Z-SIGHT: UNIFYING VISION-LANGUAGE-ACTION AND LATENT WORLD MODELING

Anonymous authors
Paper under double-blind review

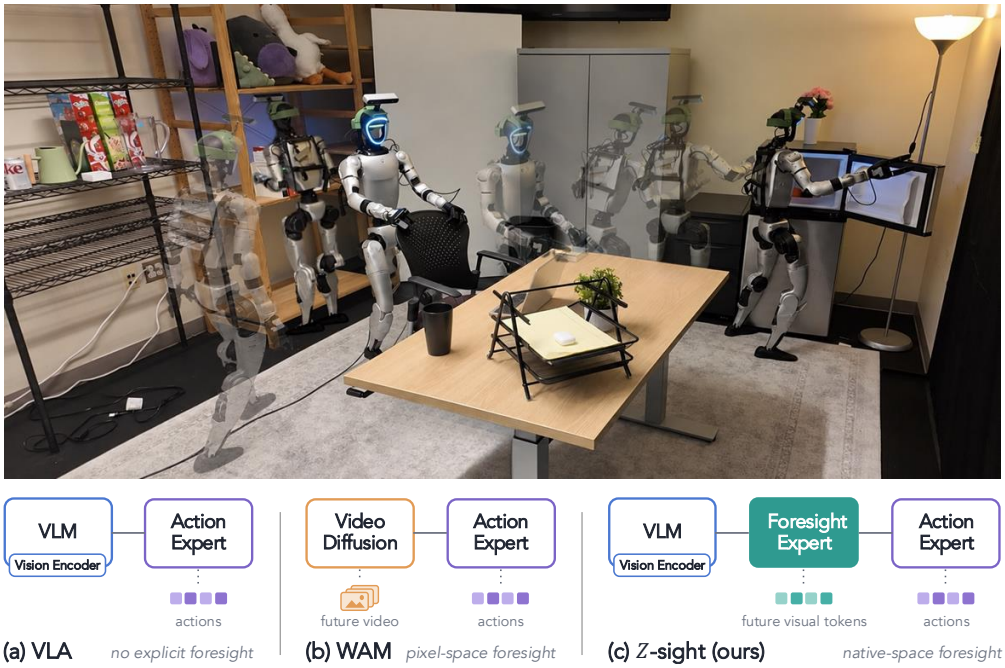

Figure 1: **Z-sight predicts the future in the policy's native representation space.** *Top:* Z-sight performs diverse manipulation tasks in a real-world setting. *Bottom:* comparison of policy designs. (a) Vision-language-action models (VLAs) map the current observation to actions through a VLM and an action expert, without predicting the future. (b) World action models (WAMs) builds on a pretrained video generator and learns to predict future video alongside actions. (c) Z-sight (ours) predicts future observations as tokens of its own VLM's vision encoder, the space in which it perceives the present, and its action expert reads this prediction while generating actions.

ABSTRACT

We introduce Z-sight, a unified robot foundation model that brings vision-language-action modeling and world modeling together through feature-space future prediction. Recent video-based world action models (WAMs) adapt pretrained video generators for robot control, combining visual dynamics modeling with action learning at substantial computational cost. Z-sight brings future prediction into the native visual representation space of a pretrained vision-language model (VLM), allowing the policy to perceive the present and predict the future within a shared visual foundation. We couple a dedicated foresight expert and a continuous action expert to the VLM through a mixture-of-transformers architecture. The foresight expert denoises future spatial visual tokens, while the action expert attends to the prediction stream to generate actions. This design supports joint learning from robot trajectories and human egocentric video without requiring pixel reconstruction or a pretrained video generator. During inference, Z-sight efficiently generates future visual features and continuous actions through joint denoising. Evaluated on a real humanoid robot, Z-sight achieves a 62% average success rate across five manipulation tasks, outperforming VLA and WAM base-

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.

lines by a large margin. Furthermore, it maintains real-time inference speed of 110 ms per chunk without the latency bottleneck of pixel-space generation.

1 INTRODUCTION

Vision-Language-Action (VLA) models transfer visual and linguistic knowledge to robotic control by building policies upon pre-trained vision-language models (Zitkovich et al., 2023; Kim et al., 2025; Black et al., 2025b). Adapting these representations for action prediction provides semantic priors that support instruction following and generalization. Recent architectures extend this paradigm by coupling a VLM backbone with a continuous action expert (Black et al., 2025a; Bjorck et al., 2025), combining multimodal understanding with high-frequency action generation.

A parallel line of work builds robot policies from pre-trained video generation models. These World Action Models (WAMs) (Ye et al., 2026; Yuan et al., 2026) adapt video backbones for control by coupling action learning with future video prediction. This approach leverages spatiotemporal priors from video pre-training and uses future observations to supervise policy learning. These models demonstrate the value of predictive visual representations for robot control. However, they retain the model capacity and computational cost of their underlying video generators. This motivates an alternative: building future prediction directly upon a vision-language foundation.

We introduce Z-sight, a unified model that builds future prediction on a pre-trained vision-language backbone and generates actions from it. Our core insight is to predict future observations in the same visual representation space used by the policy to perceive the present, as shown in Figure 1. A VLM encodes current images into spatial visual tokens. Because future observations can use the same encoding system, these tokens serve as both the perception substrate and prediction target. The action expert then uses these predicted future features. This connects current perception, future prediction, and action generation within a shared visual foundation.

Z-sight is a mixture-of-transformers (MoT) (Liang et al., 2025) with three streams that share attention, as shown in Figure 2. The VLM stream encodes the instruction, the robot state and a history of egocentric frames. We introduce a foresight expert that performs conditional denoising to predict future spatial visual tokens in the VLM vision encoder's feature space. The action expert attends to these predicted features to generate continuous actions. The two losses play different roles, where the action loss specifies what the robot should do, while the future-feature loss specifies how the future scene should change. During inference, the model jointly denoises future features and actions, allowing the predicted latent dynamics to inform action computation.

We train Z-sight on egocentric human video and robot trajectories, then adapt it to a humanoid robot that acts through a learned whole-body controller. Evaluated across five real-world household tasks, Z-sight substantially outperforms state-of-the-art VLA and WAM baselines, particularly on long-horizon tasks involving sequential stages. Furthermore, it maintains high-frequency real-time inference without the latency bottleneck of pixel-space video generation. Controlled ablations confirm that future visual supervision is critical for this performance and that the policy actively relies on its predicted latent dynamics during execution.

Our contributions are threefold:

- We introduce Z-sight, a unified model that predicts future visual tokens in the native representation space of its VLM with a dedicated denoising expert, and integrates these predictive features into continuous action generation.
- We demonstrate that this latent prediction objective enables the policy to extract dense visual supervision from large-scale human egocentric video, bridging the gap between human priors and robotic control without requiring pixel-level reconstruction.
- Through real-world humanoid experiments, we demonstrate that Z-sight outperforms strong VLA and WAM baselines in both task success and inference efficiency.

2 Related Work

Vision-language-action models. Vision-Language-Action (VLA) models adapt pre-trained VLMs to output robot actions, either as discrete tokens (Zitkovich et al., 2023; Kim et al., 2025; Pertsch et al., 2025) or through a continuous action expert trained with flow matching (Black et al., 2025b;a; Bjorck et al., 2025). By leveraging internet-scale semantic knowledge, these models exhibit strong reasoning capabilities and excel at complex instruction following. To preserve the pre-trained representations while learning complex continuous control, Driess et al. (2025) employ knowledge insulation, training the VLM with discrete action tokens while a flow expert learns continuous actions from its detached features. Recent advancements have focused on scaling these architectures with large-scale egocentric human videos, leading to highly capable foundation models for general robot control (Bjorck et al., 2025; Zheng et al., 2026; Wei et al., 2026). However, while these architectures successfully map current visual and language context to actions, their training relies primarily on action labels. This provides a relatively sparse supervisory signal for learning continuous physical dynamics. Z-sight builds upon this foundation by augmenting the learning objective with complementary future visual supervision.

World action models and unified models. In parallel, Video Action and World Action Models (WAMs) learn to predict future visual states and thus acquire a strong sense of physical dynamics. For example, DreamZero (Ye et al., 2026) builds a world action model directly on a pre-trained video diffusion backbone to jointly model video and actions. Fast-WAM (Yuan et al., 2026) utilizes video co-training to learn dynamics but skips future prediction during deployment to accelerate inference. This complementarity has motivated a growing body of work to explore whether a single unified model can combine the semantic understanding of VLAs with the predictive dynamics of WAMs. Motus (Bi et al., 2026) couples a VLM with a video generation expert and an action expert. BagelVLA (Hu et al., 2026) interleaves textual plans, generated keyframes, and actions, using single-step predictive guidance to mitigate inference costs. While these unified models successfully integrate visual prediction and provide dense supervision, they typically rely on video generation. This introduces substantial computational overhead and susceptibility to out-of-distribution visual drift during rollout. Z-sight addresses this by predicting the future directly in the native feature space of the VLM's vision encoder. By avoiding pixel-space generation, Z-sight maintains the efficiency of a standard VLA while benefiting from the dense supervision of world modeling.

Latent world modeling and representation learning. Joint-embedding predictive architectures learn by predicting representations rather than pixels (Baevski et al., 2022; Assran et al., 2023; Bardes et al., 2024), often against an exponential-moving-average (EMA) teacher (Grill et al., 2020). Building on this, latent world models plan with pre-trained visual features (Zhou et al., 2025; Assran et al., 2025). For robotic policies, recent works predict or align future representations inside the policy. FLARE (Zheng et al., 2025) aligns intermediate policy features with future observation embeddings, while UniJEPA (Zhang et al., 2026) maps predictive representations to actions. Other approaches compress views into single tokens (Tang et al., 2026), compare latent alignment objectives (Xia et al., 2026), or distill features from frozen world models (Dao et al., 2026). Together, these works highlight the growing utility of latent spaces for control (Lyu et al., 2026; Nilaksh et al., 2026). Z-sight sits within this family, but distinguishes itself by performing latent future prediction natively within the VLM's own vision encoder space. By coupling a foresight expert and an action expert via a MoT architecture, Z-sight unifies the VLA and the world model into a single, end-to-end trainable system.

3 Methodology

Z-sight is a unified model that learns to predict the future in the representation space in which it perceives the present and generates actions from that prediction. At time t, the policy receives a language instruction ℓ, a proprioceptive state $\boldsymbol{s}_t$, and an observation history $\mathbf{O}_t$ from one or more cameras, and it outputs an action chunk $\boldsymbol{A}_t = (\boldsymbol{a}_t, \ldots, \boldsymbol{a}_{t+H-1})$. During training, the observations that follow within the chunk, $\mathbf{O}_t^+$, serve as an additional target; they are never available at test time. Section 3.1 illustrates the model architecture. Section 3.2 describes the data and the three-stage training recipe. Table A.1 in Appendix A lists the hyperparameters.

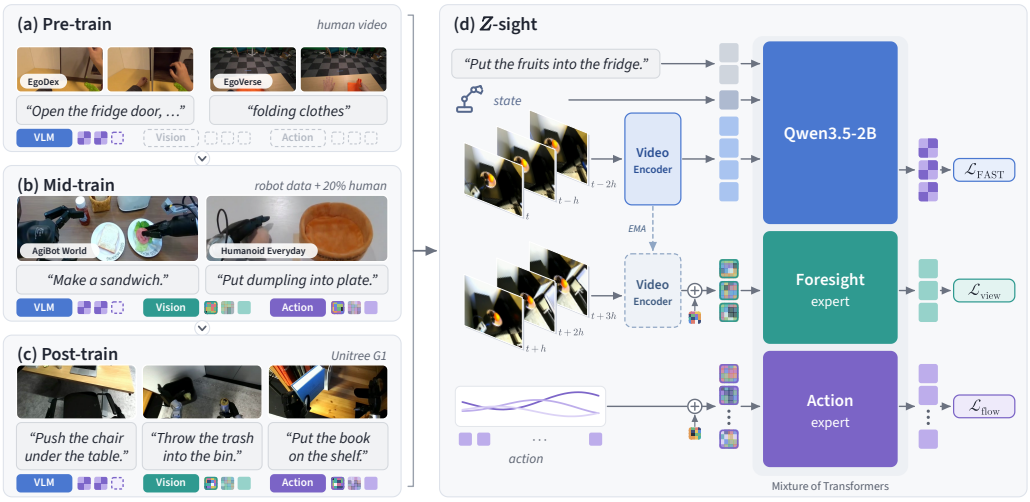

Figure 2: **Overview of Z-sight. (a)–(c) Training stages:** VLM pre-training on human video, mid-training all streams on mixed robot-human data, and post-training on the target humanoid. **(d) Architecture:** The VLM processes observation history, state, and instructions, supervised by discrete FAST tokens ($\mathcal{L}_{\text{FAST}}$). A foresight expert denoises future spatial tokens against an EMA teacher ($\mathcal{L}_{\text{view}}$). Simultaneously, an action expert generates continuous actions via flow matching ($\mathcal{L}_{\text{flow}}$) while attending to the predicted future features. All streams share attention via a MoT.

3.1 Model Architecture

Native representation space. Let E_ϕ denote the vision encoder of the VLM, with parameters ϕ. It maps an image o to P spatial tokens of width D, $E_\phi(o) \in \mathbb{R}^{P\times D}$, which the language model reads as its visual input. We call this token space the policy's *native representation space* and predict the future in it. Instead of generating future pixels, Z-sight predicts the tokens that the encoder would produce for future observations. Such a prediction has the same format as the observation the policy reads, keeps the spatial layout of the scene, and needs neither a pixel decoder nor a video generator.

Video encoder for history and future. A single future frame shows where the scene ends up but not how it gets there, and a single current frame shows a snapshot but not motion. We therefore extend E_ϕ to a video encoder V_ϕ that maps a clip of frames to the token map of its last frame, computed with the earlier frames as context, and use it on both sides:

$$\boldsymbol{z}_t = V_\phi(\mathbf{O}_t), \qquad \boldsymbol{z}_t^+ = V_{\bar{\phi}}(\mathbf{O}_t^+), \qquad \boldsymbol{z}_t, \boldsymbol{z}_t^+ \in \mathbb{R}^{P\times D}, \tag{1}$$

where $\boldsymbol{z}_t$ is the input that the policy perceives, $\boldsymbol{z}_t^+$ is the prediction target, and $\bar{\phi}$ are the parameters of the target encoder defined below. Inspired by MEM (Torne et al., 2026), V_ϕ reuses the weights of E_ϕ and adds no parameters. Most ViT blocks process each frame independently; every fourth block replaces its attention with a factorized space-time attention that reuses the block's query, key and value projections. Let $\boldsymbol{q}_{p,j}$, $\boldsymbol{k}_{p,j}$ and $\boldsymbol{v}_{p,j}$ denote the query, key and value of spatial position p in frame j. The block first mixes the values of each position causally over time and then attends within each frame over the mixed values:

$$\tilde{\boldsymbol{v}}_{p,j} = \sum_{j'\le j} \mathrm{softmax}_{j'}\left(\tfrac{1}{\sqrt{d}}\,(\boldsymbol{R}^{\mathrm{t}}_j \boldsymbol{q}_{p,j})^\top \boldsymbol{R}^{\mathrm{t}}_{j'} \boldsymbol{k}_{p,j'}\right) \boldsymbol{v}_{p,j'}, \tag{2}$$

$$\boldsymbol{y}_{p,j} = \sum_{p'} \mathrm{softmax}_{p'}\left(\tfrac{1}{\sqrt{d}}\,(\boldsymbol{R}^{\mathrm{s}}_p \boldsymbol{q}_{p,j})^\top \boldsymbol{R}^{\mathrm{s}}_{p'} \boldsymbol{k}_{p',j}\right) \tilde{\boldsymbol{v}}_{p',j}, \tag{3}$$

where $\boldsymbol{R}^{\mathrm{t}}_j$ is a one-dimensional rotary embedding of the frame index, $\boldsymbol{R}^{\mathrm{s}}_p$ is the pretrained two-dimensional rotary embedding of the spatial position, d is the head dimension, and the output $\boldsymbol{y}_{p,j}$ passes through the block's residual connection and MLP; we omit the attention heads for brevity. Because the temporal step is causal, the last frame aggregates the whole clip, and keeping only its tokens passes P tokens per camera to the VLM, as for a single image. With several cameras, each camera is encoded separately and its tokens are tagged with a learned view-role embedding.

Preventing collapse and forgetting. Since $\boldsymbol{z}_t$ and $\boldsymbol{z}_t^+$ come from the same encoder, gradients of the prediction loss through the encoder would let the model lower the loss by making its representation easier to predict, which in the limit collapses it; they would also erode the VLM's pretrained knowledge. We block both paths. First, the target encoder is an EMA of the encoder,

$$\bar{\phi} \leftarrow \mu\,\bar{\phi} + (1-\mu)\,\phi, \tag{4}$$

with decay $\mu \in [0, 1)$, and targets are computed without gradient (Assran et al., 2023). Second, following knowledge insulation (Driess et al., 2025), the experts read the VLM's keys and values through a stop-gradient $\mathrm{sg}(\cdot)$, so their losses update only the experts. The VLM is trained instead to predict the FAST tokens (Pertsch et al., 2025) $c_{1:N}$ that encode the action chunk $\boldsymbol{A}_t$:

$$\mathcal{L}_{\text{FAST}} = -\sum_{n=1}^{N} \log p_\theta\big(c_n \mid c_{<n}, \ell, \boldsymbol{s}_t, \boldsymbol{z}_t\big), \tag{5}$$

where p_θ is the next-token distribution of the VLM and θ denotes all model parameters, including ϕ. The encoder is thus trained only by an objective whose optimum does not depend on how predictable its own features are, and action supervision adapts the VLM without the prediction loss eroding its pretrained knowledge.

Foresight expert. Future prediction is carried out by a foresight expert, a transformer stream with its own weights that shares attention with the VLM in a mixture-of-transformers. It predicts $\boldsymbol{z}_t^+$ by flow matching in token space. Given a noise level $\sigma \in [0, 1]$ and $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, it receives $\boldsymbol{f}^\sigma = \sigma\boldsymbol{\xi} + (1-\sigma)\,\boldsymbol{z}_t^+$ and embeds each noisy token together with the corresponding current-frame token,

$$\boldsymbol{h} = \boldsymbol{W}_{\text{in}}\,\boldsymbol{f}^\sigma + \boldsymbol{W}_c\,\mathrm{sg}(\boldsymbol{z}_t) + \boldsymbol{e}_{\text{pos}}, \tag{6}$$

where $\boldsymbol{h}$ is the input to the expert, $\boldsymbol{W}_{\text{in}}$ and $\boldsymbol{W}_c$ are learned projections to the expert width, and $\boldsymbol{e}_{\text{pos}}$ is a learned position embedding. Because $\boldsymbol{W}_c$ is initialized to zero, the expert starts from the present and learns how it changes rather than generating the future from scratch. The noise level conditions every layer through adaptive RMSNorm. Following the x-prediction parameterization (Li & He, 2026), the expert outputs a clean estimate $\hat{\boldsymbol{z}}_t^+$ and is trained with

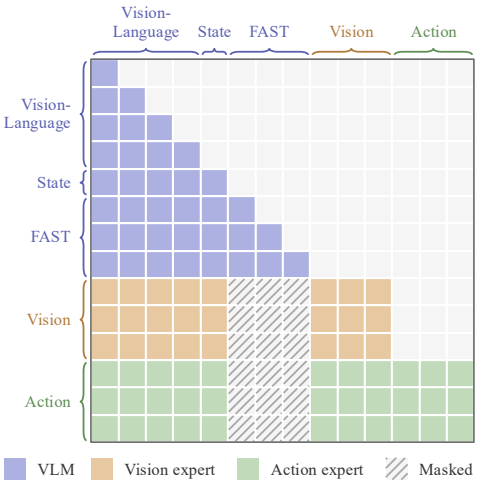

Figure 3: **Attention mask of Z-sight.** Rows are queries and columns are keys.

$$\mathcal{L}_{\text{view}} = \mathbb{E}\Big[\big\|(\boldsymbol{z}_t^+ - \hat{\boldsymbol{z}}_t^+) \,/\, \max(\sigma, \sigma_{\min})\big\|^2\Big], \tag{7}$$

which equals the error of the implied flow velocity; the small constant $\sigma_{\min}$ bounds the weight at low noise levels.

Action expert. A second expert generates the action chunk with conditional flow matching (Lipman et al., 2023; Black et al., 2025b). Given $\tau \in [0, 1]$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, it receives $\boldsymbol{A}^\tau = \tau\boldsymbol{\epsilon} + (1-\tau)\boldsymbol{A}_t$, is conditioned on τ through adaptive RMSNorm, and predicts a velocity $\boldsymbol{u}_\theta$:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}\big[\|\boldsymbol{u}_\theta(\boldsymbol{A}^\tau, \tau) - (\boldsymbol{\epsilon} - \boldsymbol{A}_t)\|^2\big]. \tag{8}$$

During mid-training, the full objective is $\mathcal{L} = \mathcal{L}_{\text{FAST}} + \mathcal{L}_{\text{flow}} + \lambda_v \mathcal{L}_{\text{view}}$, where λ_v weights the future-prediction loss.

Attention pattern. As shown in Figure 3, the streams are ordered as [`VLM` | `FAST` | `foresight` | `action`]. The VLM tokens attend causally, as in pretraining, and the FAST tokens attend causally to the VLM tokens and to earlier FAST tokens. Since the FAST tokens encode the ground-truth action chunk, both experts are blocked from attending to them. The foresight stream attends to the VLM tokens and bidirectionally to itself, but not to the action stream, so Z-sight predicts the future from the observation alone and does not simulate the consequences of candidate actions. The

action stream attends to the VLM tokens, to the foresight stream and to itself. It therefore reads the predicted future directly, in the same token space as the current observation.

Joint denoising at inference. At test time the FAST tokens are not generated. The VLM is run once per decision and its keys and values are cached. Starting from $\boldsymbol{f}^1 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and $\boldsymbol{A}^1 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, both streams are integrated with a shared schedule of K Euler steps, $\Delta = -1/K$:

$$\boldsymbol{A}^{\tau+\Delta} = \boldsymbol{A}^{\tau} + \Delta\, \boldsymbol{u}_\theta, \qquad \boldsymbol{f}^{\tau+\Delta} = \boldsymbol{f}^{\tau} + \Delta\, \frac{\boldsymbol{f}^{\tau} - \hat{\boldsymbol{z}}_t^{+}}{\max(\tau, \sigma_{\min})}, \tag{9}$$

where both updates come from one forward pass of the two experts, so at every step the action expert reads the prediction as it forms.

3.2 Training Recipe

We train Z-sight in three stages: pre-training the VLM on egocentric human video, mid-training the full model on cross-embodiment robot data with human-data replay, and post-training on the target embodiment. Figure 4 shows the composition of the training data, and Table B.1 in Appendix B lists the statistics of each source. Human hands and robot end-effectors share one state and action space during pre-train and mid-train. Humans and humanoids use a 48-D task-space hand representation (Wei et al., 2026), and other robots use end-effector poses with gripper or hand commands. Actions are relative to the current state within each chunk, and all embodiments are padded to a shared dimension with per-dimension validity masks that remove padded, missing or untracked channels from the losses.

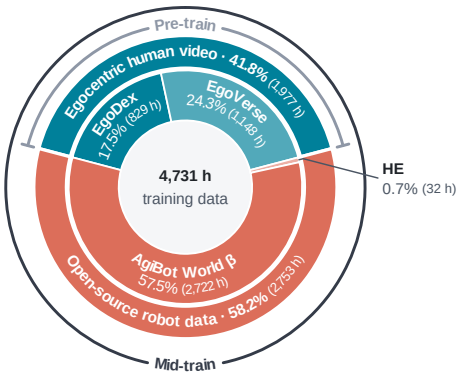

Figure 4: **Training data.** Pre-training uses 1,977 h of egocentric human video (inner arc); mid-training uses all 4,731 h (outer circle): open-source robot data with 20% human-data replay.

Pre-training on egocentric human video. We use EgoDex (Hoque et al., 2026) and EgoVerse (Punamiya et al., 2026), about 2,000 hours of egocentric manipulation video with tracked hands. Hand trajectories are expressed in the camera frame of the current observation, such that actions are not confounded by head motion. We mask joints with low tracking confidence, and we smooth the hand trajectories with a validity-weighted temporal Gaussian filter. Vision-based hand estimates jitter from frame to frame, and this high-frequency noise inflates the FAST encoding of an action chunk. Smoothing shortens the FAST encoding of a chunk by 13.1% on EgoDex and 20.9% on EgoVerse on average, while displacing the hand keypoints by 1.3 mm and 3.0 mm at the median. Appendix C describes the filter in detail. In this stage, we train only the VLM, which predicts FAST action tokens autoregressively with a tokenizer fitted to the 48-D representation.

Mid-training on cross-embodiment robot data. We add the vision and action experts and train the full objective of Section 3.1 under knowledge insulation. The data comprise about 2,750 hours from AgiBot World (AgiBot-World-Contributors et al., 2025) and Humanoid Everyday (Zhao et al., 2026). For Humanoid Everyday, which records only joint angles, we derive wrist poses and fingertip positions in the robot base frame via forward kinematics (Wei et al., 2026). Fingertips absent from the kinematic model are masked. We remove idle segments from the robot data. Human data are replayed at 20% of the sampling mass, which preserves what pre-training learned and keeps diverse visual dynamics in the prediction target.

Post-training on the target embodiment. We fine-tune on demonstrations of the target robot with $\mathcal{L}_{\text{flow}} + \lambda_v \mathcal{L}_{\text{view}}$, replaying the mid-training mixture at 20% of the sampling mass to retain what mid-training learned. The VLM is frozen except for the state projection, which must learn the new state space, and knowledge insulation is disabled so that gradients reach it through the frozen backbone. With the encoder frozen, the prediction loss cannot reshape its own target, so the collapse shortcut does not arise. Data, sampling weights and hyperparameters are listed in Appendix A.

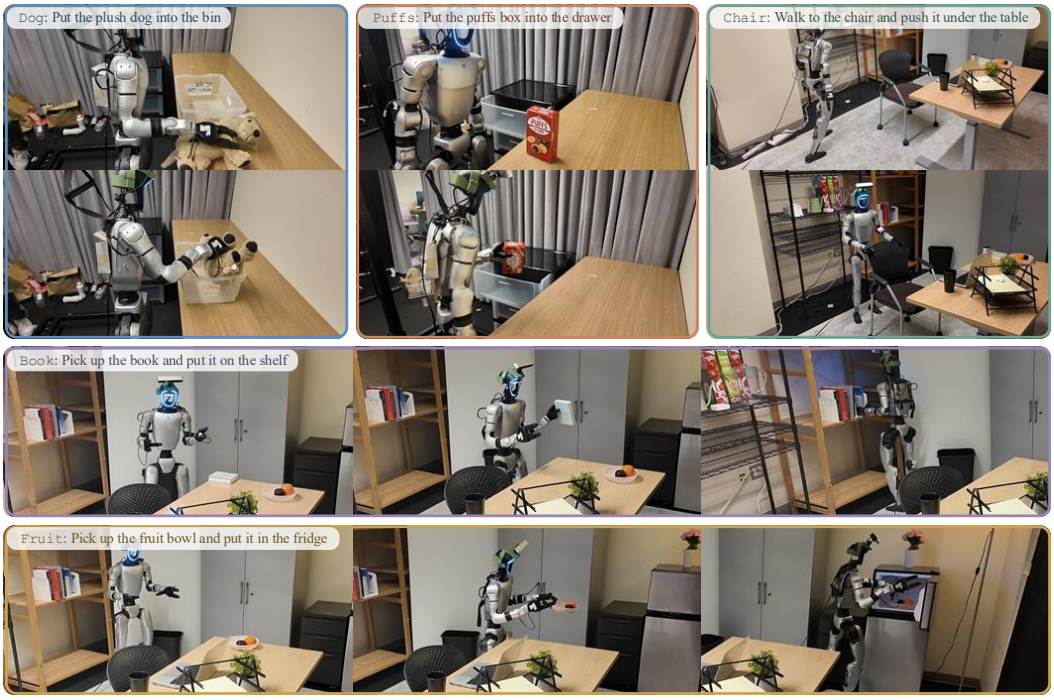

Figure 5: **Real-world tasks.** The five evaluation tasks on the Unitree G1, with the instruction for each task overlaid. `Dog` and `Puffs` are short pick-and-place tasks; `Chair`, `Book` and `Fruit` consist of several stages that must be completed in order. All tasks require the humanoid to coordinate its body, arms and dexterous hands.

4 Experiments

Z-sight predicts the future in the same visual token space in which its VLM perceives the present. Our experiments evaluate the practical benefits of the native-space foresight for robotic control[1]. Specifically, we aim to answer four key questions: (1) Does native-space foresight improve real-world manipulation over standard VLAs and video-based WAMs? (Section 4.2) (2) Does the policy actively rely on its own predicted future features during execution? (Section 4.3) (3) Is the future-feature objective the true source of the performance gain? (Section 4.4) (4) Can Z-sight maintain the low inference latency of a VLA while retaining test-time foresight? (Section 4.5)

4.1 Evaluation Setup

Robot platforms. We use the Unitree G1 humanoid equipped with two 7-DoF Dex3-1 dexterous hands for whole-body manipulation tasks. The robot perceives the scene through a single egocentric head camera. Its default Intel RealSense D435i has a narrow field of view, which suffices for tabletop tasks but not for tasks that involve locomotion, where the robot must see the surrounding scene as it moves. We therefore also mount a ZED 2i camera with a wider field of view on the head. The three tasks that involve locomotion, `Chair`, `Book` and `Fruit`, are recorded and evaluated with the ZED camera, and `Dog` and `Puffs` with the RealSense. Each task uses the same camera for data collection and evaluation. The policy controls the robot through SONIC (Luo et al., 2026), a learned whole-body controller. Each action is a latent motion token for SONIC together with joint targets for both hands, and SONIC converts the token into joint commands at 50 Hz. Appendix D describes the teleoperation and deployment pipeline.

Tasks. As shown in Figure 5, we evaluate on five household tasks that require the humanoid to coordinate its body, arms and dexterous hands; Appendix E describes each task in detail. The tasks

[1]Demo videos demonstrating real-world execution are available at `https://z-sight.pages.dev/`.

Table 1: **Real-world results on the Unitree G1.** Success rate (SR, %) and progress score (Prog., %) over 10 trials per task. The last two columns average over the five tasks. **Bold**: best per column.

		Dog		Puffs		Chair		Book		Fruit		Average	
Category	Method	SR	Prog.	SR	Prog.	SR	Prog.	SR	Prog.	SR	Prog.	SR	Prog.
WAM	Fast-WAM	60	65.0	**60**	**80.0**	60	60.0	50	65.0	0	15.0	46	57.0
VLA	$\pi_{0.5}$	70	70.0	50	65.0	40	40.0	0	5.0	0	7.5	32	37.5
	GR00T N1.7	**90**	90.0	30	45.0	30	40.0	10	35.0	0	10.0	32	44.0
Ours	Z-sight	**90**	**95.0**	**60**	70.0	**90**	**90.0**	**60**	**75.0**	**10**	**17.5**	**62**	**69.5**

differ in length. `Dog` and `Puffs` are short pick-and-place tasks, whereas `Chair`, `Book` and `Fruit` consist of several stages that must be completed in order; their demonstrations last 39, 51 and 85 s on average, compared with 16 and 28 s for the short tasks. Every method is trained as multi-task policies on the teleoperated G1 demonstrations, 30–60 per task (219 in total), and is evaluated for 10 trials per task from the same set of initial configurations. We report the success rate and a progress score that gives equal partial credit to each stage of a task.

Baselines. We choose baselines that represent the two designs Z-sight combines. On the VLA side, $\pi_{0.5}$ (Black et al., 2025a) and GR00T N1.7 (Bjorck et al., 2025) couple a VLM with a flow-matching action expert. They both map the present observation to actions without predicting the future. On the WAM side, Fast-WAM (Yuan et al., 2026) learns dynamics through video co-training and skips future generation at test time. Z-sight shares its backbone design with the first group and its future supervision with the second, so this comparison measures what foresight in the VLM's own feature space adds over each. We detail the baseline fine-tuning details in Appendix F.

4.2 Z-SIGHT VS. VLA AND WAM POLICIES

As shown in Table 1, Z-sight achieves the highest average success rate and progress score: 62% success and 69.5% progress, compared with 46% and 57.0% for Fast-WAM and 32% success for both VLAs. It matches or exceeds every baseline in success rate on every task. Over the 50 trials per method, the improvement over each VLA is statistically significant (31 vs. 16 successes, $p = 0.005$, two-sided Fisher exact test), and Z-sight also records more successes than Fast-WAM (31 vs. 23), matching or exceeding it on every task.

This performance gap is most pronounced on long, multi-stage tasks. On short tasks (`Dog`, `Puffs`), all methods perform comparably. However, on the three longer tasks, standard VLAs typically stall after the first stage, averaging only 13% success. Future-predicting methods successfully chain multiple stages, with Z-sight achieving 53% average success compared to 37% for Fast-WAM. Notably, on the most complex four-stage `Fruit` task, Z-sight is the only model to achieve full success and records the highest overall progress.

4.3 THE POLICY ACTS ON ITS PREDICTED FUTURE

Table 2: Empirical study of the foresight stream.

Variant	SR	Prog.
Z-sight	**90**	**95.0**
Test-time intervention		
Noise ($\sigma = 1$)	60	60.0
Copy of current frame	0	0.0
Post-training variant		
No view loss ($\lambda_v = 0$)	30	30.0
No foresight branch	40	40.0

If the action expert acts on the predicted future, changing that prediction at test time should change the behavior even though the weights and the current observation stay the same. We run these ablations on `Dog` with 10 trials each and intervene on the foresight stream during joint denoising in two ways. **Noise** replaces the stream with fresh Gaussian noise at every step and labels it with noise level $\sigma = 1$, so the action expert is told that no prediction is available. **Copy** sets the stream at every step to $\tau\boldsymbol{\xi} + (1-\tau)\,\boldsymbol{z}_t$ with a fixed noise sample $\boldsymbol{\xi}$ and the running label τ, *i.e.*, the trajectory the stream would follow if the foresight expert predicted at every step that nothing will change, $\hat{\boldsymbol{z}}_t^+ = \boldsymbol{z}_t$.

Table 2 shows that the policy's actions are highly sensitive to the content of its prediction. When the prediction is forced to indicate that the scene will not change (**Copy**), the policy fails in every

trial (0% vs. 90%). The robot still reaches toward the object based on current observations, but fails to execute actions that actively change the scene (*e.g.*, closing its hand or lifting the object). Conversely, replacing the prediction with pure noise is significantly less harmful (60% success). Because the action and foresight branches are sampled independently during training, the action expert has learned to be robust to noisy visual features and can fall back on the current observation. However, a systematically incorrect prediction (**Copy**) actively misleads the future trajectory. This confirms that the action expert does not merely benefit from the presence of a trained vision stream, but actively relies on what the stream predicts during deployment.

4.4 Future supervision drives the gain

From the same mid-trained checkpoint, we post-train two variants with the same data and schedule. **No view loss** sets $\lambda_v = 0$ and keeps the architecture unchanged, so the foresight expert is shaped only by the action loss. **No foresight branch** removes the foresight expert and its stream, so the action expert attends only to the VLM, effectively reducing the architecture to a standard VLA variant. Without future supervision, success drops from 90% to 30% ($p = 0.02$), and without the foresight branch it drops to 40%, below both VLAs on this task. We attribute the latter to our post-training recipe, which keeps the VLM frozen and was designed around the foresight branch: without the branch, the action expert alone must adapt the policy to the new embodiment. The branch therefore helps only when it is trained to predict the future: an additional stream that is shaped by the action loss alone is no better than none. The gain comes from the future-prediction objective rather than from the added capacity of the foresight expert.

4.5 Foresight at VLA cost

We measure the time to produce one action chunk on the RTX 4090 used for deployment. Each policy is loaded in process in its deployment configuration and timed with CUDA synchronization on a 480×640 observation, over 100 calls after 100 warm-up calls. This excludes network transport and isolates model compute.

Table 3: Inference latency per action chunk on one RTX 4090 (ms).

Method	Median	Mean $\pm$ std	Chunk
$\pi_{0.5}$	120.5	121.0 ± 1.3	30
GR00T N1.7	60.0	60.3 ± 1.4	30
Fast-WAM	155.9	156.6 ± 2.1	32
Z-sight	109.5	109.6 ± 0.2	30

As shown in Table 3, Z-sight computes an action chunk in 109.5 ms. The foresight stream accounts for 45 ms of this time, measured by disabling it at test time. Even so, Z-sight computes a chunk faster than $\pi_{0.5}$ (120.5 ms) and Fast-WAM (155.9 ms), although Fast-WAM skips future generation at test time and Z-sight does not. GR00T N1.7 is the fastest policy at 60 ms but does not predict the future. At our 10 Hz control rate, a 30-step chunk spans 3 s, so the 110 ms needed to compute it is a small fraction of its execution time.

5 Conclusion

We presented Z-sight, a unified model that predicts the future in the native visual representation space of its VLM and generates continuous actions while reading that prediction. By predicting future states directly where the policy perceives the present, Z-sight allows perception, prediction, and action to share a unified foundation without requiring a computationally heavy pixel-level video generator. Our real-world humanoid experiments confirm that this native latent world modeling provides the dense supervision necessary for complex, multi-stage manipulation, while maintaining the high-frequency inference speeds required for dynamic control. Ultimately, Z-sight suggests that an effective way to equip policies with an understanding of physical dynamics is to predict the future exactly where the policy looks.

Limitations and future work. We evaluate Z-sight on one humanoid platform. Since pre-training and mid-training already span several embodiments, applying it to other robots, such as single-arm manipulators and mobile platforms, is a natural next step. The foresight stream currently predicts the future from the observation alone. Conditioning it on candidate actions would let Z-sight compare the outcomes of different actions and open the door to test-time planning in its own representation space. The predicted tokens are not decoded into images, and attaching a lightweight decoder would make the predictions easier to inspect.

AI USE STATEMENT

We used generative AI tools (large language models), in two ways. For software, it assisted with maintaining the codebase and with writing and revising experiment and support scripts. The core implementation and all AI-assisted code were checked through AI-assisted code review and manual verification, and the authors reviewed and approved the final code. For the paper, it assisted with drafting and revising the text, organizing sections, searching for related work, and retrieving bibliographic metadata; every citation was verified against its source. The research direction, experiments and results are the authors' own. The authors reviewed all AI-assisted content and take full responsibility for this work.

ETHICS STATEMENT

Our human training data come from publicly released egocentric datasets (EgoDex and EgoVerse), used under their licenses. The robot demonstrations were collected by the authors through teleoperation in a laboratory setting, and all real-world experiments were supervised by an operator with an emergency stop. Humanoid robots operating near people pose safety risks, and learned policies can fail in unexpected ways; Z-sight is a research prototype and is not intended for deployment without additional safety measures.

REPRODUCIBILITY STATEMENT

Section 3 describes the architecture, objectives and training stages, and Appendix A lists all hyperparameters, the data mixture and the per-stage settings. Appendix B describes data processing, Appendix D the deployment pipeline, Appendix E the real-world tasks and scoring, and Appendix F the baseline fine-tuning. All pretraining and mid-training data are publicly available. We will release all code, model checkpoints and our G1 demonstration data upon acceptance.

REFERENCES

AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, Shu Jiang, et al. AgiBot World Colosseo: A Large-scale Manipulation Platform for Scalable and Intelligent Embodied Systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba, Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning. *arXiv preprint arXiv:2506.09985*, 2025.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 162, 2022.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting Feature Prediction for Learning Visual Representations from Video. *Transactions on Machine Learning Research*, 2024.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A Unified Latent Action World Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

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. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael 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 Conference on Robot Learning (CoRL)*, 2025a.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of the Robotics: Science and Systems Conference (RSS)*, 2025b.

Trung Dao, Sankalp Yamsani, Jaden Park, Joohyung Kim, and Yong Jae Lee. Think Like a World Model, Act Like a VLA: Distilling World-Model Representations into Compact Robot Policies. *arXiv preprint arXiv:2609.24682*, 2026.

Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z. Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, and Sergey Levine. Knowledge Insulating Vision-Language-Action Models: Train Fast, Run Fast, Generalize Better. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

Ryan Hoque, Peide Huang, David Yoon, Mouli sivapurapu, and Jian Zhang. EgoDex: Learning Dexterous Manipulation from Large-Scale Egocentric Video. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Yucheng Hu, Jianke Zhang, Yuanfei Luo, Yanjiang Guo, Xiaoyu Chen, Xinshu Sun, Kun Feng, Qingzhou Lu, Sheng Chen, Yangang Zhang, Wei Li, and Jianyu Chen. BagelVLA: Enhancing Long-Horizon Manipulation via Interleaved Vision-Language-Action Generation. *arXiv preprint arXiv:2602.09849*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag 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 Conference on Robot Learning (CoRL)*, volume 270, 2025.

Tianhong Li and Kaiming He. Back to Basics: Let Denoising Generative Models Denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Weixin Liang, Lili Yu, Liang Luo, Srini Iyer, Ning Dong, Chunting Zhou, Gargi Ghosh, Mike Lewis, Wen-tau Yih, Luke Zettlemoyer, and Xi Victoria Lin. Mixture-of-Transformers: A Sparse and Scalable Architecture for Multi-Modal Foundation Models. *Transactions on Machine Learning Research*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

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 "Jim" Fan, and Yuke Zhu. SONIC: Supersizing motion tracking for natural humanoid whole-body control. *Science Robotics*, 11 (117), 2026.

Jiangran Lyu, Kai Liu, Xuheng Zhang, Haoran Liao, Yusen Feng, Wenxuan Zhu, Tingrui Shen, Jiayi Chen, Jiazhao Zhang, Yifei Dong, Wenbo Cui, Senmao Qi, Shuo Wang, Yixin Zheng, Mi Yan, Xuesong Shi, Haoran Li, Dongbin Zhao, Ming-Yu Liu, Zhizheng Zhang, Li Yi, Yizhou Wang, and He Wang. LDA-1B: Scaling Latent Dynamics Action Model via Universal Embodied Data Ingestion. In *Proceedings of Robotics: Science and Systems*, 2026.

Nilaksh, Saurav Jha, Artem Zholus, and Sarath Chandar. Reconstruction or Semantics? What Makes a Latent Space Useful for Robotic World Models. *arXiv preprint arXiv:2605.06388*, 2026.

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.

Ryan Punamiya, Simar Kareer, Zeyi Liu, Josh Citron, Ri-Zhao Qiu, Xiongyi Cai, Alexey Gavryushin, Jiaqi Chen, Davide Liconti, Lawrence Y. Zhu, Patcharapong Aphiwetsa, Baoyu Li, et al. EgoVerse: An Egocentric Human Dataset for Robot Learning from Around the World. *arXiv preprint arXiv:2604.07607*, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Zuojin Tang, Shengchao Yuan, Xiaoxin Bai, Zhiyuan Jing, De Ma, Gang Pan, and Bin Liu. One Token Per Frame: Reconsidering Visual Bandwidth in World Models for VLA Policy. *arXiv preprint arXiv:2605.07931*, 2026.

Marcel Torne, Karl Pertsch, Homer Walke, Kyle Vedder, Suraj Nair, Brian Ichter, Allen Z. Ren, Haohuan Wang, Jiaming Tang, Kyle Stachowicz, Karan Dhabalia, Michael Equi, et al. MEM: Multi-Scale Embodied Memory for Vision Language Action Models. *arXiv preprint arXiv:2603.03596*, 2026.

Songlin Wei, Hongyi Jing, Boqian Li, Zhenyu Zhao, Jiageng Mao, Zhenhao Ni, Sicheng He, Sheng Zang, Xiawei Liu, Kaidi Kang, Jie Liu, Weiduo Yuan, Marco Pavone, Di Huang, and Yue Wang. Ψ_0: An Open Foundation Model Towards Universal Humanoid Loco-Manipulation. In *Proceedings of Robotics: Science and Systems*, 2026.

Guoyang Xia, Fengfa Li, Hongjin Ji, Lei Ren, Fangxiang Feng, Kun Zhan, and Yan Xie. VLAFlow: A Unified Training Framework for Vision-Language-Action Models via Co-training and Future Latent Alignment. *arXiv preprint arXiv:2607.01586*, 2026. URL `https://arxiv.org/abs/2607.01586`.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Naveed Malik, Kyungmin Lee, William Liang, Nadun Ranawaka Arachchige, 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, Danfei Xu, Yilun Du, Ryan Julian, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi Fan, and Joel Jang. World Action Models are Zero-shot Policies. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2026.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do World Action Models Need Test-time Future Imagination? *arXiv preprint arXiv:2603.16666*, 2026.

Jianke Zhang, Yucheng Hu, Yanjiang Guo, Xiaoyu Chen, Yichen Liu, Wenna Chen, Chaochao Lu, and Jianyu Chen. UniJEPA: Enhancing Robot Policy via Unified Continuous and Discrete Representation Learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026.

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 *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, Avnish Narayan, You Liang Tan, Guanzhi Wang, Qi Wang, Jiannan Xiang, Yinzhen Xu, Seonghyeon Ye, Jan Kautz, Furong Huang, Yuke Zhu, and Linxi Fan. FLARE: Robot Learning with Implicit World Modeling. In *Proceedings of The Conference on Robot Learning (CoRL)*, 2025.

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*, 2026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World Models on Pre-trained Visual Features enable Zero-shot Planning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

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 Conference on Robot Learning (CoRL)*, volume 229, 2023.

APPENDIX CONTENTS

A Implementation Details **15**

B Data Processing **15**

C Hand Trajectory Smoothing **16**

D Deployment **16**

E Real-World Tasks and Scoring **17**

F Baselines **17**

A Implementation Details

Table A.1 lists the values used for the model reported in the main paper.

Table A.1: Implementation details of Z-sight.

	Item	Value
Model	VLM backbone	Qwen3.5-2B (Qwen Team, 2026)
	Vision / action expert width	1024 / 1024
	Parameters (VLM / foresight expert / action expert)	2.22 B / 0.48 B / 0.48 B (3.18 B total)
	Image resolution (G1 egocentric camera)	224×288, $P = 63$ tokens per frame
	Observation history / future clip	3 frames each, 1 s apart
	Space-time attention	every 4th ViT block, causal temporal RoPE
	Action horizon H, data rate	30 steps (3 s), 10 Hz
Objectives	Future-feature loss weight λ_v	0.1
	Vision noise level σ	mid-training: $1 - \mathrm{sigmoid}(0.8u - 0.8)$, $u \sim \mathcal{N}(0, 1)$; post-training: $\sigma = \tau$
	$\sigma_{\min}$	0.05
	Action flow time τ	$\mathrm{Beta}(1.5, 1)$, rescaled to $[0.001, 1]$
	EMA decay μ	0.99
	Euler steps K at inference	10
Data	Human-trajectory smoothing	Gaussian, $\sigma = 2$ steps
	Idle filter	activity $< 10^{-3}$; idle runs ≥ 0.5 s; active runs ≥ 1.0 s
Pre-training	Data	EgoDex, EgoVerse
	FAST tokenizer	fitted to the 48-D space, 2,048 tokens
	Steps, global batch	200k, 1,024
	Hardware	32 H200 GPUs (4 nodes)
Mid-training	Data	AgiBot World, Humanoid Everyday, 20% human
	Steps, global batch	300k, 1,024
	Hardware	32 H200 GPUs (4 nodes)
	Learning rate	5×10^{-5} constant; vision encoder $0.1\times$
	Trainable	all (VLM via FAST loss only)
Post-training	Data	G1 demonstrations, 20% mid-training replay
	Steps, global batch	20k, 128
	Learning rate	cosine, $10^{-4} \to 10^{-5}$, 500 warm-up steps
	Trainable	experts, state projection
	Augmentation	random crop (scale 0.9–1), $\pm 5^\circ$ rotation, color jitter; state dropout 0.1

B Data Processing

Data standardization. We convert every source into a common episodic format that stores the raw signals rather than a fixed action space: camera streams with their intrinsics and per-frame extrinsics, proprioception, hand keypoints, and wrist or end-effector poses, with poses expressed in the camera frame wherever camera poses are available. Signals that a source does not provide, or that its tracker loses, are stored with an explicit ignore value, and normalization statistics are computed exactly over each dataset with these entries excluded. The mapping from raw signals to the state and action representation below happens at loading time, so the representation can change without reconverting the data, and ignored entries become the per-dimension validity masks used in the losses.

Table B.1: Training data. Hours are computed from frame counts at the native 30 Hz recording rate (10 Hz for our G1 demonstrations). Human data use the 48-D hand representation in the camera frame; Humanoid Everyday uses it in the robot base frame.

Source	Type	Embodiment	Cameras	Hours	Stages
EgoDex (Hoque et al., 2026)	human	hands	head	829.4	1, 2
EgoVerse-Mecka (Punamiya et al., 2026)	human	hands	head	826.2	1, 2
EgoVerse-Scale (Punamiya et al., 2026)	human	hands	head	216.6	1, 2
EgoVerse-Aria (Punamiya et al., 2026)	human	hands	head	104.9	1, 2
AgiBot World, gripper (AgiBot-World-Contributors et al., 2025)	robot	dual-arm, grippers	head + 2 wrist	2,615.7	2
AgiBot World, dexterous (AgiBot-World-Contributors et al., 2025)	robot	dual-arm, dex. hands	head	105.9	2
Humanoid Everyday G1 (Zhao et al., 2026)	robot	humanoid	head	16.5	2
Humanoid Everyday H1 (Zhao et al., 2026)	robot	humanoid	head	15.3	2
G1 demonstrations (ours)	robot	humanoid + SONIC	head	5.2	3

C HAND TRAJECTORY SMOOTHING

Hand poses in EgoDex and EgoVerse are estimated per frame from video, and the estimates jitter by a few millimeters between frames. Because FAST encodes an action chunk through its discrete cosine transform coefficients, this high-frequency jitter produces many small but non-zero high-frequency coefficients and therefore longer token sequences. We smooth each 48-D human trajectory along time with a Gaussian kernel of standard deviation $\sigma_s = 2$ steps. To keep invalid entries from contaminating their neighbors, the filter is normalized by the smoothed validity mask,

$$\tilde{x}_t = \frac{\sum_j g_{\sigma_s}(t-j)\, m_j x_j}{\sum_j g_{\sigma_s}(t-j)\, m_j}, \tag{10}$$

where $m_j \in \{0, 1\}$ marks valid entries, and entries that are invalid remain masked in the loss. We load $3\sigma_s$ additional frames on each side of the training window so that every smoothed value lies in the interior of the filter. The current state is taken from the raw, unsmoothed trajectory. Robot trajectories are computed from commanded joint angles and are not smoothed.

Table C.1 measures the effect of smoothing on 1,024 paired 30-step action chunks from 45 EgoDex episodes and 3,072 from 24 EgoVerse episodes (1,024 from each of its three subsets); each chunk is encoded with and without smoothing. With our fitted tokenizer, smoothing shortens the FAST token sequence by 13.1% on EgoDex and 20.9% on EgoVerse on average, and by 18.9% and about 32% at the 99th percentile. The released FAST tokenizer shows a similar reduction (13.1% and 22.9% on average), so the saving comes from the smoothing rather than from the refitted vocabulary; our tokenizer is separately shorter in absolute terms, for instance 160 versus 169 tokens for an unsmoothed EgoVerse chunk. Smoothing displaces the hand keypoints by 1.3 mm (EgoDex) and 3.0 mm (EgoVerse) at the median and 5.4 mm and 8.0 mm on average. The tail is larger, 59 mm and 74 mm at the 99th percentile, where the kernel flattens the peak of a fast reaching motion.

Table C.1: Effect of temporal smoothing ($\sigma_s = 0 \to 2$) on human action chunks. Token counts are means per chunk. Means are exact; values marked ≈ are EgoVerse quantiles averaged over its three subsets.

		EgoDex	EgoVerse
Fitted tokenizer	tokens per chunk	120 → 105	160 → 126
	reduction, mean	13.1%	20.9%
	reduction, 99th pct.	18.9%	≈32.0%
Released tokenizer	tokens per chunk	124 → 107	169 → 130
	reduction, mean	13.1%	22.9%
	reduction, 99th pct.	18.4%	≈31.6%
Keypoint displacement (mm)	median	1.3	≈3.0
	mean	5.4	8.0
	99th pct.	58.8	≈73.8

D DEPLOYMENT

We deploy Z-sight on a Unitree G1 humanoid with two dexterous hands and a single egocentric head camera. Rather than commanding joint positions for the whole body, the policy acts through SONIC (Luo et al., 2026), a learned whole-body controller whose motion-tracking policy is driven by a latent motion token. Each action consists of a SONIC motion token together with joint targets for both hands, and the state consists of the body joint positions together with the commanded hand targets. Demonstrations are collected in the same interface: a teleoperator wearing a VR headset drives SONIC with tracked upper-body poses, hand-pose commands and locomotion commands, and we record the resulting motion tokens and hand commands as actions, synchronized with the robot's camera and proprioception. At test time, the SONIC decoder converts each predicted token into whole-body joint commands in a 50 Hz closed control loop, so balance and lower-body coordination are handled by the controller while Z-sight decides what the body should do. The policy runs on a single NVIDIA RTX 4090 GPU and streams actions to the controller at the 10 Hz data rate.

E Real-World Tasks and Scoring

Table E.1 summarizes the five tasks. `Dog` and `Puffs` are recorded with the robot's default head camera (RealSense), and `Chair`, `Book` and `Fruit` with a wide field-of-view head camera (ZED). The training set also contains 236 demonstrations of five further tasks that are not evaluated in this paper; all methods are trained on the same data, with 10% of the episodes held out for validation. The progress score of a trial is the fraction of stages completed in order; a trial is a success if all stages are completed. Table E.2 breaks down the four stages of `Fruit`: every method picks up the plate in several trials, but only Z-sight gets past opening the door, placing the plate inside and closing the door.

Table E.1: Real-world tasks. Duration is the mean demonstration length.

Task	Demos	Duration	Camera	Stages (credit)
`Dog`	60	16 s	head (RealSense)	place the toy dog on the box (0.5); the dog is contained within the box (0.5)
`Puffs`	49	28 s	head (RealSense)	pick up the box (0.5); turn around and place it in the drawer (0.5)
`Chair`	46	39 s	head (ZED)	push the chair in (0.5); chair at the target location and robot back in a standing pose (0.5)
`Book`	34	51 s	head (ZED)	pick up and hold the book (0.5); place it on the shelf (0.5)
`Fruit`	30	85 s	head (ZED)	pick up the plate (0.25); open the door (0.25); place the plate inside (0.25); close the door (0.25)

Table E.2: Stage-wise results on `Fruit`: number of trials (out of 10) in which each stage was completed. Stages must be completed in order.

Stage	Fast-WAM	$\pi_{0.5}$	GR00T N1.7	Z-sight
1. Pick up the plate	**5**	3	3	3
2. Open the door	1	0	1	**2**
3. Place the plate inside	0	0	0	**1**
4. Close the door	0	0	0	**1**
Progress (%)	15.0	7.5	10.0	**17.5**

F Baselines

All methods are trained on the same G1 demonstrations, with the same 78-D action (SONIC motion token and hand joints), 43-D state, 10% episode holdout and per-task sampling weights proportional to the square root of the number of frames, and all are evaluated with the same deployment stack. Each baseline otherwise follows its official fine-tuning recipe.

$\pi_{0.5}$. We fine-tune the released $\pi_{0.5}$ base checkpoint with every parameter trainable and the official OpenPI preprocessing, for 20k steps at an effective batch size of 128, with a cosine learning rate from 2.5×10^{-5} to 2.5×10^{-6} and 1,000 warm-up steps. The action horizon is 30, matching Z-sight, and the action projections are re-initialized for the 78-D action.

GR00T N1.7. We fine-tune the released GR00T N1.7 checkpoint with the official recipe: the projector, the diffusion transformer and the vision-language layer norm are trained, and the language model and vision encoder are frozen. We train for 20k steps at batch size 128, with a cosine learning rate from 10^{-4} to 0 and 500 warm-up steps, state dropout 0.2, percentile normalization, and the official image augmentation. The action horizon is 30, padded to the pretrained head's 40, and inference uses 4 denoising steps.

Fast-WAM. We use the Fast-WAM architecture without future generation at test time: a Wan2.2-TI2V-5B video diffusion transformer coupled with a 1024-wide action diffusion transformer, trained on 33-frame clips at 224×320 resolution with min–max normalization. We train for 7,500 steps with a cosine learning rate from 10^{-4}, weight decay 0.01 and an effective batch size of 192 (4 per GPU on 8 H200 GPUs, with gradient accumulation over 6 steps). The action horizon is 32.