers to move together. Reaching shifts the robot's weight, and stepping changes where its hands can reach. A generalist policy that predicts joint commands directly must learn how to coordinate these movements while also learning which movement the instruction requires.

Yet the robot demonstrations available for learning both are limited. Humanoid teleoperation requires a robot and an operator, making it expensive to record many tasks in many environments. Existing humanoid generalists adapt to target tasks using additional robot demonstrations (Bjorck et al., 2025; Wei et al., 2026; Ma et al., 2026), so deploying them on a new task can require another round of data collection.

Recordings of people walking, reaching, and handling objects could expand the demonstrations available for training a humanoid generalist policy. But these recordings contain human motion, not the robot joint commands that the policy would normally learn to predict. Existing approaches obtain robot actions by having a robot follow the person (Fu et al., 2025), or use retargeted human motion to supervise the policy (Kareer et al., 2025; Qiu et al., 2025; Zheng et al., 2026b). Our approach changes the prediction target: rather than joint commands directly, the generalist policy predicts commands for a pretrained whole-body controller.

Given a desired motion and the robot's current state, a pretrained whole-body controller computes joint commands to follow the motion while maintaining balance (He et al., 2025a;b; Luo et al., 2026). Recent latent motion controllers accept the desired motion as a learned vector, or *motion latent*, produced by an encoder (Peng et al., 2022; Gehring et al., 2023; Yao et al., 2024; Luo et al., 2026). A generalist policy can then predict motion latents instead of joint commands: it chooses where and when to move, while the controller determines how the joints must move to execute that choice.

An encoder that maps human motion into the controller's latent space also supplies the missing action targets for human recordings. For example, encoding a person's walking motion produces a latent that commands walking on the robot. Paired with the person's camera image and instruction, that latent tells the generalist policy what to predict. On the robot, the generalist policy predicts a walking latent, which the controller converts into joint commands. The human demonstration can therefore teach the robot when to walk, even though the recording contains no robot joint commands.

Our method, VioLA uses these motion latents as the action space of a generalist policy (Figure 1). The policy predicts body and hand latents together from an egocentric image, an instruction, and proprioception. Pretrained human and robot motion encoders convert demonstrations into the latents the policy learns to predict; pretrained robot decoders convert its predictions into joint commands. This choice of action space has two benefits:

- **Predicting motions, not joints.** The generalist policy selects body and hand motions. The body controller uses the robot's measured posture and motion to adjust its joint commands and maintain balance, while the hand decoder converts hand latents into finger commands.
- **Human motion as supervision.** The human encoders turn recorded body and hand motion into action targets. Human images, instructions, and encoded motion train the same generalist policy as robot demonstrations, without recording a robot performing each human demonstration.

We use SONIC (Luo et al., 2026) for the body encoders and controller, and a separately pretrained hand module for the hand encoders and decoders. All remain frozen during generalist-policy training. Human recordings retain their original images and instructions; simulation supplies the robot joint positions and orientation needed as policy inputs. The converted demonstration pool contains

140.6 million frames, 93.2% from humans. Section 3 describes how encoders construct targets and the generalist policy predicts them.

On a real Unitree G1, VioLA achieves 100% locomotion success and 88.6% manipulation success in our laboratory, without task-specific fine-tuning for either. It uses GR00T N1.7 as its backbone; experiments with $\pi_{0.5}$ and DiT4DiT compare other VLA and world-action backbones. A generalist policy trained exclusively on human demonstrations reaches 40% locomotion success, showing that encoded human motion can supervise behaviors executed on the robot.

Our contributions. (1) A body-and-hand motion action space that delegates joint coordination to pretrained controllers, together with a pipeline that converts human and robot demonstrations into common training targets (Section 3). (2) Zero-shot locomotion and manipulation on a real Unitree G1 in a new environment with different object instances, without task-specific fine-tuning (Sections 4.4 and 4.5). (3) Comparisons of VLA and world-action backbones and of human and robot supervision, including real-robot locomotion learned from human demonstrations alone (Sections 4.6 and 4.3).

2 Related Work

Generalist robot policies. VLAs such as RT-2, OpenVLA, and $\pi_{0.5}$ predict robot actions from images and instructions (Zitkovich et al., 2023; Kim et al., 2025; Black et al., 2025a). WAMs such as DreamZero and DiT4DiT also predict future visual observations (Ye et al., 2026; Ma et al., 2026). GR00T and Ψ_0 apply generalist policies to humanoid hardware (Bjorck et al., 2025; Wei et al., 2026), while HumanVLA and Humanoid-VLA study language- and vision-conditioned humanoid behavior in simulation (Xu et al., 2024; Ding et al., 2025). We retain existing backbones and action experts but change their prediction targets to body and hand motion latents that pretrained robot decoders turn into joint commands. Human motion encoders produce targets in these same latent spaces, allowing human demonstrations to train the generalist policy's actions.

Motion priors and whole-body control. DeepMimic learns to reproduce reference motions (Peng et al., 2018); ASE and MoConVQ learn latent representations of movements that a controller can execute (Peng et al., 2022; Yao et al., 2024). Gehring et al. (2023) learn a model of how motion latents follow one another over time and use it to guide exploration, regularization, and temporal abstraction in downstream reinforcement learning. These methods show how learned motion spaces can simplify higher-level decisions. We use encoded motion as the target for supervised learning: the generalist policy predicts the demonstrated movement from an image and an instruction. On humanoid hardware, OmniH2O, HOVER, and SONIC use the robot's current state to track desired motions (He et al., 2025a;b; Luo et al., 2026). We use SONIC because its human and robot encoders map motion into a shared latent space that its pretrained controller can execute. This supplies both training targets and a robot controller for our generalist policy; a separate pretrained module provides the hand representation.

Hierarchical humanoid policies. Separating movement selection from joint control is established in humanoid learning. Helix 02 separates semantic reasoning, visuomotor prediction, and whole-body control (Figure AI, 2026). LeVERB first learns a vision-language latent space from rendered kinematic demonstrations, then trains a controller to execute those latents (Xue et al., 2025). Its hardware experiments replay latent sequences generated in simulation. In our method, the decoders are pretrained before the generalist policy, and their encoders label human recordings, retaining the person's original images and instructions. The generalist policy then uses the real robot's camera observations during execution. WholeBodyVLA learns latent targets from changes between video frames and grounds them in robot commands through teleoperation-based fine-tuning (Jiang et al., 2026). Our targets come from motion encoders paired with robot decoders that remain frozen during generalist-policy training. Concurrent MotionWAM also predicts SONIC body latents, alongside continuous end-effector commands (Zheng et al., 2026a). We additionally predict hand latents and use human motion to supervise body and hand predictions. HumanoidArena benchmarks hierarchical policies using egocentric observations (Wang et al., 2026); our experiments measure task success on a real humanoid.

Shared action representations. FAST compresses robot action trajectories for efficient prediction (Pertsch et al., 2025). LAPA learns latent actions from changes between video frames, then uses robot data to learn how those actions should be executed (Ye et al., 2025). Other methods share action representations between embodiments: Yan & Lee (2026) study human–robot latent control for arm reaching, UniT aligns human and humanoid representations through visual and action re-

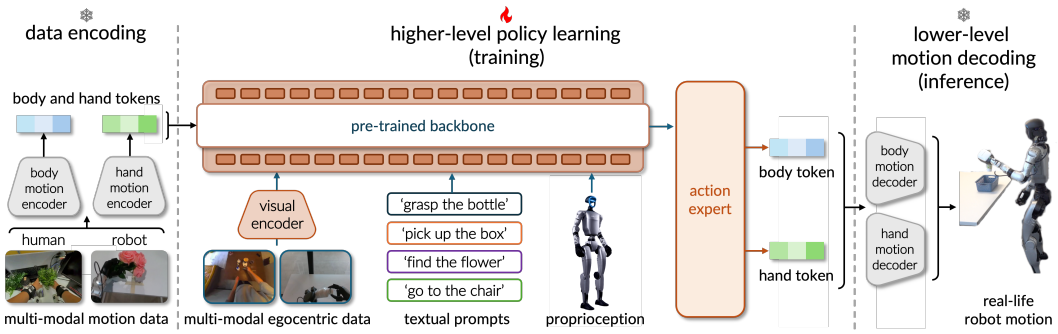


Figure 2: **VioLA overview.** *Left:* frozen motion encoders construct body and hand action targets from human or robot demonstrations. *Center:* the generalist policy predicts motion latents from an egocentric image, an instruction, and proprioception. It is the only component updated during training. *Right:* a frozen body controller and hand decoder convert the predicted latents into robot joint commands. Human and robot encoders produce targets for these same decoders.

construction (Chen et al., 2026), and UniAct learns actions shared across robots (Zheng et al., 2025). UniT trains a policy to predict shared tokens alongside an action expert that produces embodiment-specific commands. In VioLA, the generalist policy's action expert predicts the body and hand latents that frozen decoders execute. The shared representation is therefore the command passed to the pretrained controller, as well as the target extracted from a demonstration.

Learning from human demonstrations. HumanPlus records robot demonstrations by having a humanoid shadow a person (Fu et al., 2025). VideoMimic reconstructs human motion and scenes to train controllers in simulation (Allshire et al., 2025), while EgoMimic and HAT train policies using retargeted human motion (Kareer et al., 2025; Qiu et al., 2025). Being-H0, EgoVLA, and EgoScale also use human motion to supervise actions (Luo et al., 2025; Yang et al., 2025; Zheng et al., 2026b); human video is thus useful beyond the visual pretraining used by Ψ_0 (Wei et al., 2026). EgoScale learns from human wrist and retargeted hand actions, then adapts with aligned human–robot data. We encode annotated human body and hand motion into targets for a generalist policy whose predictions are executed by frozen robot decoders. This requires motion annotations and simulated robot states, but no corresponding physical robot demonstration for each human recording. Our human-only experiment tests whether this supervision suffices for instruction-following locomotion. UniHM studies language-conditioned dexterous hand manipulation (Zhang et al., 2026); our policy predicts hand and body motion for walking, reaching, and grasping.

3 VioLA

VioLA predicts body and hand motion latents from an egocentric image, an instruction, and proprioception. A motion latent specifies a desired movement to a pretrained robot decoder. Human and robot motion encoders supply the training targets; a body controller and a hand decoder execute the predictions (Figure 2). All motion encoders and decoders remain frozen; training updates only the generalist policy.

3.1 Predicting body and hand motions

Human and robot encoders share a body-motion latent space and a separate hand-motion latent space. Corresponding human and robot movements must produce latents that the robot decoder can execute as similar motions. The body decoder also reads robot state to coordinate the joints and maintain balance; the hand decoder maps its latent to finger commands. These decoders must support the movements required by the tasks. We instantiate the body modules with SONIC (Luo et al., 2026) and use a separately pretrained hand module (Appendix A.2).

Each action $\mathbf{z}_t = [\mathbf{z}_t^{\text{body}}; \mathbf{z}_t^{\text{hand}}]$ specifies the motion for the hands and the body. Given image I_t, instruction ℓ, and robot state $\mathbf{s}_t$, the generalist policy p_θ predicts a chunk of H actions:

$$\widehat{\mathbf{Z}}_t \sim p_\theta(\mathbf{Z}_t \mid I_t, \ell, \mathbf{s}_t), \qquad \mathbf{Z}_t = [\mathbf{z}_t, \ldots, \mathbf{z}_{t+H-1}]^{\mathsf{T}}. \tag{1}$$

Predicting body and hand motions together lets the policy learn their timing, such as closing the hand when the arm reaches an object. The policy selects movements appropriate to the task, while

the pretrained body controller supplies joint coordination. The encoders produce quantized codes as training targets, while the policy predicts continuous latent coordinates; Appendix A.1 gives their construction.

3.2 Learning from human and robot demonstrations

A demonstration supplies images, an instruction, and annotated motion. We select the encoders for its demonstrator $e \in \{\text{human}, \text{robot}\}$ and concatenate their body and hand outputs into the action target:

$$\mathbf{z}_t = \left[E^e_{\text{body}}(\mathbf{m}^{\text{body}}_t; \gamma_t);\ E^e_{\text{hand}}(\mathbf{m}^{\text{hand}}_t)\right]. \tag{2}$$

Here $\mathbf{m}^{\text{body}}_t$ and $\mathbf{m}^{\text{hand}}_t$ are short motion windows starting at t. The body encoder uses the simulated robot's heading γ_t to express the desired movement relative to the robot. Consecutive targets form the chunk $\mathbf{Z}_t$ that the generalist policy learns to predict.

Robot demonstrations pair these targets with recorded images, instructions, and proprioception. Human demonstrations retain the person's original images and instructions, but have no robot proprioception. We obtain it by tracking the demonstrated motion with the body controller in simulation, then pairing the simulated robot states with the corresponding human frames. Body targets for both sources are computed through this simulation; Appendix A.1 gives the conversion procedure.

For example, consider an annotated human recording of picking up a cup. The body encoder converts the reach into body latents, and the hand encoder converts the finger motion into hand latents. Simulation supplies robot proprioception aligned with the recording. The person's camera image and instruction, together with that state, train the generalist policy to predict the reach and grasp latents. The image remains the human recording, and no physical robot demonstration of that example is needed.

We train the generalist policy to predict motion-latent chunks from images, instructions, and robot proprioception, using the same prediction head and loss for human and robot demonstrations. Motion targets are computed before training; examples without hand annotations supervise only the body predictions.

VioLA uses a GR00T N1.7 (NVIDIA, 2026) backbone and an action expert conditioned on visual-language features and robot state. The action expert learns to predict motion-latent chunks with the existing flow-matching objective (Lipman et al., 2023). Variants with $\pi_{0.5}$ and DiT4DiT backbones use the same targets; DiT4DiT also retains its future-video prediction objective (Black et al., 2025a; Ma et al., 2026). Appendix A.4 gives the losses and treatment of missing annotations.

3.3 Executing the predicted motion

At each control step, the body controller converts the predicted body latent and recent robot state $\mathbf{h}_t$ into joint commands. The hand decoder converts the hand latent into finger commands:

$$\mathbf{u}^{\text{body}}_t = D_{\text{body}}\left(\widehat{\mathbf{z}}^{\text{body}}_t, \mathbf{h}_t\right), \qquad \mathbf{u}^{\text{hand}}_t = D^{\text{robot}}_{\text{hand}}\left(\widehat{\mathbf{z}}^{\text{hand}}_t\right). \tag{3}$$

The body controller updates at 50 Hz, adjusting its commands as the robot moves. Meanwhile, the generalist policy uses new observations to predict the next chunk. We apply real-time chunking (RTC) (Black et al., 2025b) in motion-latent space, conditioning the next prediction on the latents scheduled for execution while it is being computed. Appendix A.5 gives execution details, and Section 4.7 evaluates continuity across chunk switches.

4 Experiments

We first examine the motion predictions and the contribution of human supervision, then evaluate zero-shot locomotion, manipulation, and alternative policy backbones.

4.1 Setup

We evaluate VioLA on a Unitree G1 with five-fingered Inspire hands in our laboratory, with scenes and objects different from the training recordings. All policies are deployed without task-specific fine-tuning. We test six locomotion and posture tasks and seven manipulation tasks, with five trials each. Trials begin from the same idle standing pose. Locomotion trials allow 5 s, or 10 s for Bend

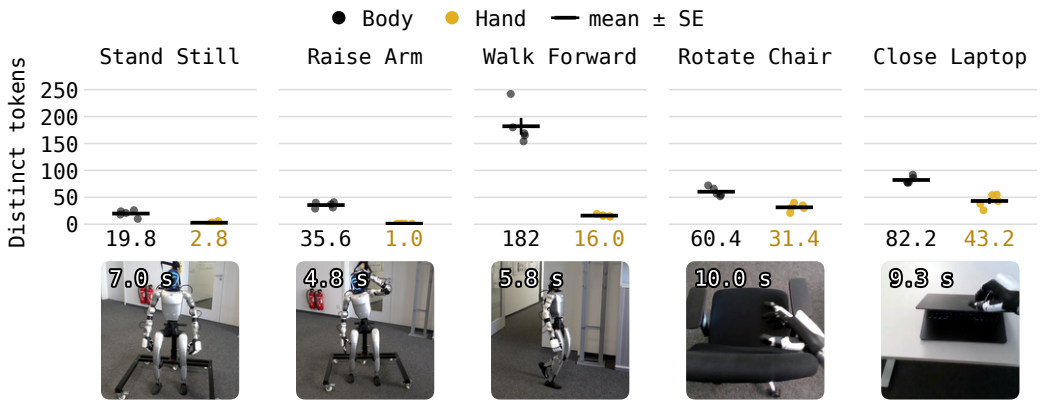


Figure 3: **Motion token counts.** Each dot shows the number of distinct body (black) or hand (gold) representatives over one recorded execution. Horizontal black marks show the mean and vertical whiskers show one standard error over five recordings per task; numbers below give mean counts and durations. Counts cover the full recording.

Down; manipulation trials have no fixed time limit. Success requires completing the instruction, while progress also credits partial completion. We give each task equal weight and report one standard error. Appendix A.6 gives task descriptions and scoring details.

The training pool contains 140.6 million frames, 93.2% from human demonstrations. VioLA uses GR00T N1.7 and trains for 150k steps on human data followed by 50k steps with uniform sampling across five human and robot corpora (Appendix A). We compare policies trained on different combinations of human and robot data, the released GR00T N1.7 (NVIDIA, 2026) and Ψ_0 (Wei et al., 2026) base models, and policies using $\pi_{0.5}$ and DiT4DiT backbones.

4.2 Predicting motions, not joints

How many distinct motion tokens describe the movements that VioLA predicts to complete a task? We examine recorded executions of standing, arm raising, walking, chair rotation, and laptop closing. Because the generalist policy predicts continuous latents, numerically different predictions can encode similar poses. For counting, we group nearby predictions and let one predicted latent represent each group. We count each group as one distinct token, with separate counts for the body and hands. The grouping thresholds are calibrated by comparing the poses decoded from original predictions and their representatives (Appendix A.11).

Over a full chair-rotation recording, VioLA's predictions can be described by an average of 60 body tokens and 31 hand tokens. Closing a laptop takes 82 body and 43 hand tokens under the same grouping rule. Walking requires more distinct body tokens—182 on average—and 16 hand tokens. Figure 3 averages five recordings per task; chair rotation and laptop closing last 10.0 and 9.3 s on average. For context, 10 s of execution at 50 Hz contains 500 latent samples for each of the body and hands. These counts describe the policy's motion predictions across complete recordings, rather than individual time steps. They show that tens of distinct tokens can summarize the predictions for the two manipulation tasks, while pretrained controllers supply the joint commands. The robot executes the original predictions; these counts summarize them after grouping, not execution with fewer tokens.

4.3 Learning from human demonstrations

We test whether human demonstrations teach behaviors that transfer to the robot by training VioLA on human data alone. We then compare human-then-mixed training with robot-only training (Figure 4).

Human-only supervision. Without any robot demonstration in generalist-policy training, VioLA succeeds in 12/30 locomotion trials (40%). It bends down in 4/5 trials, walks forward in 3/5, sits in 2/5, and stands up in 3/5. It does not successfully follow the standing-still or arm-raising instructions. Figure 4 shows the human-only policy walking to a yellow couch and squatting down before returning to standing. Human demonstrations therefore transfer several instructed behaviors to the robot, but do not yet support the full task set.

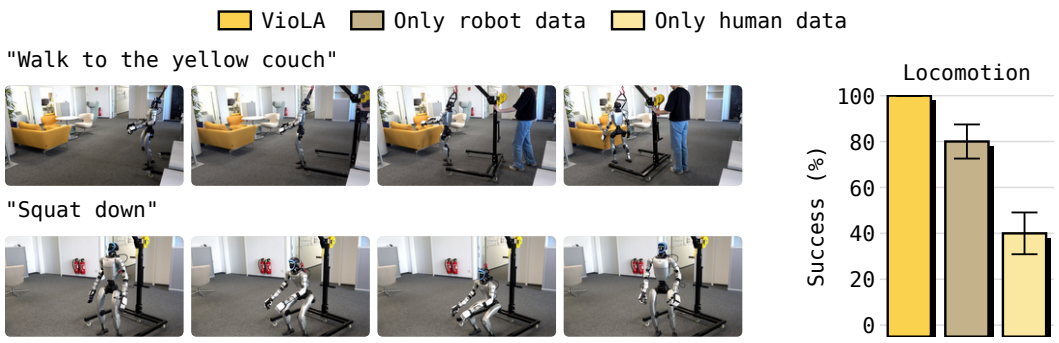


Figure 4: **Learning from human data.** Left: VioLA trained only on human demonstrations follows "walk to the yellow couch" (top) and "squat down" (bottom). Right: locomotion success with human-then-mixed training (30/30 trials), robot-only training (24/30), and human-only training (12/30). Error bars show one standard error over trials. All three use VioLA trained on different combinations of human and robot data, without task-specific fine-tuning.

Adding human supervision. After 200k training steps, VioLA completes 30/30 locomotion trials, compared with 24/30 (80%) for the variant trained for 200k steps on robot data alone. The robot-only variant fails all five bending trials and keeps its arms extended in one sitting trial, receiving partial credit. Human-then-mixed training therefore improves locomotion success by 20 percentage points at the same total number of training steps. Together with human-only transfer, this comparison supports using human motion as supervision for humanoid generalist policies.

4.4 Zero-shot locomotion

VioLA completes all 30 locomotion trials (Figure 5, top), following instructions to stand still, raise an arm, walk, bend, sit, and stand up. The released GR00T N1.7 succeeds only at standing still, reaching 5/30 successes (16.7%); Ψ_0 fails all five trials of each task (0/30). These results show that VioLA transfers the six behaviors to our laboratory without task-specific adaptation.

4.5 Zero-shot manipulation

VioLA achieves 88.6% success on manipulation (31/35), also without task-specific fine-tuning. It completes all trials of Rotate Chair, Close Laptop, Place Dumpling, Close Kettle, and Push Duck (Figure 5, bottom). The released GR00T N1.7 completes none of these manipulation trials. Appendix A.8 shows successful executions by VioLA and a failed trial in which it grasps the coat but drops it.

Failures concentrate on Hang Coat (4/5 successes) and Store Shampoo (2/5). In the three failed shampoo trials, the robot knocks the bottle over while reaching to grasp it. An additional bottle-carrying task fails in all five trials because the bottle falls at release. Appendix A.7 analyzes these grasping and placement failures.

4.6 Comparing policy backbones

We compare GR00T N1.7 and $\pi_{0.5}$, two VLA backbones, with the DiT4DiT world-action backbone under VioLA (Figure 6). All learn from the same demonstration pool and are evaluated on the same humanoid tasks without task-specific fine-tuning. VioLA and the variant with a DiT4DiT backbone each complete all 30 locomotion trials. The $\pi_{0.5}$ backbone succeeds in 20/30: it completes standing, arm raising, sitting, and standing up, but fails bending and walking. Manipulation separates the backbones more strongly: VioLA reaches 88.6% success, compared with 45.7% with $\pi_{0.5}$ and 25.7% with DiT4DiT.

Acquiring and retaining objects remains difficult for the other backbones. All three policies rotate the chair in 5/5 trials, but neither $\pi_{0.5}$ nor DiT4DiT completes the coat or shampoo tasks. Partial progress exceeds success for both: $\pi_{0.5}$ reaches 51.4% manipulation progress and DiT4DiT 29.4%, reflecting correct initial movements followed by failure.

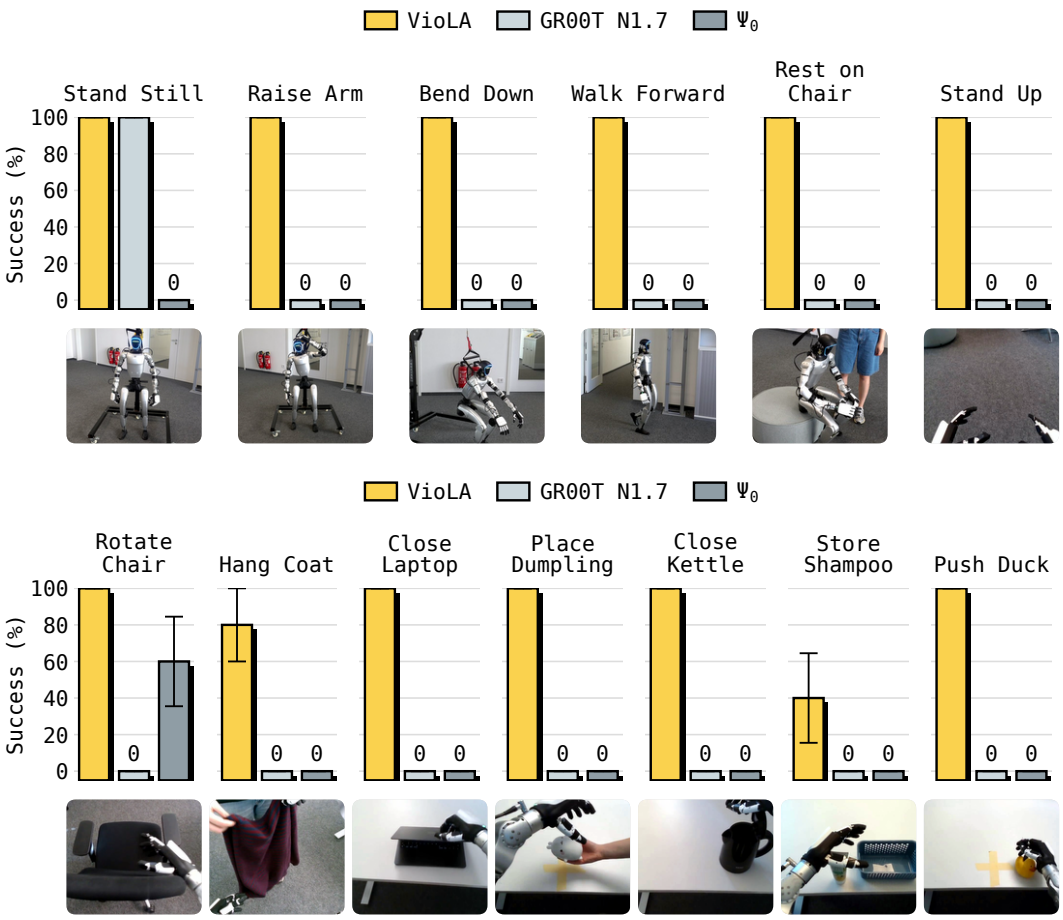


Figure 5: **One policy follows locomotion and manipulation instructions without task-specific fine-tuning.** VioLA completes 30/30 locomotion trials (top) and 31/35 manipulation trials (bottom). Manipulation failures occur on Hang Coat (4/5 successes) and Store Shampoo (2/5). Bars show success over five trials per task, with one standard error. Photos illustrate the tasks.

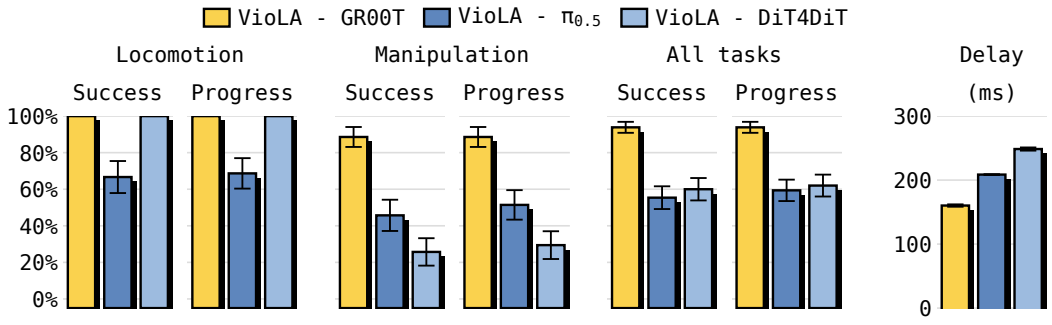


Figure 6: **Comparing backbones for humanoid instruction following.** All three policies use VioLA and are evaluated without task-specific fine-tuning. Success requires completing the instruction; progress credits partial completion. Performance is averaged over 30 locomotion trials, 35 manipulation trials, or all 65. VioLA and the variant with a DiT4DiT backbone succeed in every locomotion trial; the GR00T N1.7 backbone achieves the highest manipulation success. The final panel shows prediction delay. Error bars indicate one standard error over trials for performance and over episodes for delay.

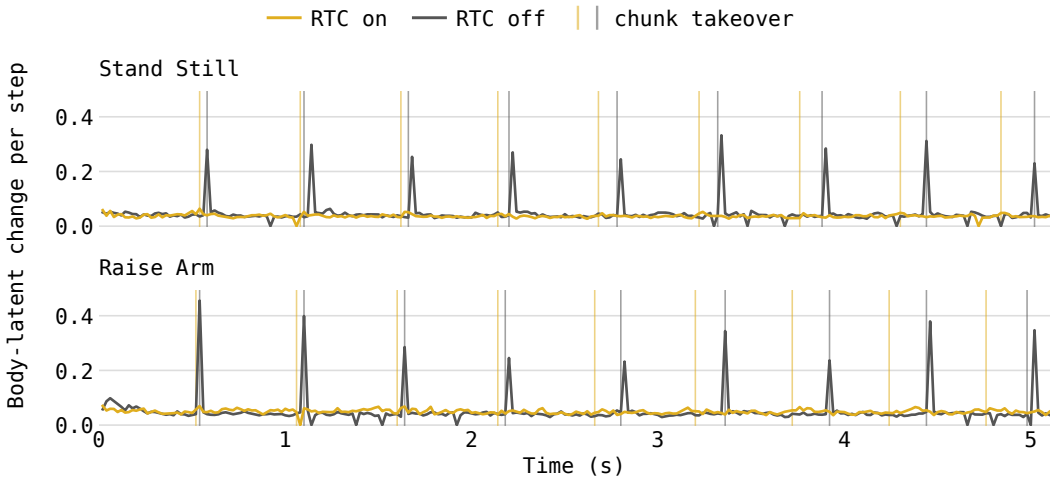


Figure 7: **RTC reduces jumps between predicted motion chunks.** Each row compares VioLA recordings with RTC on (gold) and off (gray). Curves show the L2 change in the 64-coordinate body latent between consecutive 20 ms control steps. Faint vertical lines mark when execution switches to a newly predicted chunk, in the corresponding condition's color. RTC-off traces show sharp spikes near chunk switches, whereas RTC-on traces remain close to their between-switch variation. Each recording is aligned to its first chunk switch and shown for 5.1 s. Time is measured from the recording timestamps; the initial idle-to-policy transition is excluded.

4.7 Continuity across motion chunks

While VioLA computes a new action chunk, the robot continues executing the previous one. If their predictions disagree at the switch, the commanded motion can jump. We apply real-time chunking (RTC) (Black et al., 2025b) to the predicted motion latents before decoding them into joint commands. Figure 7 compares RTC on and off for Stand Still and Raise Arm, measuring $\Delta_t = \|\mathbf{z}_t^{\text{body}} - \mathbf{z}_{t-1}^{\text{body}}\|_2$ between consecutive 50 Hz control steps.

RTC reduces the sharp changes near chunk switches in these recordings. Over the displayed 5.1 s windows, the largest change falls from 0.332 to 0.064 for Stand Still and from 0.455 to 0.070 for Raise Arm. These measurements assess prediction continuity; Appendix A.10 gives recording selection and timing details.

5 Limitations and Conclusion

Limitations. VioLA delegates motion execution to frozen decoders, so its range of behaviors depends on their motion coverage. The remaining manipulation failures also show that coordinated movement alone is insufficient for reliable object interaction: the robot can reach a bottle yet knock it over, or transport it to a table but release it while tilted. The hand decoder predicts finger positions without modeling contact forces, motivating tighter integration of grasping and contact feedback. Human supervision introduces a different constraint. It requires motion annotations and pairs human images with simulated robot states; unannotated video cannot directly provide these training targets. Human-only training transfers several behaviors to the robot but follows instructions less consistently than mixed training. Improving motion annotation and the correspondence between human observations and robot execution could help close this gap.

Conclusion. VioLA learns whole-body instruction following by predicting body and hand motion latents while pretrained controllers produce the joint commands. Because human and robot motion can be encoded into these same targets, human recordings directly supervise the generalist policy. On a real Unitree G1, VioLA achieves 100% locomotion success and 88.6% manipulation success without task-specific fine-tuning. Human-only supervision yields 40% locomotion success, and human-then-mixed training improves locomotion over robot-only training from 80% to 100%. These results support learning task-directed movements from human and robot demonstrations while reusing pretrained control skills for execution.

AI USE STATEMENT

We used generative AI tools for code creation, manuscript editing, figure preparation, and experimental feedback. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Arthur Allshire, Hongsuk Choi, Junyi Zhang, David McAllister, Anthony Zhang, Chung Min Kim, Trevor Darrell, Pieter Abbeel, Jitendra Malik, and Angjoo Kanazawa. Visual imitation enables contextual humanoid control. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 794–815, 2025. URL https://proceedings.mlr.press/v305/allshire25a.html.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi 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. *arXiv preprint arXiv:2503.14734*, 2025. URL https://arxiv.org/abs/2503.14734.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40, 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/300ccb2187dedd4edcc07f7e76d8e553-Abstract-Conference.html.

Boyu Chen, Yi Chen, Lu Qiu, Jerry Bai, Yuying Ge, and Yixiao Ge. UniT: Toward a unified physical language for human-to-humanoid policy learning and world modeling. *arXiv preprint arXiv:2604.19734*, 2026. URL https://arxiv.org/abs/2604.19734.

Pengxiang Ding, Jianfei Ma, Xinyang Tong, Binghong Zou, Xinxin Luo, Yiguo Fan, Ting Wang, Hongchao Lu, Panzhong Mo, Jinxin Liu, Yuefan Wang, Huaicheng Zhou, Wenshuo Feng, Jiacheng Liu, Siteng Huang, and Donglin Wang. Humanoid-VLA: Towards universal humanoid control with visual integration. *arXiv preprint arXiv:2502.14795*, 2025. URL https://arxiv.org/abs/2502.14795.

Figure AI. Introducing Helix 02: Full-body autonomy. Project website, January 2026. URL https://www.figure.ai/news/helix-02. Published January 27, 2026.

Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. HumanPlus: Humanoid shadowing and imitation from humans. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2828–2844, 2025. URL https://proceedings.mlr.press/v270/fu25a.html.

Jonas Gehring, Deepak Gopinath, Jungdam Won, Andreas Krause, Gabriel Synnaeve, and Nicolas Usunier. Leveraging demonstrations with latent space priors. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=OzGIu4T4Cz.

Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris M. Kitani, Changliu Liu, and Guanya Shi. OmniH2O: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 1516–1540, 2025a. URL https://proceedings.mlr.press/v270/he25b.html.

Tairan He, Wenli Xiao, Toru Lin, Zhengyi Luo, Zhenjia Xu, Zhenyu Jiang, Jan Kautz, Changliu Liu, Guanya Shi, Xiaolong Wang, Linxi Fan, and Yuke Zhu. HOVER: Versatile neural whole-body controller for humanoid robots. In *IEEE International Conference on Robotics and Automation*, 2025b. URL https://research.nvidia.com/labs/lpr/publication/he2025hover/.

Haoran Jiang, Jin Chen, Qingwen Bu, Li Chen, Modi Shi, Yanjie Zhang, Delong Li, Chuanzhe Suo, Chuang Wang, Zhihui Peng, and Hongyang Li. WholeBodyVLA: Towards unified latent VLA for whole-body loco-manipulation control. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2512.11047.

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 *IEEE International Conference on Robotics and Automation*, 2025. URL https://egomimic.github.io/.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Lightwheel AI. EgoStandard. Dataset, 2026. URL https://huggingface.co/datasets/LightwheelAI/EgoStandard. Accessed September 10, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):248:1–248:16, 2015. URL https://smpl.is.tue.mpg.de/.

Hao Luo, Yicheng Feng, Wanpeng Zhang, Sipeng Zheng, Ye Wang, Haoqi Yuan, Jiazheng Liu, Chaoyi Xu, Qin Jin, and Zongqing Lu. Being-H0: Vision-language-action pretraining from large-scale human videos. *arXiv preprint arXiv:2507.15597*, 2025. URL https://arxiv.org/abs/2507.15597.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Fernando Castañeda, Sirui Chen, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, Jinhyung Park, David Sami, Zi Wang, Xingye Da, Runyu Ding, Cyrus Hogg, Lina Song, Edy Lim, Eugene Jeong, Tairan He, Haoru Xue, Wenli Xiao, Simon Yuen, Jan Kautz, Yan Chang, Umar Iqbal, Linxi Fan, and Yuke Zhu. SONIC: Supersizing motion tracking for natural humanoid whole-body control. *Science Robotics*, 11(117):eaed4592, 2026. doi: 10.1126/scirobotics.aed4592. URL https://arxiv.org/abs/2511.07820.

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. DiT4DiT: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026. URL https://arxiv.org/abs/2603.10448.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/e2dd53601de57c773343a7cdf09fae1c-Abstract-Conference.html.

NVIDIA. GR00T-N1.7-3B. Model card and pretrained checkpoint, 2026. URL https://huggingface.co/nvidia/GR00T-N1.7-3B. Accessed September 11, 2026.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. DeepMimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics*, 37(4):143:1–143:14, 2018. doi: 10.1145/3197517.3201311. URL https://xbpeng.github.io/projects/DeepMimic/index.html.

Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. ASE: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions on Graphics*, 41(4), 2022. URL https://xbpeng.github.io/projects/ASE/index.html.

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. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.012. URL https://www.roboticsproceedings.org/rss21/p012.html.

Ri-Zhao Qiu, Shiqi Yang, Xuxin Cheng, Chaitanya Chawla, Jialong Li, Tairan He, Ge Yan, David J. Yoon, Ryan Hoque, Lars Paulsen, Ge Yang, Jian Zhang, Sha Yi, Guanya Shi, and Xiaolong Wang. Humanoid policy $\sim$ human policy. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 2888–2906, 2025. URL https://proceedings.mlr.press/v305/qiu25a.html.

Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6):245:1–245:17, 2017. URL https://mano.is.tue.mpg.de/.

Unitree Robotics. UnifoLM-WBT dataset. Dataset collection, 2026. URL https://huggingface.co/collections/unitreerobotics/unifolm-wbt-dataset. Accessed September 11, 2026.

Taowen Wang, Zikang Xie, Bin Yang, Yunheng Wang, Zizhao Yuan, Yuetong Fang, Yixiao Feng, Yichi Wang, Xingyu Chen, Haodong Chen, Qiwei Wu, Weisheng Xu, Lihan Chen, Lusong Li, Zecui Zeng, and Renjing Xu. HumanoidArena: Benchmarking egocentric hierarchical whole-body learning. *arXiv preprint arXiv:2606.17833*, 2026. URL https://arxiv.org/abs/2606.17833.

Songlin Wei, Hongyi Jing, Boqian Li, Zhenyu Zhao, Jiageng Mao, Zhenhao Ni, Sicheng He, Jie Liu, Xiawei Liu, Kaidi Kang, Sheng Zang, Weiduo Yuan, Marco Pavone, Di Huang, and Yue Wang. Ψ_0: An open foundation model towards universal humanoid loco-manipulation. *arXiv preprint arXiv:2603.12263*, 2026. URL https://arxiv.org/abs/2603.12263.

Xinyu Xu, Yizheng Zhang, Yong-Lu Li, Lei Han, and Cewu Lu. HumanVLA: Towards vision-language directed object rearrangement by physical humanoid. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/215aeb07b5996c969c0123c3c6ee8f54-Abstract-Conference.html.

Haoru Xue, Xiaoyu Huang, Dantong Niu, Qiayuan Liao, Thomas Kragerud, Jan Tommy Gravdahl, Xue Bin Peng, Guanya Shi, Trevor Darrell, Koushil Sreenath, and Shankar Sastry. LeVERB: Humanoid whole-body control with latent vision-language instruction. *arXiv preprint arXiv:2506.13751*, 2025. URL https://arxiv.org/abs/2506.13751.

Yashuai Yan and Dongheui Lee. Learning a unified latent space for cross-embodiment robot control. *arXiv preprint arXiv:2601.15419*, 2026. URL https://arxiv.org/abs/2601.15419.

Ruihan Yang, Qinxi Yu, Yecheng Wu, Rui Yan, Borui Li, An-Chieh Cheng, Xueyan Zou, Yunhao Fang, Xuxin Cheng, Ri-Zhao Qiu, Hongxu Yin, Sifei Liu, Song Han, Yao Lu, and Xiaolong Wang. EgoVLA: Learning vision-language-action models from egocentric human videos. *arXiv preprint arXiv:2507.12440*, 2025. URL https://arxiv.org/abs/2507.12440.

Heyuan Yao, Zhenhua Song, Yuyang Zhou, Tenglong Ao, Baoquan Chen, and Libin Liu. MoConVQ: Unified physics-based motion control via scalable discrete representations. *ACM Transactions on Graphics*, 43(4):144:1–144:21, 2024. doi: 10.1145/3658137. URL https://doi.org/10.1145/3658137.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/45d74e190008c7bff2845ffc8e3facd3-Paper-Conference.pdf.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi Fan, and Joel Jang. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. URL https://arxiv.org/abs/2602.15922.

Zhenhao Zhang, Jiaxin Liu, Ye Shi, and Jingya Wang. UniHM: Unified dexterous hand manipulation with vision language model. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2603.00732.

Zhenyu Zhao, Hongyi Jing, Xiawei Liu, Jiageng Mao, Abha Jha, Hanwen Yang, Rong Xue, Sergey Zakharov, Vitor Guizilini, and Yue Wang. Humanoid everyday: A comprehensive robotic dataset for open-world humanoid manipulation. In *IEEE International Conference on Robotics and Automation*, 2026. URL https://arxiv.org/abs/2510.08807.

Jia Zheng, Teli Ma, Yudong Fan, Zifan Wang, Shuo Yang, and Junwei Liang. MotionWAM: Towards foundation world action models for real-time humanoid loco-manipulation. *arXiv preprint arXiv:2606.09215*, 2026a. URL https://arxiv.org/abs/2606.09215.

Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. Universal actions for enhanced embodied foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2501.10105.

Ruijie Zheng, Dantong Niu, Yuqi Xie, Jing Wang, Mengda Xu, Yunfan Jiang, Fernando Castañeda, Fengyuan Hu, You Liang Tan, Letian Fu, Trevor Darrell, Furong Huang, Yuke Zhu, Danfei Xu, and Linxi Fan. EgoScale: Scaling dexterous manipulation with diverse egocentric human data. *arXiv preprint arXiv:2602.16710*, 2026b. URL https://arxiv.org/abs/2602.16710.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

VIOLA: LEARNING GENERALIST HUMANOID CONTROL POLICIES FROM HUMAN DATA

SUPPLEMENTARY MATERIAL

A IMPLEMENTATION DETAILS

A.1 MOTION REPRESENTATION AND DATA CONVERSION

Token coordinates and reference windows. Each encoder produces a 64-coordinate motion code, formed by flattening two 32-dimensional finite scalar quantization (FSQ) codes (Mentzer et al., 2024). Coordinates take values in $\{k/16 : k = -16, \ldots, 15\}$. Body and hand codes use this same format, but have separate encoders and decoders. A code summarizes a reference window beginning at the current frame. At 50 Hz, the SONIC v1.1 robot encoder and the hand encoders read ten frames at offsets $0, 5, \ldots, 45$; the human body encoder reads ten consecutive frames at offsets $0, 1, \ldots, 9$. Windows extending beyond the demonstration repeat its final frame. This reference window determines what one code represents; the policy's chunk length H determines how many successive codes it predicts.

Motion targets. Equation 2 concatenates the body and hand encoder outputs. In this implementation, $\mathbf{z}_t \in \mathcal{G}^{128}$ with $\mathcal{G} = \{k/16 : k = -16, \ldots, 15\}$. The human body encoder reads SMPL motion (Loper et al., 2015); human hand motion is obtained by fitting MANO (Romero et al., 2017) to annotated keypoints. Each training example pairs the chunk with an image, instruction, and robot state:

$$\big(I_t,\ \ell,\ \mathbf{s}_t,\ \mathbf{Z}_t\big), \qquad \mathbf{Z}_t = [\mathbf{z}_t, \ldots, \mathbf{z}_{t+H-1}]^\mathsf{T}, \tag{4}$$

Robot examples use recorded proprioception, while human examples use simulated robot states. The body targets for both sources are constructed through the simulation procedure below. This converts annotated human motion into latent targets without optimizing a separate mapping to robot joints for each recording.

Training example construction. Body targets are generated by tracking each demonstration with the frozen controller in simulation. This step is necessary because the encoder describes the desired motion relative to the robot's current heading. If the robot turns more slowly than the reference, the next code must reflect that difference. At each simulation step, the encoder uses the current heading to compute a code, the controller decodes it into joint commands, and the simulator advances:

$$\mathbf{z}_t^{\text{body}} = E^e_{\text{body}}\Big(\mathbf{m}_t^{\text{body}};\, \gamma(\mathbf{s}_t^{\text{sim}})\Big), \qquad \mathbf{s}_{t+1}^{\text{sim}} = \text{Sim}\Big(\mathbf{s}_t^{\text{sim}},\, D_{\text{body}}(\mathbf{z}_t^{\text{body}}, \mathbf{h}_t^{\text{sim}})\Big). \tag{5}$$

Each code is stored before the simulator advances and paired with the demonstration image at the corresponding time. Human examples use the simulated robot state as proprioception; the unified robot-data pipeline retains recorded robot proprioception. The converted human pool contains 20,680 of the 26,382 source episodes, excluding 5,702 during conversion and filtering.

Hand targets come from commanded finger motion where available, while hand proprioception comes from measured motion. This distinction matters during grasping: the commanded fingers can close farther than the object permits them to move. Examples without hand annotations contribute only to the body loss (Equation 7). Motion targets are computed before generalist-policy training; training does not run the simulator or update either motion module.

Closed-loop body encoding. SONIC v1.1 uses a 1,751-coordinate encoder input containing the reference motion in the simulated robot's heading frame. The body decoder receives the current 64-coordinate code and ten frames of proprioceptive history: joint offsets, joint velocities, base angular velocity, projected gravity, and previous actions. Together these form 994 input coordinates. The decoder's 29 outputs are converted to joint-position targets using the controller's default pose and action scale. Control runs at 50 Hz, with four 200 Hz physics steps per update.

Source	Episodes	Frames	Hours	Targets
Humanoid Everyday	4,064	2,962,764	16.46	Body and hands
PSI	835	1,431,454	7.95	Body and hands
UnifoLM-WBT	2,854	4,454,240	24.75	Body and hands
LeVERB	3,555	713,279	3.96	Body only
EgoSuite	20,680	131,010,575	727.84	Body and hands
Total	31,988	140,572,312	780.96	

Table 1: **Converted demonstration pool.** Humanoid Everyday, PSI, and UnifoLM-WBT contain real-G1 demonstrations; LeVERB contains simulated G1 demonstrations, and EgoSuite contains human recordings. All are converted to 50 Hz, with hand targets where available. Hours are frames divided by 50 Hz. Human data are 64.6% of episodes and 93.2% of frames. Sampling weights, not corpus size, determine exposure to each source during training.

Available converted data. The training pool combines human recordings from EgoSuite, derived from EgoStandard (Lightwheel AI, 2026), with four robot corpora: Humanoid Everyday (Zhao et al., 2026), PSI (Wei et al., 2026), UnifoLM-WBT (Unitree Robotics, 2026), and LeVERB (Xue et al., 2025). Table 1 lists the retained demonstrations before the training/evaluation split. The unified loader reserves 338 Humanoid Everyday episodes for evaluation. It also excludes episodes too short to supply a complete action chunk; at $H = 50$, 46 of the 3,555 LeVERB episodes provide no training window. Humanoid Everyday and UnifoLM-WBT supply commanded hand motion, whereas PSI supplies measured hand motion because commanded trajectories are unavailable. Human hand targets are derived from fitted MANO trajectories. LeVERB supplies body targets only.

VioLA trains for 150k steps on human demonstrations, followed by 50k steps with sampling probability 0.20 for each of the five corpora (Figure 8). The robot-only variants use the four robot corpora throughout and are evaluated after 50k and 200k steps. The human-only variant uses human demonstrations throughout. No variant receives task-specific adaptation. With a constant batch size across stages, the VioLA schedule assigns an expected 80% of training examples to human data, distinct from their 93.2% share of the available frames.

A.2 Hand motion interface

A pretrained hand module maps human hand motion and robot hand motion into a shared 64-coordinate space. Its Dex3 and Inspire decoders translate these codes into finger trajectories for the corresponding hardware. All hand encoders and decoders remain frozen during generalist-policy training. The policy learns to predict their codes; it does not learn the hand prior itself.

A.3 Backbone and action-expert instantiations

Each body–hand policy predicts 128 coordinates per action step: 64 for the body and 64 for the hands. VioLA predicts $H = 40$ successive steps; the other integrations use $H = 50$ (Table 2). State inputs differ by backbone. The $\pi_{0.5}$ configuration uses 29 body joint offsets, three gravity coordinates, 14 current finger positions expressed in the Dex3 representation, and a 64-coordinate hand code delayed by 50 ticks. The one-second delay prevents the code's reference window from exposing future targets. During training, this delayed code is zeroed with probability 0.5; it can also be zeroed during deployment. Current finger state is reconstructed from measured hand motion, using frame zero of the hand decoder when conversion is needed. Section A.9 describes a discrepancy between the $\pi_{0.5}$ training configuration and its recorded evaluation outputs.

GR00T N1.7 (NVIDIA, 2026) combines a Cosmos-Reason2 vision-language backbone with a flow-matching action transformer. Its state encoder receives 29 body joint offsets, 14 finger positions, and three gravity coordinates, with state dropout of 0.2. Human and robot examples use the same action expert and embodiment tag. All methods directly predict native FSQ coordinates without normalization. State inputs are normalized with statistics fitted on training data.

$\pi_{0.5}$ discretizes the normalized values into 256 bins for its input prompt, whereas DiT4DiT projects them into learned features.

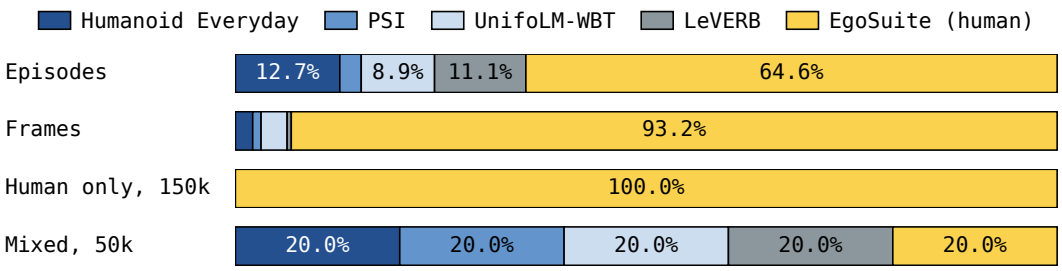


Figure 8: **Dataset size and VioLA training schedule.** The first two bars show each source's share of episodes and frames in the converted pool. The last two show sampling probabilities during 150k steps of human-only training and 50k steps of uniform sampling over all five corpora. Human recordings supply 93.2% of available frames; the training schedule determines how often they are sampled.

DiT4DiT uses Cosmos-Predict2.5-2B, with features from layer 17 conditioning the action transformer. During training, the video model receives supervision from the current frame and four future frames at offsets $12, 24, 36, 48$ on the 50 Hz timeline. The objective adds video and motion flow-matching losses, and gradients from the action loss also update the video backbone. At deployment, only the current observation is available; future video features are predicted by the model.

A.4 Flow-matching conventions and masks

The action expert v_ψ is conditioned on backbone features and robot state, collectively denoted $\mathbf{c}_t$. Flow matching mixes each target with noise and trains the expert to predict the difference. Inference starts from noise and integrates the learned velocity toward a clean action chunk. Encoder codes are treated as continuous coordinates in this regression.

For a target chunk $\mathbf{Z}_t$, noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ of the same shape, and a noise level $\tau \in [0, 1]$,

$$\mathbf{X}_{t,\tau} = (1 - \tau)\mathbf{Z}_t + \tau\boldsymbol{\epsilon}, \qquad \mathbf{U}_t = \boldsymbol{\epsilon} - \mathbf{Z}_t, \tag{6}$$

and the action expert v_ψ is trained to output $\mathbf{U}_t$ from $\mathbf{X}_{t,\tau}$:

$$\mathcal{L}_{\text{motion}} = \mathbb{E}_{\boldsymbol{\epsilon},\tau}\left[\sum_{n\in\mathcal{B}}\sum_{k=0}^{H-1}\sum_{j=1}^{128} w_{nkj}\big(v_\psi(\mathbf{X}_{t,\tau}, \tau \mid \mathbf{c}_t)_{n,kj} - U_{n,kj}\big)^2\right], \tag{7}$$

where n indexes the examples in a batch $\mathcal{B}$, each with its own target chunk, noise, and conditioning, k the step within the chunk, and j the latent coordinate. The weight w_{nkj} normalizes the loss and is zero for coordinates without a target, such as the hand coordinates of a recording without hand annotations. The reductions below specify each backbone's normalization. Human and robot examples pass through the same action expert and the same loss; no explicit human–robot source label is supplied to the action expert. Because all targets are precomputed, training never runs the motion modules or a simulator.

DiT4DiT is a world-action model: it is trained to predict the future video frames $\mathbf{V}_t$ that follow the current image, as well as the actions. We keep that objective. Its video branch is trained with the same flow-matching loss as Equation 7, with the future frames in place of the motion chunk, and the two losses are summed:

$$\mathcal{L}_{\text{WAM}} = \mathcal{L}_{\text{video}}(\mathbf{V}_t \mid I_t, \ell) + \mathcal{L}_{\text{motion}}(\mathbf{Z}_t \mid I_t, \ell, \mathbf{s}_t). \tag{8}$$

For a human clip, the recorded future frames supervise video prediction, and the encoded body and hand motion supervises action prediction.

The loss excludes invalid frames and missing hand annotations. Let N denote batch size and $m_{nkj} \in \{0, 1\}$ indicate whether coordinate j at chunk step k in example n has a valid target. For $\pi_{0.5}$, Equation 7 uses

$$w_{nkj} = \frac{m_{nkj}}{NH \max(1, \sum_{j'} m_{nkj'})}. \tag{9}$$

Backbone family	H	State conditioning	Motion-target scaling
$\pi_{0.5}$	50	110 values, tokenized	Identity
GR00T N1.7	40	46 values, state encoder	Identity
DiT4DiT	50	32 values, state encoder	Identity

Table 2: **Backbone-specific interfaces.** The motion spaces are shared across models; chunk length, state processing, and target scaling follow each integration.

Thus each step first averages over its annotated coordinates. GR00T and DiT4DiT instead average over all valid coordinates in the batch:

$$w_{nkj} = \frac{m_{nkj}}{\sum_{n',k',j'} m_{n'k'j'}}. \tag{10}$$

GR00T adds 10^{-6} to the denominator, and DiT4DiT requires at least one valid coordinate. For fully annotated batches, the reductions are equivalent up to the numerical offset. With missing hand annotations, the first gives equal weight to action steps, while the second gives equal weight to annotated coordinates.

Equation 6 uses $\tau = 1$ for noise and $\tau = 0$ for clean targets. The $\pi_{0.5}$ implementation samples $\tau = 0.999b + 0.001$, with $b \sim \mathrm{Beta}(1.5, 1)$. DiT4DiT samples $\tau = b/0.999$ with the same beta parameters and discretizes time into 1,000 conditioning buckets. GR00T uses the opposite time direction: its implementation samples $r = 0.999(1-b)$, forms $(1-r)\boldsymbol{\epsilon} + r\mathbf{Y}$, and regresses $\mathbf{Y} - \boldsymbol{\epsilon}$. Setting $\tau = 1-r$ and reversing the velocity sign gives the convention of Equation 6. Each backbone retains its own time sampler and numerical solver settings.

A.5 Decoder execution

The body controller updates its proprioceptive history and joint commands every 20 ms, while the generalist policy computes its next chunk. The hand decoder reconstructs ten frames from each hand latent. Execution selects frame index five, applies a five-tick moving average, and converts the result to the hand's joint commands. Frame zero is used separately to reconstruct current hand state.

The Unitree G1 uses five-fingered Inspire hands, with six actuated joints per hand. Predicted latents pass through the Inspire decoder to produce 12 finger targets. For policies trained with a 14-value Dex3 hand state, measured Inspire motion is encoded and then reconstructed with the Dex3 decoder. This conversion changes the state representation presented to the policy; the execution targets still use the Inspire decoder.

Although encoder targets lie on the FSQ grid, the action experts predict continuous coordinates. The $\pi_{0.5}$ execution path clips predictions to $[-1, 1]$ and passes them directly to the decoders without rounding. The hand decoder is selected for the robot's hardware before converting its output to finger commands.

Chunk replacement and RTC. While the decoders execute a chunk, the generalist policy uses a new image and robot state to predict its replacement. If the predictions disagree where the new chunk takes over, the commanded motion can change abruptly. RTC operates on motion latents, conditioning the next prediction on the part of the current chunk that will execute before inference finishes. Keeping these latents fixed helps the new chunk continue the current motion. Section 4.7 compares transitions with and without RTC.

A.6 Evaluation protocol

Instructions follow the task descriptions in LeVERB and Humanoid Everyday. All tasks are evaluated in our laboratory with locally available objects rather than the physical objects in the datasets. Each policy is tested five times per task from the same idle standing pose. Locomotion trials allow 5 s, except Bend Down, which allows 10 s. Manipulation trials have no fixed time limit. A frozen SONIC v1.1 controller and a hand decoder execute motion predictions at 50 Hz.

Success requires completing the instruction. Locomotion trials test standing, raising an arm, bending, walking forward, remaining seated, or rising from a chair. Standing up and then walking still

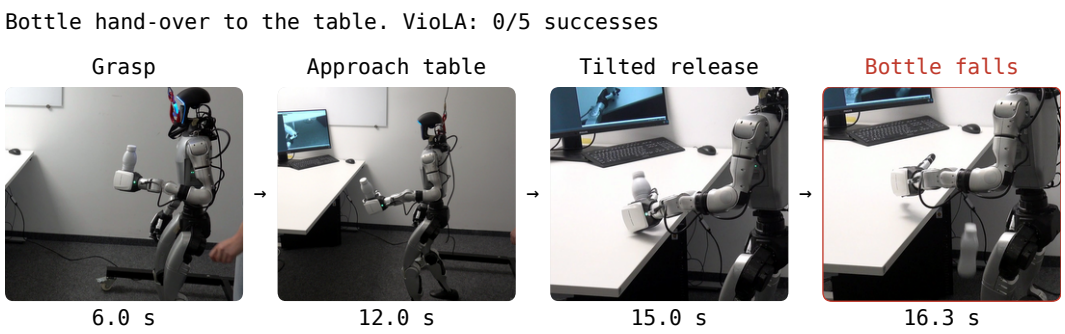


Figure 9: **Bottle carrying fails at release.** One VioLA trial from a third-person camera. The robot takes the bottle from a person, walks to the table, and opens its hand while the bottle is tilted; the bottle then falls off the table. Times are seconds from the start of the clip. The task scores 0/5 and is reported separately from the seven manipulation tasks in Figure 5.

counts as success; sitting and immediately standing again does not. Manipulation trials require rotating the chair, folding and returning the coat, closing the laptop or kettle lid, placing the dumpling on the desk, putting the shampoo in the container, or pushing the duck to the desk's center. Progress also credits partial completion: grasping the coat but dropping it scores 0.5, and beginning an insufficient bend scores 0.3. Only a score of one counts as success. We average scores over five trials per task, giving each task equal weight; error bars show one standard error.

A.7 Failures in grasping and object transport

Bottle carrying. Bottle carrying requires the robot to take a bottle from a person with its left hand, walk to a table, and place it upright. VioLA fails all five trials. It approaches the table but releases the bottle while tilted, causing it to fall. The $\pi_{0.5}$, DiT4DiT, and robot-only 200k GR00T policies also fail all five trials. This task is analyzed separately from the seven manipulation tasks in the main figure; including it reduces VioLA's manipulation success from 31/35 (88.6%) to 31/40 (77.5%).

The policies fail at different stages. Robot-only GR00T grasps the bottle but does not start walking toward the table. The $\pi_{0.5}$ policy repeatedly reaches toward it but rarely closes its hand. DiT4DiT closes its hand before reaching the bottle. Together, these failures show that grasp timing, the transition from grasping to walking, and stable release remain difficult.

Other manipulation failures. VioLA also fails three of five shampoo trials by knocking the bottle over during the reach, and one of five coat trials. In the failed coat trial (Figure 10), the robot acquires the coat but drops it before completing the instruction. These cases distinguish reaching the object from maintaining control of it through placement or handover.

A.8 Example executions

Figure 10 shows VioLA bending down, closing a laptop, and placing a dumpling toy. The successful manipulation sequences coordinate the arm's approach with hand contact and object movement. In a failed coat trial, the robot grasps the coat but drops it before completing the instruction.

A.9 Execution timing

Timing is measured over 161 GR00T episodes, 83 $\pi_{0.5}$ episodes, and 57 DiT4DiT episodes, including recordings beyond the five scored trials per task. For each episode, prediction delay is the mean time from requesting a chunk to receiving it. The replanning interval is the mean time between consecutive chunk arrivals, measured from their frame indices at 50 Hz. Table 3 averages these episode-level measurements, giving each episode equal weight.

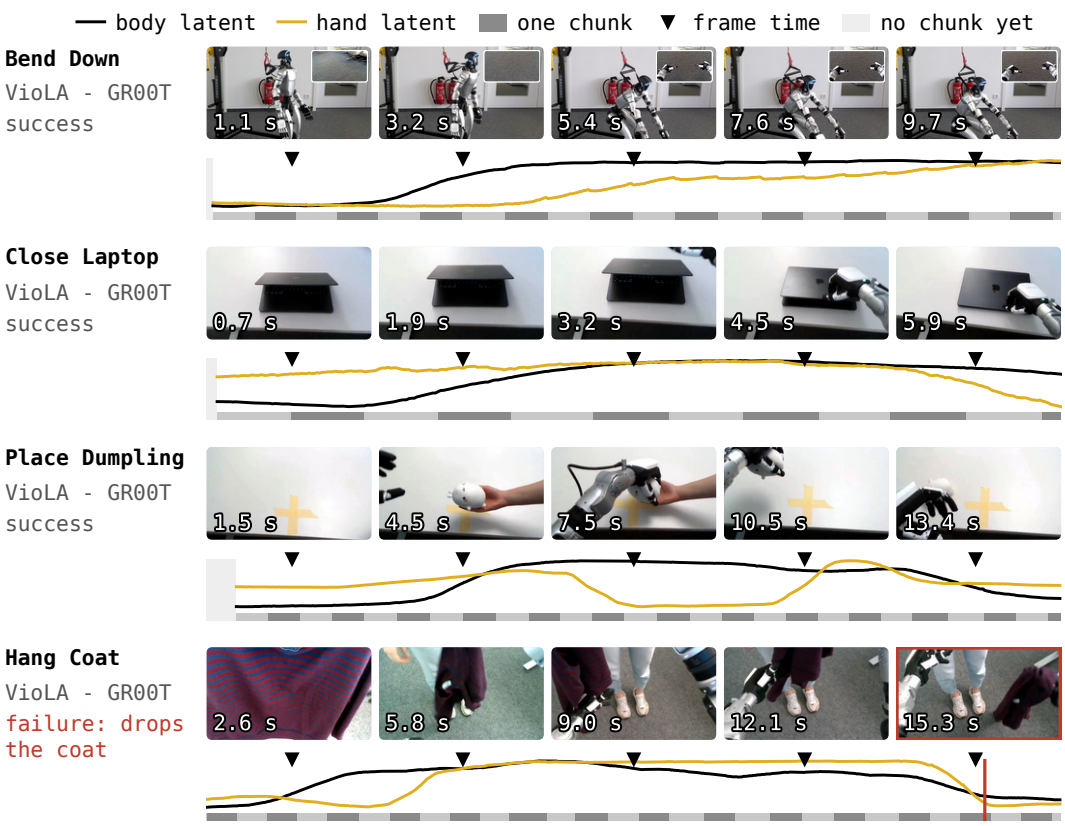


Figure 10: **From body motion to object interaction.** VioLA bends down, closes a laptop, and places a dumpling toy on the desk (first three rows). In a failed coat trial, VioLA grasps the coat but then drops it (last row; failure marked in red). Black and gold curves show the first principal component of each trial's executed body and hand latents (63–91% and 47–87% of their variance), scaled separately within each row. Gray bands mark successive action chunks; triangles align image timestamps with the curves.

A.10 Measuring chunk-transition continuity

For each task and RTC condition, Figure 7 uses the first recording in chronological order with at least four seconds of policy execution. All four traces are cropped to 5.1 s, the shortest selected duration rounded down to a tenth of a second. This selection does not use latent-change magnitude. The metric is computed directly from the logged, continuous body latents, without grid rounding or projection. Consecutive steps are identified by recording timestamps separated by 20 ms, with a 1 ms tolerance. Differences across timestamp gaps are excluded; the four selected recordings contain no such gaps. Repeated or skipped policy-counter values do not exclude a sample when its recording timestamp remains consecutive. The first policy command is excluded because its preceding command belongs to the idle pose token.

Each chunk-switch marker indicates when execution begins using a new prediction. It is placed at the timestamp of the first recorded sample whose policy counter reaches the new chunk's logged takeover index. Elapsed time is measured relative to the first policy sample.

The selected recordings run at 50 Hz. Mean prediction delays are 138.9 ms with RTC and 154.1 ms without RTC for Stand Still, and 130.0 ms and 155.3 ms for Raise Arm. Corresponding mean chunk-arrival intervals are approximately 537 and 552 ms for Stand Still, and 529 and 553 ms for Raise Arm, measured from the timestamp-aligned takeover markers. Thus the traces compare recorded executions with slightly different timing, rather than imposing identical inference delays. The reported maxima describe these four examples, not averages over repeated trials.

Backbone	Episodes	Chunks	Delay (ms)	Replan interval (ms)
GR00T	161	3,871	160.2 ± 1.8	559.9 ± 1.8
$\pi_{0.5}$	83	2,554	208.6 ± 0.6	608.4 ± 0.6
DiT4DiT	57	1,035	248.5 ± 2.5	648.0 ± 2.4

Table 3: **Measured chunk timing.** Mean $\pm$ one standard error across recorded episodes. Delay measures the time from requesting to receiving a prediction; the replanning interval measures time between successive chunk arrivals. The episode counts exceed the scored-trial counts because the recording archive includes additional runs.

A.11 Motion token counts and calibration

Counting distinct motion tokens. We count the distinct representatives needed to describe the policy's body and hand predictions over a task execution. For grouping, each latent $\mathbf{z}$ is mapped to integer grid coordinates, $\mathbf{q} = \mathrm{clip}(\mathrm{round}(16\mathbf{z}), -16, 15)$, with distance $d(\mathbf{q}, \mathbf{q}') = \sum_j |q_j - q'_j|$. We scan each recording in time order, retaining the first prediction. A subsequent prediction is assigned to its nearest retained representative if the distance is at most t; otherwise, it becomes a new representative. Ties select the earliest representative. Representatives are original predicted vectors, not averages or grid-rounded vectors. Body and hand predictions are grouped separately, with a new set of representatives for each execution. We report counts per execution, with duration for context, and average counts across executions of each task.

Decoded-pose calibration. Calibration uses 84 episodes: 32 chair-rotation and 32 laptop-closing demonstrations, plus 10 policy recordings each of Place Dumpling and Push Duck. The demonstration latents are encoder targets; the policy latents are continuous predictions. A further 21 episodes are held out for validation, with 8, 8, 3, and 2 episodes from the respective tasks. We test body thresholds in $\{0, 1, 2, 4, 8, 16, 32, 64\}$ and extend the hand grid with $\{96, 128, 192, 256, 384, 512\}$. At each threshold, we group the latents and compare the decoded poses of each original latent and its representative at the same robot state. Body error is the mean root-relative position difference over the 14 links used by SONIC's MPJPE-L metric. Hand error is the mean position difference over six Dex3 distal-finger link centers of mass, used as fingertip proxies, with the decoder evaluated at frame zero. Each episode's error is the 95th percentile of its framewise mean point errors. Invalid frames and frames at or after a simulated fall are excluded.

Tracking-error references. We use tracking accuracy to set a physical reference scale for the grouping error. For the body, the reference is SONIC's reported real-world MPJPE-L of 25.7 mm (2.57 cm) over 14 links (Luo et al., 2026). For the hands, we measure the discrepancy between commanded and observed finger joint poses in 83 Humanoid Everyday episodes (chair rotation, laptop closing, and chair-alt). Forward kinematics gives a mean error of 1.44 cm over six fingertip proxies and 32,516 frames. This hand reference uses the same points as the grouping analysis. These values are tracking-error reference budgets, not task-success tolerances or bounds on the effect of replacing a complete motion sequence.

Threshold selection. For each modality, we find the largest tested threshold for which at least 95% of calibration episodes have their episode error within the tracking-error reference budget. This rule permits thresholds of 16 for the body and 96 for the hands; requiring every calibration episode to pass gives the same values. Episodes receive equal weight, and none are removed as outliers.

We report finer thresholds of 8 for the body and 32 for the hands because averaging over links or fingertips can hide larger deviations at individual points. Reducing the body threshold from 16 to 8 lowers the mean episode-level wrist p95 from 5.09 to 3.03 cm on calibration episodes. Reducing the hand threshold from 96 to 32 lowers the worst-finger p95 from 1.08 to 0.31 cm for chair rotation and from 1.25 to 0.31 cm for laptop closing. At these finer thresholds, both modalities satisfy their mean-error budgets in all 84 calibration episodes, all 21 held-out episodes, and three additional chair-alt transfer episodes. Figure 3 reports counts over five recordings per task at these thresholds.

Figures 11 and 12 report the error curves and sensitivity of counts to the grouping threshold. The robot executes the original continuous predictions; the analysis describes them at a resolution referenced to tracking accuracy. It measures local differences in decoded pose, not accumulated execution error. Hand calibration uses Dex3 fingertip proxies rather than the deployed Inspire hand trajectory.

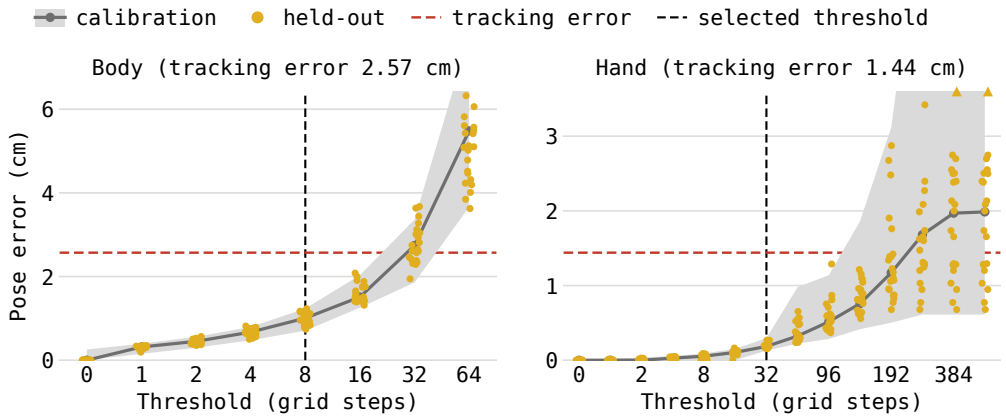


Figure 11: **Threshold calibration.** Each episode is summarized by the 95th percentile of framewise error, averaged over 14 body links or six fingertip proxies. The gray line is the error bound covering at least 95% of calibration episodes; shading spans their error range. Gold dots show held-out episodes. Red horizontal lines mark the tracking-error references (2.57 cm body, 1.44 cm hand). Vertical dashed lines mark the reported thresholds of 8 and 32, chosen finer than the permitted 16 and 96 to reduce wrist and individual-finger deviations.

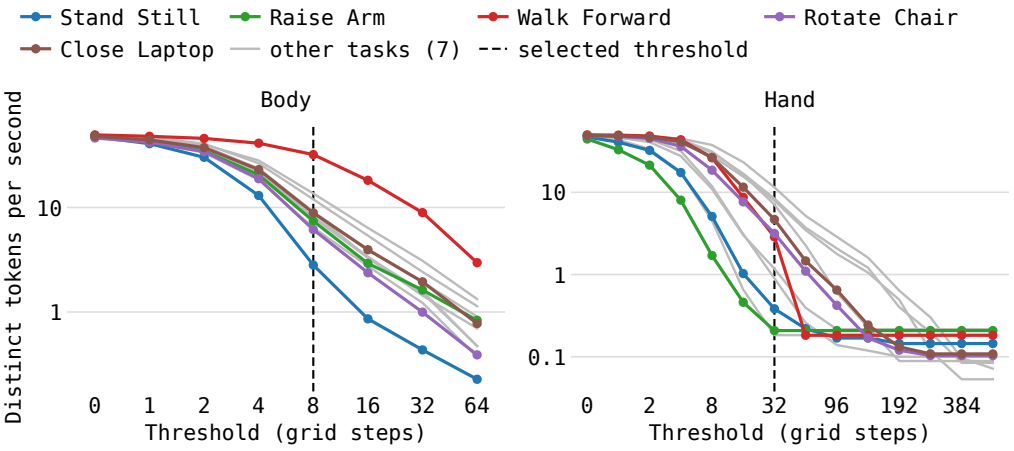


Figure 12: **Token counts vs. threshold.** Curves show how grouping distance changes the number of distinct body and hand representatives. Each recording's count is divided by its duration before averaging within each task; the vertical axis is logarithmic. Colored curves identify the five tasks in Figure 3; gray curves show seven additional tasks. Dashed lines mark the reported body and hand thresholds of 8 and 32.