$\mathcal{Z}$-sight: Unifying Vision-Language-Action and Latent World Modeling

Junjie Ye[1,2,*], Yiqi Zhao[2,*], Muchen Xu[2,*], Celina Shiyu Wang[2], Hao Dong[2], Ruohai Ge[2], Rong Xue[2], Benjamin J Kolligs[1], Aditya Dutt[1], Sasanka Nagavalli[1], Nassim A. Samad[1], Gaurav S. Sukhatme[2], Jyotirmoy V. Deshmukh[2], Yue Wang[2,†], Yuliang Guo[1,†]

[1]Maven Robotics [2]University of Southern California

[*]Equal Contribution [†]Equal Advising

We introduce $\mathcal{Z}$-sight, a unified robot foundation model that couples vision-language-action (VLA) and generative latent world modeling. Video-based world action models (WAMs) show the value of joint future-and-action generation for control, yet inherit the heavy cost of video generators. Recent latent-space policies avoid pixel generation, yet typically align intermediate features or rely on frozen target encoders, decoupling future prediction from the VLM's native feature space. $\mathcal{Z}$-sight instead predicts the future directly in the visual token space in which its VLM perceives the present. Coupling a foresight expert and a continuous action expert with the VLM in a mixture-of-transformers, $\mathcal{Z}$-sight jointly denoises future spatial tokens and actions at inference without pixel reconstruction or external encoders. Trained on human video and robot trajectories, $\mathcal{Z}$-sight achieves 81.9% success on SIMPLE (+9.6 points over the strongest baseline) and 73.3% across six Unitree G1 tasks (+20.0 points) at 110 ms per chunk, while remaining competitive on tabletop benchmarks.

Website: https://junjieye.com/Z-sight/

1 Introduction

Vision-language-action (VLA) models transfer visual and linguistic knowledge to robotic control by building policies on pre-trained VLMs (Zitkovich et al., 2023; Kim et al., 2025; Black et al., 2025b). Recent architectures extend this paradigm by coupling a VLM backbone with a continuous action expert (Black et al., 2025a; Bjorck et al., 2025; Wei et al., 2026a), combining multimodal reasoning with high-frequency action generation.

In parallel, world action models (WAMs) (Ye et al., 2026; Bi et al., 2026; Yuan et al., 2026; Zhou et al., 2026) adapt pre-trained video generators to jointly model future observations and actions, capturing physical dynamics at the cost of large video diffusion backbones. Recent methods avoid pixel generation by moving future supervision into latent space, yet either align intermediate policy features to compact embeddings (Zheng et al., 2025; Li et al., 2026) without generating a future stream, or predict features of frozen target encoders (Zhang et al., 2026b; Sun et al., 2026; Zhang et al., 2026a) that decouple from the VLM's native feature space.

We introduce $\mathcal{Z}$-sight, a unified model that brings joint future-and-action generation into the policy's co-evolving native visual space (Figure 1). A VLM encodes observations into spatial visual tokens. By supervising future predictions against an exponential-moving-average (EMA) teacher of this same vision encoder, perception and prediction remain aligned throughout pre-training and downstream adaptation, without introducing a video generator or an external target encoder.

As shown in Figure 2, $\mathcal{Z}$-sight couples the VLM stream with a foresight expert and a continuous action expert in a mixture-of-transformers (MoT) (Liang et al., 2025). Conditioned on the VLM's instruction, state, and observation history, the foresight expert predicts future spatial tokens via flow matching while the action expert attends directly to these predicted features. During inference, both experts jointly denoise future visual tokens and actions, informing control with online latent dynamics at high efficiency.

We train $\mathcal{Z}$-sight on egocentric human video and cross-embodiment robot trajectories, then adapt it across single-arm, bimanual, and humanoid settings spanning LIBERO (Liu et al., 2023), RoboTwin 2.0 (Chen et al., 2026), SIMPLE (Wei et al., 2026b), and six physical Unitree G1 tasks. $\mathcal{Z}$-sight achieves strong humanoid loco-manipulation performance,

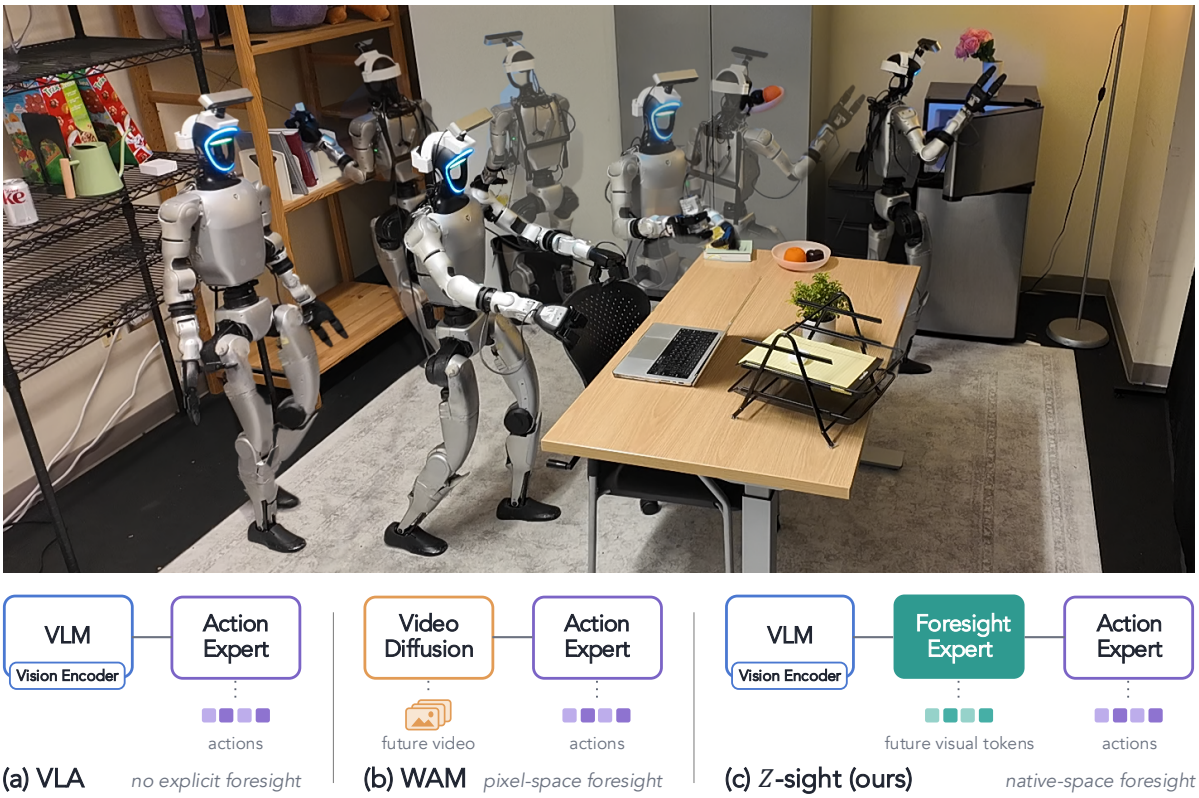


Figure 1. $\mathcal{Z}$-sight predicts the future in the policy's native representation space. *Top:* $\mathcal{Z}$-sight performs diverse loco-manipulation tasks in a real-world setting. *Bottom:* comparison of policy designs. (a) Vision-language-action models (VLAs) map the current observation to actions through a VLM and an action expert. (b) World action models (WAMs) build on a pretrained video generator to predict future video alongside actions. (c) $\mathcal{Z}$-sight (ours) co-denoises future spatial tokens and actions in the co-evolving visual space of its own VLM, allowing the action expert to read the forming prediction during generation.

reaching 81.9% success on SIMPLE and 73.3% on the real robot (+9.6 and +20.0 points over the strongest baselines) at 110 ms per action chunk, while remaining competitive on tabletop manipulation benchmarks. Our contributions are threefold:

- We introduce $\mathcal{Z}$-sight, a unified VLA and latent world model that jointly denoises future spatial visual tokens and continuous actions within the VLM's co-evolving visual representation space.
- Through matched architectural controls and test-time interventions, we demonstrate the advantages of co-evolving native targets and joint denoising over implicit feature alignment and frozen external targets.
- $\mathcal{Z}$-sight achieves strong humanoid loco-manipulation results on SIMPLE and six real-world Unitree G1 tasks at 110 ms per chunk, while remaining competitive on tabletop manipulation benchmarks.

2 Related Work

Vision-language-action models. Vision-language-action (VLA) models adapt pre-trained VLMs to output discrete action tokens (Zitkovich et al., 2023; Kim et al., 2025; Black et al., 2025b; Pertsch et al., 2025) or continuous actions via flow-matching experts (Black et al., 2025a; Bjorck et al., 2025; Wei et al., 2026a). Leveraging internet-scale multimodal pre-training, these policies exhibit strong semantic reasoning and instruction following. To preserve pre-trained representations while learning continuous control, Driess et al. (2025) employ knowledge insulation, training the VLM with discrete action tokens while a flow expert learns continuous actions from detached VLM features. While effective at mapping current context to actions, standard VLAs rely primarily on action labels, providing a relatively sparse supervision signal. $\mathcal{Z}$-sight builds upon this foundation and augments it with generative latent world modeling.

World action models and unified models. World action models (WAMs) capture physical dynamics by predicting future video alongside actions, either jointly denoising video and actions on video diffusion backbones (Ye et al.,

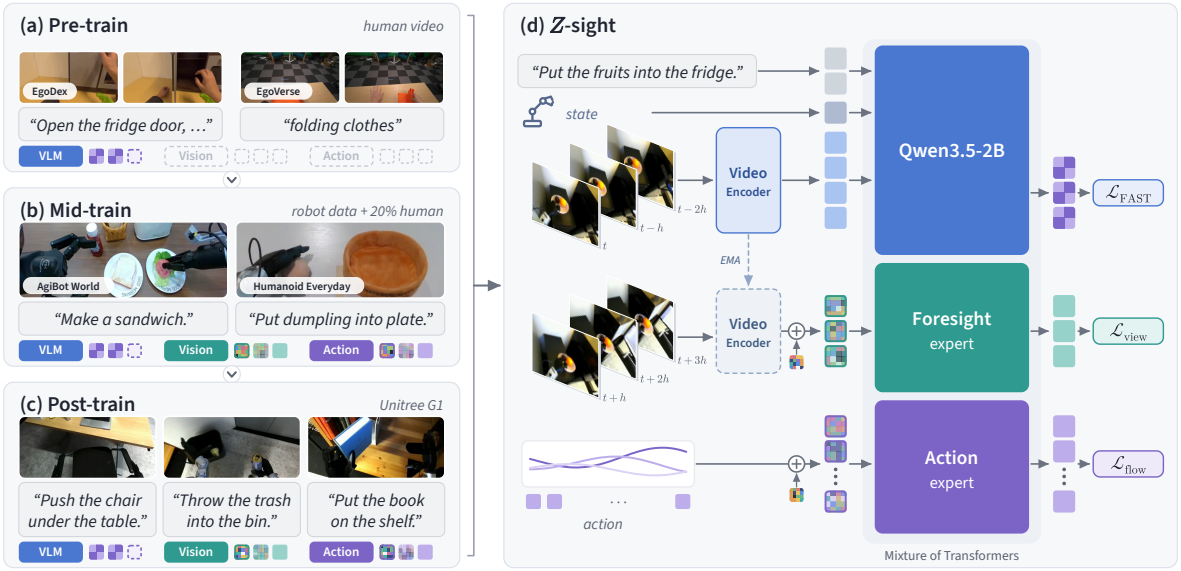


Figure 2. Overview of 𝒵-sight. (a)–(c) Training stages: VLM pre-training on human video, mid-training all streams on mixed robot-human data, and post-training on the target embodiment. **(d) Architecture:** The VLM processes observation history, state, and instructions, supervised by discrete FAST tokens ($\mathcal{L}_{\text{FAST}}$). 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.

2026; Zhou et al., 2026) or using video co-training while skipping test-time generation (Yuan et al., 2026). Motivated by this complementarity, unified models combine a VLM with visual generation and action experts. For example, Motus (Bi et al., 2026) couples a VLM, a video generation expert, and an action expert, while BagelVLA (Hu et al., 2026) interleaves textual plans, generated keyframes, and actions with single-step guidance. Related designs include UP-VLA (Zhang et al., 2025), dual-stream VLAs (Won et al., 2026), F1 (Lv et al., 2025), InternVLA-A1 (Cai et al., 2026a), and MotuBrain (Xiang et al., 2026). However, relying on pixel or codec generation introduces substantial computational cost. 𝒵-sight instead performs joint future-and-action generation directly in the VLM's visual token space without a video generator.

Latent world modeling and predictive policies. Joint-embedding predictive architectures learn representations by predicting features rather than pixels (Baevski et al., 2022; Assran et al., 2023; Bardes et al., 2024) against EMA teachers (Grill et al., 2020), enabling planning with pre-trained visual features (Zhou et al., 2025; Assran et al., 2025). In robotic control (Lyu et al., 2026; Nilaksh et al., 2026), implicit alignment methods align internal policy activations with compact future embeddings (Zheng et al., 2025; NVIDIA, 2025) or spatial and world-model features (Li et al., 2026; Dao et al., 2026) without generating a spatial future stream. Explicit latent predictors instead supervise world tokens with frozen external teachers in DreamVLA (Zhang et al., 2026b), or frozen V-JEPA 2 and SigLIP encoders in VLA-JEPA (Sun et al., 2026) and UniJEPA (Zhang et al., 2026a). Instead, 𝒵-sight models future states as dense spatial tokens in the co-evolving perceptual space of the policy's own vision encoder, jointly denoising future tokens and continuous actions via flow matching (Appendix G).

3 Methodology

𝒵-sight predicts the future in the same visual representation space in which it perceives the present and uses that prediction to generate actions. At time t, given a language instruction ℓ, proprioceptive state $\boldsymbol{s}_t$, and multi-camera observation history $\mathbf{O}_t$, the policy outputs an action chunk $\boldsymbol{A}_t = (\boldsymbol{a}_t, \ldots, \boldsymbol{a}_{t+H-1})$. During training, the future observations within the chunk, $\mathbf{O}_t^+$, serve as an additional supervision signal that is unavailable at test time.

3.1 Model Architecture

Native representation space. The VLM's vision encoder E_ϕ maps an image o to P spatial tokens $E_\phi(o) \in \mathbb{R}^{P\times D}$ consumed by the language model. We call this token space the policy's *native representation space*. Instead of

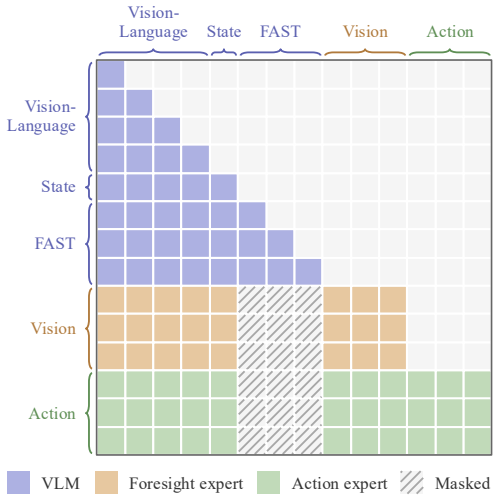


Figure 3. Attention mask of $\mathcal{Z}$-sight. The VLM prefix attends causally and cannot see either expert. The foresight expert attends to the prefix and to itself. The action expert attends to the prefix, the foresight stream, and itself.

synthesizing pixels, $\mathcal{Z}$-sight predicts the spatial tokens that E_ϕ would produce for future observations. This preserves the spatial layout of the scene without introducing a pixel decoder or video generator.

Video encoder for history and future. A single frame lacks motion cues, so we extend E_ϕ into a parameter-free causal video encoder V_ϕ following MEM (Torne et al., 2026). Every fourth ViT block replaces frame-wise attention with causal temporal attention followed by spatial attention, and only the final frame's P tokens are passed to the VLM, retaining the single-image token budget (Appendix A.1). Multiple cameras are encoded independently with learned view-role embeddings. To construct prediction targets, an EMA teacher $V_{\bar{\phi}}$ tracks the online encoder via

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

with decay $\mu \in [0, 1)$. Applying the two encoders to the observation history and future clip yields

$$\boldsymbol{z}_t = V_\phi(\mathbf{O}_t), \quad \boldsymbol{z}_t^+ = \mathrm{LN}\big(\mathrm{sg}\big[V_{\bar{\phi}}(\mathbf{O}_t^+)\big]\big)\,, \tag{2}$$

where $\boldsymbol{z}_t, \boldsymbol{z}_t^+ \in \mathbb{R}^{P\times D}$ are the perceived visual input and the future target, $\mathrm{sg}[\cdot]$ denotes stop-gradient, and $\mathrm{LN}(\cdot)$ applies per-token layer normalization across feature channels to stabilize target scale.

Preventing collapse and forgetting. Because $\boldsymbol{z}_t$ and $\boldsymbol{z}_t^+$ share the same encoder family, backpropagating the future-prediction loss into V_ϕ could cause representation collapse and erode the VLM's pre-trained knowledge. We prevent both issues through two mechanisms. First, stop-gradient EMA targets (Assran et al., 2023) keep $\boldsymbol{z}_t^+$ aligned with the evolving perceptual encoder without allowing the encoder to trivialize its own prediction target. Second, we block $\mathcal{L}_{\text{view}}$ gradients from entering the VLM across all stages. During mid-training, when the newly added experts are randomly initialized, we apply full knowledge insulation (Driess et al., 2025) so both experts attend to the VLM's keys and values through stop-gradient, while the VLM is adapted by autoregressively predicting discrete FAST action tokens (Pertsch et al., 2025) $c_{1:N}$ of $\boldsymbol{A}_t$

$$\mathcal{L}_{\text{FAST}} = -\sum_{n=1}^{N} \log p_\theta\big(c_n \mid c_{<n}, \ell, \boldsymbol{s}_t, \boldsymbol{z}_t\big), \tag{3}$$

where p_θ is the VLM's next-token distribution and θ includes ϕ. Thus V_ϕ adapts solely through action supervision, while its EMA teacher supplies co-evolving future targets. Stage-wise gradient paths are detailed in Appendix A.

Foresight expert. Future prediction is performed by a dedicated foresight expert in the mixture-of-transformers via flow matching in token space. Given visual noise level $\sigma \in [0, 1]$ and $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, it receives noisy future tokens $\boldsymbol{f}^\sigma = \sigma\boldsymbol{\xi} + (1-\sigma)\,\boldsymbol{z}_t^+$ combined with normalized current-frame tokens

$$\boldsymbol{h} = \boldsymbol{f}^\sigma \boldsymbol{W}_{\text{in}} + \mathrm{sg}[\mathrm{LN}(\boldsymbol{z}_t)]\,\boldsymbol{W}_c + \boldsymbol{e}_{\text{pos}}, \tag{4}$$

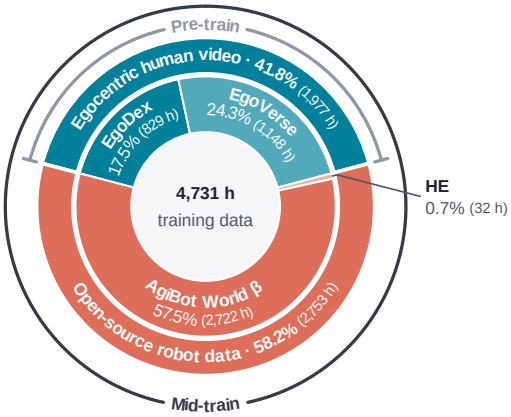


Figure 4. Training data. Pre-training uses 1,977 h of egocentric human video (inner arc) and mid-training uses all 4,731 h (outer circle), combining robot data with 20% human-data replay.

where $\boldsymbol{W}_{\text{in}}$ and $\boldsymbol{W}_c$ project channel dimension D to the expert width and $\boldsymbol{e}_{\text{pos}}$ is a learned position embedding. Zero-initializing $\boldsymbol{W}_c$ allows the expert to gradually acquire a direct current-frame conditioning path alongside VLM attention. Every layer is conditioned on σ via adaptive RMSNorm. Using the x-prediction parameterization (Li and He, 2026), the expert predicts clean tokens $\hat{\boldsymbol{z}}_t^+$ under the implied velocity error

$$\mathcal{L}_{\text{view}} = \mathbb{E}\Big[\big\|(\boldsymbol{z}_t^+ - \hat{\boldsymbol{z}}_t^+) \,/\, \max(\sigma, \sigma_{\min})\big\|^2\Big], \tag{5}$$

where $\sigma_{\min}$ bounds the weight at small noise levels.

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

$$\mathcal{L}_{\text{flow}} = \mathbb{E}\big[\|\hat{\boldsymbol{u}}_t - (\boldsymbol{\epsilon} - \boldsymbol{A}_t)\|^2\big]. \tag{6}$$

During training, we sample $\sigma = \max(\tau, u) \geq \tau$ with a random floor u, exposing the action expert to future streams with varying uncertainty.

Attention pattern and joint denoising. As shown in Figure 3, tokens are ordered as [`VLM` | `FAST` | `foresight` | `action`]. VLM and FAST tokens attend causally, and both experts are masked from reading ground-truth FAST tokens. The foresight stream attends to the VLM prefix and bidirectionally to itself. The action stream attends to the VLM prefix, the foresight stream, and itself, reading predicted future tokens directly in the native visual space. At inference, FAST tokens are omitted, the VLM prefix is encoded once with cached keys and values, and both experts are integrated jointly from $\boldsymbol{f}^1, \boldsymbol{A}^1 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ over K shared Euler steps ($\Delta = -1/K$, $\sigma = \tau$)

$$\boldsymbol{A}^{\tau+\Delta} = \boldsymbol{A}^\tau + \Delta\,\hat{\boldsymbol{u}}_t, \quad \boldsymbol{f}^{\tau+\Delta} = \boldsymbol{f}^\tau + \Delta\,\frac{\boldsymbol{f}^\tau - \hat{\boldsymbol{z}}_t^+}{\max(\tau, \sigma_{\min})}, \tag{7}$$

updating both streams in a single forward pass per step without pixel decoding.

3.2 Training Recipe

We train $\mathcal{Z}$-sight in three stages, comprising VLM pre-training on egocentric human video, full-model mid-training on cross-embodiment robot data with human replay, and post-training on the target embodiment (Figure 4, Appendix B). During pre-training and mid-training, human hands and humanoids use a 48-D task-space hand representation (Wei et al., 2026a), while other robots use end-effector poses with gripper or hand commands. Actions are expressed relative to the current state within each chunk and padded to a shared dimension with per-channel validity masks.

Pre-training on human video. We train only the VLM with $\mathcal{L}_{\text{FAST}}$ on ∼2,000 hours of video from EgoDex (Hoque et al., 2026) and EgoVerse (Punamiya et al., 2026). Hand trajectories are expressed in the current frame to factor out head motion, low-confidence joints are masked, and trajectories are smoothed with a validity-weighted temporal Gaussian filter to suppress frame-to-frame tracking jitter in FAST tokens (Appendix B.2).

Mid-training on cross-embodiment data. We introduce the foresight and action experts and optimize $\mathcal{L}_{\text{FAST}} + \mathcal{L}_{\text{flow}} + \lambda_v \mathcal{L}_{\text{view}}$ under knowledge insulation on ∼2,750 hours from AgiBot World (AgiBot-World-Contributors et al., 2025) and Humanoid Everyday (Zhao et al., 2026), deriving task-space poses for joint-only humanoid logs via forward kinematics (Wei et al., 2026a) and filtering idle segments. Replaying human video at 20% of the sampling mass retains pre-trained priors and supplies diverse visual dynamics.

Post-training on the target embodiment. On target demonstrations, we drop $\mathcal{L}_{\text{FAST}}$ and optimize $\mathcal{L}_{\text{flow}} + \lambda_v \mathcal{L}_{\text{view}}$. Action gradients update the VLM, state projection, and experts, while $\mathcal{L}_{\text{view}}$ updates only the foresight expert with its gradient into the VLM blocked. The EMA teacher continues tracking the adapting vision encoder.

4 Experiments

$\mathcal{Z}$-sight jointly denoises future visual tokens and continuous actions within the VLM's native feature space. Our experiments evaluate the practical benefits and underlying mechanisms of this design. Specifically, we aim to answer four key questions. (1) How does $\mathcal{Z}$-sight perform compared to strong VLAs and video-based WAMs? (Section 4.2) (2) How much do continued future supervision and the co-evolving native target design contribute relative to implicit alignment and frozen external encoders? (Section 4.3) (3) What scene changes does the foresight stream capture, and how does the action expert rely on its content and uncertainty during execution? (Section 4.4) (4) Can $\mathcal{Z}$-sight retain online foresight at high efficiency? (Section 4.5)

4.1 Evaluation Setup

Benchmarks. We evaluate $\mathcal{Z}$-sight on three simulation benchmarks and a physical humanoid. **SIMPLE** (Wei et al., 2026b) evaluates whole-body humanoid loco-manipulation across three levels of variation in materials, distractors, lighting, and target positions (L0–L2). On a physical **Unitree G1**, we evaluate six tasks spanning tabletop manipulation, object transport, and multi-stage loco-manipulation, reporting success rates and stage-based progress scores (Appendices C and C.2). **LIBERO** (Liu et al., 2023) tests single-arm manipulation across spatial, object, goal, and long-horizon suites, while **RoboTwin 2.0** (Chen et al., 2026) evaluates bimanual manipulation over 50 tasks trained on clean demonstrations and tested under clean (C2C) and randomized (C2R) conditions.

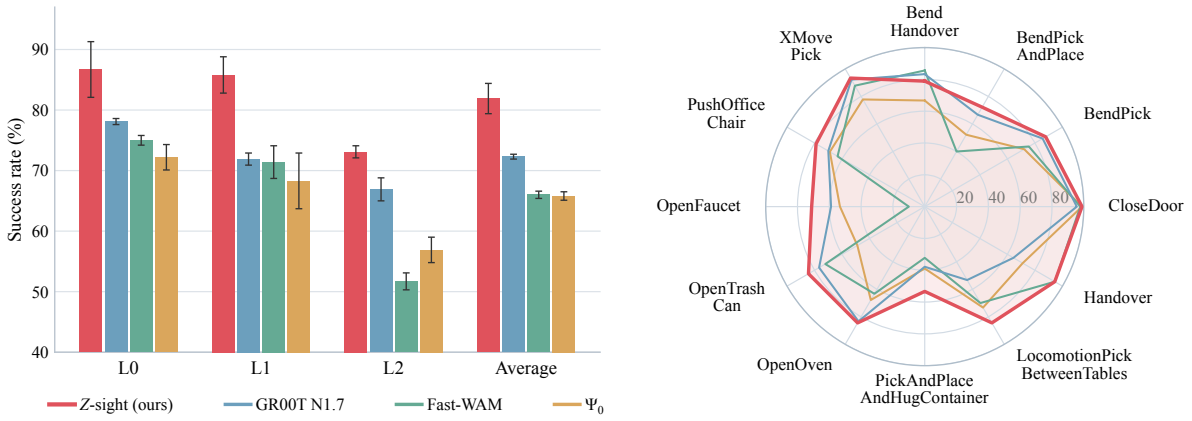
Baselines. Our primary baselines span representative VLAs ($\pi_{0.5}$ (Black et al., 2025a), GR00T N1.7 (NVIDIA, 2026), and Ψ_0 (Wei et al., 2026a)) and a video-based WAM (Fast-WAM (Yuan et al., 2026)). Additional tabletop baselines and adaptation protocols are detailed in Appendices D and E.

4.2 Performance across Embodiments

Because our pre-training and mid-training incorporate human egocentric video and humanoid trajectories, we first evaluate whole-body humanoid loco-manipulation in simulation and the real world, then test whether the same model extends to single-arm and bimanual tabletop manipulation.

Humanoid loco-manipulation in SIMPLE. All four methods are adapted on the same 17-task SIMPLE dataset and evaluated over three passes on the 12 teleoperation tasks (full protocol and motion-planned task results in Appendix E.1). Figure 5a shows that $\mathcal{Z}$-sight achieves 81.9% overall success, outperforming GR00T N1.7 by 9.6 points, Fast-WAM by 15.9 points, and Ψ_0 by 16.1 points, while leading at every difficulty level (86.7% on L0, 85.8% on L1, and 73.1% on L2). Across individual tasks in Figure 5b, $\mathcal{Z}$-sight ranks first or tied on 11 of the 12 tasks, with the largest margins on tasks requiring coupled navigation, whole-body bending, or articulated contact such as `LocomotionPickBetweenTables` (84.4% vs. 73.3% for Ψ_0 and 53.3% for GR00T N1.7), `PickAndPlaceAndHugContainer` (53.3% vs. 38.9%), and `OpenFaucet` (71.1% vs. 58.9%). While action-only VLAs and test-time foresight-free Fast-WAM degrade sharply under target-position randomization at L2, co-denoising future spatial tokens helps $\mathcal{Z}$-sight sustain coordinated whole-body reaching and contact transitions.

Real-world humanoid loco-manipulation. Figures 6 and 7 present our evaluation across six everyday tasks on a physical Unitree G1 with 10 trials per task. $\mathcal{Z}$-sight reaches 73.3% average success (44/60 trials) and 81.7% progress, outperforming the strongest baseline Fast-WAM (53.3% success, 63.8% progress) as well as Ψ_0 (40.0%), GR00T N1.7 (36.7%), and $\pi_{0.5}$ (26.7%), while matching or exceeding every baseline on every task. The advantage is most pronounced on the four multi-stage tasks (`Chair`, `Book`, `Trash`, and `Fruit`), where $\mathcal{Z}$-sight averages 72.5% success compared with 50.0% for Fast-WAM and at most 32.5% for the VLA baselines, completing both four-stage tasks (`Trash` and `Fruit`) in 7 of 10 trials. Whereas VLA baselines frequently stall after completing the initial subgoal, online native-space foresight helps $\mathcal{Z}$-sight carry sequential stage transitions through to completion.



(a) Overall success rate

(b) Per-task success rate

Figure 5. Results on SIMPLE. (a) Success rates by difficulty level and overall average. (b) Success rates across the 12 teleoperation tasks (averaged over L0–L2). Values are means over three evaluation passes of one checkpoint per method, with error bars showing ± one sample standard deviation across passes (ten episodes per task and level per pass). 𝒵-sight leads at every difficulty level (81.9% average, +9.6 points over GR00T N1.7) and ranks first or tied on 11 of the 12 tasks.

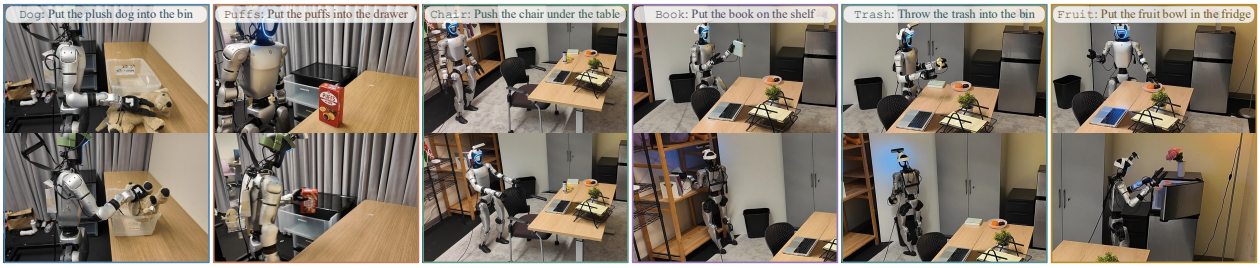


Figure 6. Real-world tasks. The six evaluation tasks on the Unitree G1 with task instructions overlaid. `Dog` and `Puffs` are short pick-and-place tasks, whereas `Chair`, `Book`, `Trash`, and `Fruit` are multi-stage loco-manipulation tasks requiring coordinated whole-body motion and dexterous grasping.

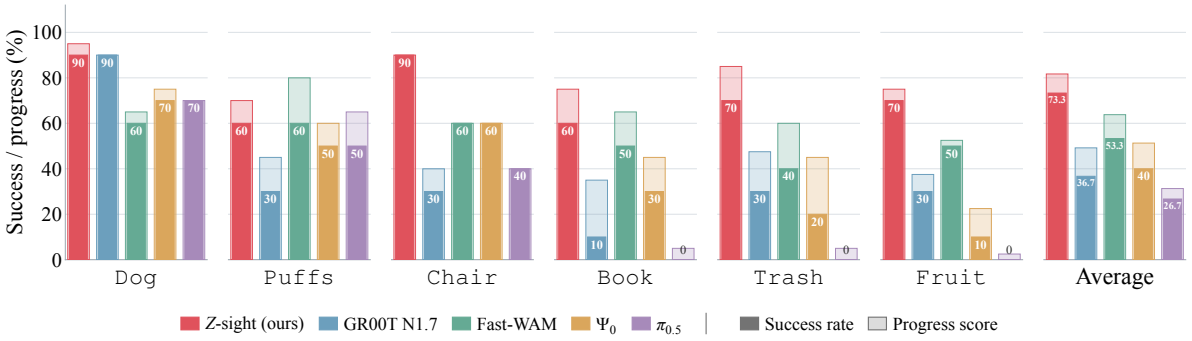


Figure 7. Real-world results on the Unitree G1. Success rate (SR) and progress score (%) over 10 trials per task. Solid bars and labels show SR, overlaid on translucent bars showing progress. Average weights all six tasks equally. We compare Fast-WAM (Yuan et al., 2026), $\pi_{0.5}$ (Black et al., 2025a), GR00T N1.7 (NVIDIA, 2026), and Ψ_0 (Wei et al., 2026a). 𝒵-sight matches or exceeds every baseline on every task, raising overall success to 73.3% (+20.0 points over Fast-WAM) with the largest gains on multi-stage tasks.

Single-arm and bimanual tabletop manipulation. Table 1 reports results on single-arm LIBERO (Liu et al., 2023), where 𝒵-sight achieves 97.7% average success. It outperforms the VLA baselines ($\pi_{0.5}$ at 96.9%, GR00T N1.7 at 97.0%, and starVLA at 96.6%) as well as VLA-JEPA (Sun et al., 2026) (97.2%), with the clearest margin over VLAs on the long-horizon `Long` suite (96.6% vs. 92.4–94.4%), while matching the video-based Motus average without a video generator.

Table 1. LIBERO success rates (%). 𝒵-sight outperforms the VLA baselines and VLA-JEPA overall, while matching Motus.

Method	Spatial	Object	Goal	Long	Avg.
π_0 (Black et al., 2025b)	96.8	98.8	95.8	85.2	94.1
$\pi_{0.5}$ (Black et al., 2025a)	**98.8**	98.2	**98.0**	92.4	96.9
GR00T N1.7 (NVIDIA, 2026)	97.7	98.5	97.5	94.4	97.0
VLA-JEPA (Sun et al., 2026)	96.2	99.6	97.2	95.8	97.2
Motus (Bi et al., 2026)	96.8	99.8	96.6	**97.6**	**97.7**
Fast-WAM (Yuan et al., 2026)	98.2	**100.0**	97.0	95.2	97.6
starVLA (StarVLA Community, 2026)	97.8	98.6	96.2	93.8	96.6
𝒵-sight (ours)	97.2	**100.0**	97.0	96.6	**97.7**

Table 2. RoboTwin 2.0. Success rates (%) under clean (C2C) and randomized (C2R) evaluation.

Method	C2C	C2R	Avg.
GR00T N1.7 (NVIDIA, 2026)	43.6	20.7	32.15
starVLA (StarVLA Community, 2026)	46.52	3.16	24.84
π_0 (Black et al., 2025b)	46.42	16.34	31.38
$\pi_{0.5}$ (Black et al., 2025a)	70.66	**46.00**	58.33
Abot-M0 (Yang et al., 2026)	57.40	30.36	43.88
Xiaomi Robotics-0 (Cai et al., 2026b)	62.86	18.22	40.54
Fast-WAM (Yuan et al., 2026)	77.76	1.92	39.84
X-WAM (Guo et al., 2026)	69.98	25.84	47.91
Spatial Forcing (Li et al., 2026)	77.18	26.74	51.96
𝒵-sight (ours)	**82.44**	39.00	**60.72**

Table 3. Training ablations of the foresight stream. Scores (%), with subscripts showing relative drops from the full model, and gray marking the best. Removing either the future-prediction loss or the entire foresight branch consistently degrades success across real-world and simulated benchmarks, confirming that the gains come from continued predictive supervision rather than added model capacity.

Variant	SIMPLE Hard SR	Real-world Dog SR	Real-world Dog Prog.	LIBERO Avg. SR	RoboTwin 2.0 C2C	RoboTwin 2.0 C2R
𝒵-sight (ours)	73.1	90	95.0	97.70	82.40	40.10
No view loss	$25.0_{(\downarrow 65.8\%)}$	$30_{(\downarrow 66.7\%)}$	$30.0_{(\downarrow 68.4\%)}$	$74.65_{(\downarrow 23.6\%)}$	$60.90_{(\downarrow 26.1\%)}$	$11.60_{(\downarrow 71.1\%)}$
No foresight branch	$68.3_{(\downarrow 6.6\%)}$	$40_{(\downarrow 55.6\%)}$	$40.0_{(\downarrow 57.9\%)}$	$95.35_{(\downarrow 2.4\%)}$	$76.30_{(\downarrow 7.4\%)}$	$22.00_{(\downarrow 45.1\%)}$

Table 4. Future-prediction designs. Success rates (%) on SIMPLE Hard and RoboTwin 2.0. 𝒵-sight consistently outperforms both implicit feature alignment and frozen external targets.

Variant	SIMPLE Hard	RoboTwin 2.0 C2C	RoboTwin 2.0 C2R
FLARE-style (Zheng et al., 2025)	67.5	74.6	20.9
DINOv2-B target (Oquab et al., 2024)	70.8	76.7	26.6
𝒵-sight (ours)	**73.1**	**82.4**	**40.1**

On bimanual RoboTwin 2.0 (Chen et al., 2026), a single RGB-only policy trained on 50 clean demonstrations per task is evaluated over 100 episodes per task under clean (C2C) and randomized (C2R) conditions. In Table 2, 𝒵-sight reaches 82.44% on C2C, surpassing all compared VLAs and WAMs including Fast-WAM (+4.68 points) and Spatial Forcing (+5.26 points). Under randomized evaluation (C2R), 𝒵-sight reaches 39.00%, trailing $\pi_{0.5}$ (46.00%) but substantially outperforming every other VLA and WAM baseline to achieve the highest C2C/C2R average (60.72%). Together, these benchmarks confirm that 𝒵-sight transfers competitively to standard tabletop manipulation.

4.3 Ablating Foresight and Prediction Design

Starting from the shared mid-trained checkpoint, we first isolate the effect of continued future-feature supervision during target-domain adaptation, holding demonstrations and optimization schedules fixed within each benchmark. **No view loss** keeps the foresight expert and token stream while setting $\lambda_v = 0$, so the branch is updated only by action-loss gradients. **No foresight branch** removes the foresight stream altogether, reducing the policy to a standard VLA.

Across all four settings in Table 3, removing the view loss consistently degrades performance, dropping success from 73.1% to 25.0% on SIMPLE Hard, from 90% to 30% on G1 Dog, from 97.70% to 74.65% on LIBERO, and from 40.1% to 11.6% on RoboTwin C2R. Removing the foresight branch entirely also trails the full model (e.g., 68.3% vs. 73.1% on SIMPLE Hard and 22.0% vs. 40.1% on RoboTwin C2R). Because the no-view-loss variant preserves parameter count and architecture, these gains stem from continued predictive supervision rather than extra model capacity.

Prediction design. To test whether different forms of latent future supervision are interchangeable, Table 4 compares two controlled alternatives from the same mid-trained checkpoint. **DINOv2-B target** retains the foresight expert and

Table 5. Foresight prediction and action quality. Pooled over 11,676 held-out G1 windows (predicted α, present β, unrelated γ, true future y).

Comparison	Metric	Value
Predicted vs. future	$\cos(\alpha, y)$	0.955
Present vs. future	$\cos(\beta, y)$	0.921
Unrelated vs. future	$\cos(\gamma, y)$	0.850
Gain over present	$\cos(\alpha, y) - \cos(\beta, y)$	+0.034
Change cosine	$\cos(\alpha - \beta, y - \beta)$	0.646
Future-feature input	**Action MSE ↓**	**Change**
Model prediction α	0.0495	
True future y	0.0436	↓ 11.9%

Table 6. Future-stream interventions. Success rates (%) with fixed weights ($\sigma = \tau$: shared schedule; $\sigma = 1$: pure noise).

		SIMPLE	RoboTwin	
Future stream	Label	Hard	C2C	C2R
Model prediction	$\sigma = \tau$	73.1	82.4	40.1
Gaussian noise	$\sigma = 1$	61.7	79.0	33.9
Gaussian noise	$\sigma = \tau$	16.7	50.5	15.3
Swapped prediction	$\sigma = \tau$	11.7	7.5	4.0
Present frame (Copy)	$\sigma = \tau$	5.8	7.8	2.0
Attention blocked (Detach)	$\sigma = \tau$	7.5	5.8	1.8

Table 7. Joint versus sequential denoising. Success rates (%) with shared weights and median chunk latency on one RTX 4090 (ms).

		SIMPLE	RoboTwin		LIBERO	Latency
Schedule	Steps	Hard	C2C	C2R	Avg.	(ms) ↓
Joint (ours)	10	**73.1**	82.4	**40.1**	**97.70**	**114.8**
Sequential	20	64.2	**82.6**	37.9	97.05	208.0

Table 8. Inference latency on one RTX 4090 (ms). $\mathcal{Z}$-sight runs faster than $\pi_{0.5}$, Ψ_0, and Fast-WAM with online foresight.

Method	Median	Mean ± std	Chunk
$\pi_{0.5}$ (Black et al., 2025a)	120.5	121.0 ± 1.3	30
GR00T N1.7 (NVIDIA, 2026)	60.0	60.3 ± 1.4	30
Fast-WAM (Yuan et al., 2026)	155.9	156.6 ± 2.1	32
Ψ_0 (Wei et al., 2026a)	149.7	149.4 ± 2.6	30
$\mathcal{Z}$-sight (ours)	109.5	109.6 ± 0.2	30

joint denoising but replaces our co-evolving native targets with frozen DINOv2-B/14 features. **FLARE-style** removes the denoising expert and aligns future-query tokens on the policy to the same DINOv2-B features via cosine loss. Keeping an explicit generative foresight stream with frozen DINOv2-B targets improves over implicit alignment (26.6% vs. 20.9% on RoboTwin C2R and 70.8% vs. 67.5% on SIMPLE Hard). Adopting $\mathcal{Z}$-sight's co-evolving native EMA targets yields a further substantial jump to 40.1% on RoboTwin C2R (+13.5 points), 82.4% on C2C (+5.7 points), and 73.1% on SIMPLE Hard, supporting the combination of a co-evolving native target space and joint generative denoising.

4.4 Foresight Content and Policy Reliance

Prediction quality and action error. On 11,676 held-out observation-to-future windows across all six physical G1 tasks, Table 5 compares the predicted future tokens α against the EMA target encoder's ground-truth future tokens y. $\mathcal{Z}$-sight achieves 0.955 cosine similarity, outperforming both copying present tokens β (0.921) and an unrelated future γ (0.850), while reaching 0.646 cosine similarity on feature changes $(\alpha - \beta, y - \beta)$. Moreover, holding initial action noise fixed on the same trajectories, replacing α with ground-truth future tokens y lowers normalized action MSE against demonstrations by 11.9% (from 0.0495 to 0.0436), confirming that more accurate future features directly improve sampled actions.

Closed-loop interventions. To disentangle reliance on predicted content from the stream's noise-level cue, Table 6 evaluates test-time interventions on the foresight stream with fixed weights, with extended benchmarks in Appendix F. Injecting Gaussian noise while labeling the stream unreliable ($\sigma = 1$) retains 61.7% success on SIMPLE Hard and 79.0/33.9% on RoboTwin, as training with $\sigma \geq \tau$ teaches the policy to fall back on current observations under high declared uncertainty. Changing only the label of that noise to the shared schedule $\sigma = \tau$ drops success to 16.7% and 50.5/15.3%. Under the same $\sigma = \tau$ schedule, replacing the stream with another episode's valid prediction (**Swap**) preserves token statistics and the denoising schedule yet collapses success to 11.7% on SIMPLE and 7.5/4.0% on RoboTwin, matching the severe drops from asserting a static scene (**Copy**, 5.8% and 7.8/2.0%) or blocking action-to-foresight attention (**Detach**, 7.5% and 5.8/1.8%). Thus, closed-loop control depends on both declared uncertainty and observation-specific predicted content.

Joint versus sequential denoising. Because $\mathcal{Z}$-sight updates future tokens and actions simultaneously, a natural question is whether conditioning actions on intermediate noisy predictions hurts control compared with waiting for a fully denoised future estimate. Table 7 evaluates the same weights under sequential inference, completing 10 future-denoising steps before running 10 action-denoising steps. Sequential inference yields similar results on LIBERO (97.05% vs. 97.70%) and RoboTwin (82.6/37.9% vs. 82.4/40.1%), but drops SIMPLE Hard success by 8.9 points from 73.1% to 64.2%

while increasing median latency by $1.81\times$ from 114.8 to 208.0 ms on an RTX 4090 (profiling details in Appendix D.2). Single-schedule joint denoising is therefore both faster and at least as effective.

4.5 Inference Efficiency

We time in-process action-chunk generation on a single deployment RTX 4090 over 100 CUDA-synchronized calls on 480×640 observations after 100 warm-up calls (Appendix D.2). As shown in Table 8, $\mathcal{Z}$-sight computes a 30-step chunk in 109.5 ms, faster than $\pi_{0.5}$ (120.5 ms), Ψ_0 (149.7 ms), and Fast-WAM (155.9 ms), even though Fast-WAM skips future generation at test time while $\mathcal{Z}$-sight retains online foresight. GR00T N1.7 runs in 60.0 ms using four denoising steps without future generation. At our 10 Hz control rate, a 30-step chunk spans 3 s of execution for 110 ms of model compute.

5 Conclusion

We presented $\mathcal{Z}$-sight, a unified vision-language-action and latent world model. By jointly denoising future spatial tokens and continuous actions in the VLM's co-evolving perceptual space, $\mathcal{Z}$-sight unifies semantic reasoning, physical foresight, and control without video generators or external encoders. Across simulation and real-world humanoid tasks, $\mathcal{Z}$-sight substantially improves whole-body loco-manipulation over VLA and WAM baselines at high efficiency, and controlled interventions confirm that the policy actively relies on its predicted dynamics. More broadly, our findings show that a VLM's perceptual representations, when adapted jointly with foresight and control, provide an effective substrate for embodied world modeling.

Limitations. The foresight stream currently predicts the future from the observation alone. Conditioning it on candidate actions would let $\mathcal{Z}$-sight compare the outcomes of different actions and open the door to test-time planning in its own representation space.

Acknowledgement

We are grateful to our friends and colleagues, including Hamza Derbas, Kevin Huang, Jason Pries, Songlin Wei, Hongyi Jing, Hongyang Du, and others from Maven Robotics and USC Physical Superintelligence Lab, for their tremendous support, helpful discussions, and insightful suggestions.

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 *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 *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 *ICML*, 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. *TMLR*, 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 *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 *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 *RSS*, 2025b.

Junhao Cai, Zetao Cai, Jiafei Cao, Yilun Chen, Zeyu He, Lei Jiang, Hang Li, Hengjie Li, Yang Li, Yufei Liu, Yanan Lu, Qi Lv, Haoxiang Ma, Jiangmiao Pang, Yu Qiao, Zherui Qiu, Yanqing Shen, Xu Shi, Yang Tian, Bolun Wang, Hanqing Wang, Jiaheng Wang, Tai Wang, Xueyuan Wei, Chao Wu, Yiman Xie, Boyang Xing, Yuqiang Yang, Yuyin Yang, Qiaojun Yu, Feng Yuan, Jia Zeng, Jingjing Zhang, Shenghan Zhang, Shi Zhang, Zhuoma Zhaxi, Bowen Zhou, Yuanzhen Zhou, Yunsong Zhou, Hongrui Zhu, Yangkun Zhu, and Yuchen Zhu. InternVLA-A1: Unifying Understanding, Generation and Action for Robotic Manipulation. *arXiv preprint arXiv:2601.02456*, 2026a.

Rui Cai, Jun Guo, Xinze He, Piaopiao Jin, Jie Li, Bingxuan Lin, Futeng Liu, Wei Liu, Fei Ma, Kun Ma, Feng Qiu, Heng Qu, Yifei Su, Qiao Sun, Dong Wang, Donghao Wang, Yunhong Wang, Rujie Wu, Diyun Xiang, Yu Yang, Hangjun Ye, Yuan Zhang, and Quanyun Zhou. Xiaomi-Robotics-0: An Open-Sourced Vision-Language-Action Model with Real-Time Execution. *arXiv preprint arXiv:2602.12684*, 2026b.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A Scalable Data Generator and Benchmark with Strong Domain Randomization for Robust Bimanual Robotic Manipulation. In *ICML*, 2026.

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 *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 *NeurIPS*, 2020.

Jun Guo, Qiwei Li, Peiyan Li, Zilong Chen, Nan Sun, Yifei Su, Heyun Wang, Yuan Zhang, Xinghang Li, and Huaping Liu. Unified 4D World Action Modeling from Video Priors with Asynchronous Denoising. *arXiv preprint arXiv:2604.26694*, 2026.

Ryan Hoque, Peide Huang, David Yoon, Mouli Sivapurapu, and Jian Zhang. EgoDex: Learning Dexterous Manipulation from Large-Scale Egocentric Video. In *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 *CoRL*, 2025.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial Forcing: Implicit Spatial Representation Alignment for Vision-Language-Action Model. In *ICLR*, 2026.

Tianhong Li and Kaiming He. Back to Basics: Let Denoising Generative Models Denoise. In *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. *TMLR*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. In *ICLR*, 2023.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. In *NeurIPS*, 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.

Qi Lv, Weijie Kong, Hao Li, Jia Zeng, Zherui Qiu, Delin Qu, Haoming Song, Qizhi Chen, Xiang Deng, and Jiangmiao Pang. F1: A Vision-Language-Action Model Bridging Understanding and Generation to Actions. *arXiv preprint arXiv:2509.06951*, 2025.

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 *RSS*, 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.

NVIDIA. GR00T-N1.5: An Improved Open Foundation Model for Generalist Humanoid Robots. NVIDIA Research, 2025.

NVIDIA. GR00T-N1.7-3B. Hugging Face model card, 2026. Accessed November 16, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *TMLR*, 2024.

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 *RSS*, 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. In *RSS*, 2026.

Qwen Team. Qwen3.5: Towards Native Multimodal Agents. Qwen Blog, 2026.

StarVLA Community. StarVLA: A Lego-like Codebase for Vision-Language-Action Model Developing. *arXiv preprint arXiv:2604.05014*, 2026.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. VLA-JEPA: Enhancing Vision-Language-Action Model with Latent World Model. In *CVPR*, 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. In *CoRL*, 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 *RSS*, 2026a.

Songlin Wei, Zhenhao Ni, Jie Liu, Zhenyu Zhao, Junjie Ye, Hongyi Jing, Junkai Xia, Xiawei Liu, Michael Leong, Liang Heng, Di Huang, and Yue Wang. SIMPLE: Simulation-Based Policy Learning and Evaluation for Humanoid Loco-Manipulation. In *CoRL*, 2026b.

John Won, Kyungmin Lee, Huiwon Jang, Dongyoung Kim, and Jinwoo Shin. Dual-Stream Diffusion for World-Model Augmented Vision-Language-Action Model. In *ICML*, 2026.

Chendong Xiang, Fan Bao, Haitian Liu, Hengkai Tan, Hongzhe Bi, James Li, Jiabao Liu, Jingrui Pang, Kiro Jing, Louis Liu, Mengchen Cai, Rongxu Cui, Ruowen Zhao, Runqing Wang, Shuhe Huang, Yao Feng, Yinze Rong, Zeyuan Wang, and Jun Zhu. Motubrain: An Advanced World Action Model for Robot Control. *arXiv preprint arXiv:2604.27792*, 2026.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, Feng Xiong, Xing Wei, Zhiheng Ma, and Mu Xu. ABot-M0: VLA Foundation Model for Robotic Manipulation with Action Manifold Learning. *arXiv preprint arXiv:2602.11236*, 2026.

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 *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, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. UP-VLA: A Unified Understanding and Prediction Model for Embodied Agent. In *ICML*, 2025.

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 *ICML*, 2026a.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, Fan Lu, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. DreamVLA: A Vision-Language-Action Model Dreamed with Comprehensive World Knowledge. In *NeurIPS*, 2026b.

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 *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 *CoRL*, 2025.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World Models on Pre-trained Visual Features enable Zero-shot Planning. In *ICML*, 2025.

Yinghua Zhou, Junjie Ye, Yiqi Zhao, Hao Dong, Celina Shiyu Wang, Ruohai Ge, Tingyi Yang, Basile Van Hoorick, Gaurav Sukhatme, Vitor Guizilini, and Yue Wang. Rolling-WAM: World Action Models with Rolling Imagination. *arXiv preprint arXiv:2609.30247*, 2026.

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 *CoRL*, 2023.

Appendix

Contents

A Architecture and Implementation Details **14**
A.1 Factorized Space-Time Visual Encoding . . . 14
A.2 Model Configuration and Stage-Wise Training . . . 15
A.3 Target-Domain Adaptation and Ablation Controls . . . 15

B Training Data and Preprocessing **15**
B.1 Dataset Composition and Standardization . . . 15
B.2 Temporal Smoothing of Human Hand Trajectories . . . 16

C Real-World Humanoid Setup and Tasks **16**
C.1 Hardware and SONIC Teleoperation-Deployment Pipeline . . . 17
C.2 Evaluation Tasks, Scoring, and Stage-Wise Outcomes . . . 17

D Baselines and Inference Profiling **17**
D.1 Baseline Adaptation Recipes and Compute Budgets . . . 17
D.2 Deployment Latency Profiling . . . 18

E Simulation Benchmark Protocols and Detailed Results **19**
E.1 SIMPLE Whole-Body Loco-Manipulation . . . 19
E.2 LIBERO and RoboTwin 2.0 Tabletop Protocols . . . 20

F Extended Foresight Quality and Intervention Analyses **20**
F.1 Held-Out Prediction Quality and Offline Action-Error Protocol . . . 20
F.2 Additional Inference Interventions on Physical G1 and LIBERO . . . 21

G Discussion of Related Predictive Robot Policies **21**

A Architecture and Implementation Details

This section details the factorized space-time visual encoder (Appendix A.1), shared model configuration and stage-wise training objectives (Appendix A.2), and target-domain adaptation schedules (Appendix A.3).

A.1 Factorized Space-Time Visual Encoding

Inspired by MEM (Torne et al., 2026), the causal video encoder V_ϕ reuses the weights of the VLM's vision encoder E_ϕ and introduces no extra parameters. Most ViT blocks process each frame independently, while every fourth block replaces frame-wise attention with factorized space-time attention reusing 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 at spatial position p in frame j. The block first mixes values causally across time at each spatial location and then attends spatially within each frame over the temporally mixed values

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

$$\boldsymbol{y}_{p,j} = \sum_{p'} \mathrm{softmax}_{p'} \left(\tfrac{1}{\sqrt{d}} \left(\boldsymbol{R}^{\mathrm{s}}_p \boldsymbol{q}_{p,j} \right)^\top \boldsymbol{R}^{\mathrm{s}}_{p'} \boldsymbol{k}_{p',j} \right) \tilde{\boldsymbol{v}}_{p',j}, \tag{A.2}$$

where $\boldsymbol{R}^{\mathrm{t}}_j$ is a 1D rotary embedding of the frame index, $\boldsymbol{R}^{\mathrm{s}}_p$ is the pretrained 2D spatial rotary embedding, d is the head dimension, and $\boldsymbol{y}_{p,j}$ passes through the block's residual connection and MLP (attention heads omitted for brevity). Because temporal mixing is causal, the final frame aggregates the full clip, and passing only its P tokens per camera to

Table A.1. Shared architecture and prediction settings.

Item	Setting
VLM backbone	Qwen3.5-2B (Qwen Team, 2026)
Foresight / action expert width	1024 / 1024
Parameters (VLM / foresight / action)	2.22B / 0.48B / 0.48B (3.18B total)
Temporal encoding	Space-time attention every fourth ViT block
Future prediction target	Spatial tokens of the final future frame
Target normalization	Parameter-free per-token layer normalization
Future-feature loss weight λ_v	0.1
Velocity denominator floor $\sigma_{\min}$	0.05
Action noise τ	$\mathrm{Beta}(1.5, 1)$, rescaled to $[0.001, 1]$
Inference schedule	10 joint Euler steps ($\sigma = \tau$)

Table A.2. Training stages and gradient routing. F, A, and V denote FAST cross-entropy, action flow matching, and future-feature prediction, respectively. Future targets are always stop-gradient EMA features. During mid-training only F updates the VLM, whereas during post-training A updates the VLM while V remains insulated from it. Experts and modality projections are trainable whenever present.

Stage	Data	Losses	Trainable modules	Visual noise σ
Pre-training	EgoDex, EgoVerse	F	VLM and state projection	Not used
Mid-training	AgiBot World, Humanoid Everyday, 20% human replay	F + A + 0.1V	VLM via F, foresight via A and V, action expert via A	$\sigma = \max(\tau, u)$
Post-training	Target-domain demonstrations	A + 0.1V	VLM via A, foresight via A and V, action expert via A	$\sigma = \max(\tau, u)$

the VLM preserves the single-image token budget. Multiple cameras are encoded independently and tagged with learned view-role embeddings.

A.2 Model Configuration and Stage-Wise Training

Table A.1 summarizes the shared model architecture and prediction hyperparameters, and Table A.2 specifies the objectives, gradient paths, and visual-noise schedules across the three training stages. We pre-train and mid-train for 200k and 300k optimizer steps, respectively, at global batch size 1,024 on 32 H200 GPUs. The peak learning rate is 5×10^{-5} after 2k warm-up steps, with a $0.1\times$ multiplier on the vision tower. During mid-training, the EMA target encoder uses decay 0.99. For future-feature prediction, the noise floor is sampled as $u = 1 - \mathrm{sigmoid}(0.8z - 0.8)$ with $z \sim \mathcal{N}(0, 1)$, setting visual noise to $\sigma = \max(\tau, u)$ in mid-training and post-training.

A.3 Target-Domain Adaptation and Ablation Controls

Table A.3 lists the adaptation schedules across all four evaluation settings. All main models optimize $\mathcal{L}_{\text{flow}} + \lambda_v \mathcal{L}_{\text{view}}$ using AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.01, and EMA decay 0.999 for the vision teacher.We decay the learning rate by $10\times$ with a cosine schedule. Warm-up lasts 500 steps for G1 and SIMPLE, 2k for LIBERO, and 1k for RoboTwin. Physical G1 adaptation uses the collected demonstrations with input resolutions of 224×288 (RealSense) and 224×384 (ZED), a 78-D action space, and a 43-D state padded to 48. Across all ablation controls in Table 3, variants share each benchmark's mid-trained initialization, demonstrations, and optimization schedule, including trainable vision and language backbones.

B Training Data and Preprocessing

B.1 Dataset Composition and Standardization

We convert every training source into a unified episodic format (LeRobotDataset v3.0[1]) storing raw multimodal streams including camera images with intrinsics and per-frame extrinsics, proprioception, tracked hand keypoints, and wrist or

[1] https://huggingface.co/docs/lerobot/en/lerobot-dataset-v3

Table A.3. Target-domain adaptation schedules. Batch sizes are global. LR denotes peak expert learning rate, with a $0.1\times$ multiplier on the vision tower and the listed multiplier on the language model (LM). History and stride refer to the causal video encoder in frames and dataset timesteps.

Setting	Steps	Batch	LR	LM scale	Action horizon	History	Stride
Unitree G1	20k	128	10^{-4}	$0.5\times$	30	3	10
LIBERO	80k	128	5×10^{-5}	$1.0\times$	16	2	8
RoboTwin 2.0	100k	256	10^{-4}	$0.5\times$	50	2	25
SIMPLE	50k	256	10^{-4}	$0.5\times$	30	3	10

Table B.1. Training datasets across stages. Hours are computed from frame counts at the native 30 Hz recording rate (10 Hz for our G1 demonstrations). Human datasets use the 48-D hand representation in the camera frame, whereas 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	6.8	3

end-effector poses expressed in the camera frame whenever extrinsics are available. Signals absent from a dataset or lost during tracking receive an explicit ignore marker, and per-dataset normalization statistics are computed with ignored entries excluded. Mapping raw signals into the unified state and action space occurs on the fly at data-loading time, turning ignored entries into per-channel validity masks in the training losses. Table B.1 summarizes all pre-training, mid-training, and G1 post-training sources.

B.2 Temporal Smoothing of Human Hand Trajectories

Per-frame vision-based hand estimates in EgoDex and EgoVerse exhibit millimeter-scale frame-to-frame jitter. Because FAST tokenizes action chunks via discrete cosine transform (DCT) coefficients, high-frequency jitter creates many small non-zero coefficients and inflates sequence length. We smooth each 48-D human trajectory along time using a Gaussian kernel with standard deviation $\sigma_s = 2$ steps normalized by the smoothed validity mask

$$\tilde{x}_t = \frac{\sum_j g_{\sigma_s}(t-j)\, m_j x_j}{\sum_j g_{\sigma_s}(t-j)\, m_j}, \tag{B.1}$$

where $m_j \in \{0, 1\}$ indicates valid entries, while invalid entries remain masked in the loss. We load $3\sigma_s$ padding frames on each side of the training window to avoid boundary artifacts, while keeping the current state $\boldsymbol{s}_t$ unsmoothed. Robot trajectories come from commanded joint angles and are not smoothed.

Table B.2 evaluates smoothing on 1,024 paired 30-step chunks from 45 EgoDex episodes and 3,072 chunks from 24 EgoVerse episodes (1,024 per subset). With our fitted tokenizer, smoothing shortens FAST sequences by 13.1% on EgoDex and 20.9% on EgoVerse on average (and by 18.9% and roughly 32% at the 99th percentile). The released FAST tokenizer exhibits comparable relative reductions (13.1% and 22.9%), confirming that compression gains come from removing jitter rather than vocabulary refitting alone. Median keypoint displacement is only 1.3 mm on EgoDex and 3.0 mm on EgoVerse, with larger 99th-percentile displacements (58.8 mm and 73.8 mm) occurring only at peak velocities of rapid reaching motions.

C Real-World Humanoid Setup and Tasks

Table B.2. Effect of temporal smoothing ($\sigma_s = 0 \rightarrow 2$) on human action chunks. Token counts are means per chunk. Values marked $\approx$ are EgoVerse quantiles averaged across its three subsets.

Metric group	Metric	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

C.1 Hardware and SONIC Teleoperation-Deployment Pipeline

We deploy real-world policies on a Unitree G1 humanoid equipped with two 7-DoF Dex3-1 dexterous hands and a single egocentric head camera. The default head-mounted Intel RealSense D435i provides a narrow field of view suited to tabletop tasks (`Dog` and `Puffs`), whereas tasks involving locomotion (`Chair`, `Book`, `Trash`, and `Fruit`) require wider peripheral visibility and use a head-mounted ZED 2i camera. Each task uses the same camera during data collection and evaluation.

Rather than outputting raw whole-body joint targets, high-level policies act through SONIC (Luo et al., 2026), a learned whole-body motion-tracking controller. Each 78-D action comprises a latent SONIC motion token paired with joint targets for both dexterous hands, while the 43-D proprioceptive state comprises body joint positions and commanded hand targets. Demonstrations are collected through the identical control interface by a teleoperator wearing a VR headset whose tracked upper-body poses, hand commands, and locomotion inputs drive SONIC at 50 Hz, recording synchronized camera frames, proprioception, motion tokens, and hand targets at 10 Hz. During deployment, $\mathcal{Z}$-sight runs on a single NVIDIA RTX 4090 GPU in synchronous closed-loop execution, streaming complete 30-step action chunks at 10 Hz while the SONIC decoder tracks each token in a 50 Hz balance and whole-body control loop.

C.2 Evaluation Tasks, Scoring, and Stage-Wise Outcomes

We collect 557 teleoperated G1 demonstrations in total, comprising 355 demonstrations across six evaluated tasks (49–70 per task, Table C.1) and 202 demonstrations across four additional household skills (such as picking up a bottle, stacking plates, and pouring water) that supply related manipulation priors during multi-task training. We selected the six evaluation tasks for skill diversity and difficulty, spanning short tabletop pick-and-place, object transport, articulated-object manipulation, and locomotion coupled to contact. Within each task, episodes are partitioned at the episode level using a deterministic hash (seed 42, nominal 10% validation allocation with the remainder used for training), and all methods are evaluated from their final training checkpoint over 10 trials per task from matched initial configurations.

As summarized in Table C.1, the six evaluation tasks span short tabletop pick-and-place (`Dog`, `Puffs`, 16–28 s mean duration) and long-horizon multi-stage loco-manipulation (`Chair`, `Book`, `Trash`, `Fruit`, 35–84 s mean duration). Each stage carries equal partial progress credit (0.5 for two-stage tasks and 0.25 for four-stage tasks), and full task success requires completing all stages. Table C.2 reports the complete trial-count distributions over completed subgoals across all six tasks and five evaluated policies.

D Baselines and Inference Profiling

D.1 Baseline Adaptation Recipes and Compute Budgets

In addition to the primary humanoid baselines in Section 4, the tabletop evaluations compare against π_0 (Black et al., 2025b), VLA-JEPA (Sun et al., 2026), Motus (Bi et al., 2026), starVLA (StarVLA Community, 2026), ABot-M0 (Yang et al., 2026), Xiaomi Robotics-0 (Cai et al., 2026b), X-WAM (Guo et al., 2026), and Spatial Forcing (Li et al., 2026). On the physical Unitree G1 and in SIMPLE, all evaluated policies share the same demonstration datasets, state-action interfaces, and whole-body control stacks. Adaptation budgets and trainable modules are summarized in Table D.1.

Ψ_0. We fine-tune the humanoid foundation model of Wei et al. (2026a) using its official implementation for 40,000 steps

Table C.1. Real-world Unitree G1 task specifications. Duration denotes mean demonstration length.

Task	Demos	Duration	Camera	Stages (progress credit)
`Dog`	60	15.9 s	RealSense	Place the toy dog on the box (0.5), contain the dog within the box (0.5)
`Puffs`	49	27.7 s	RealSense	Pick up the puffs box (0.5), turn around and place it in the drawer (0.5)
`Chair`	70	35.4 s	ZED 2i	Push the chair under the table (0.5), reach target pose and return to standing (0.5)
`Book`	61	54.7 s	ZED 2i	Pick up and hold the book (0.5), walk and place it on the shelf (0.5)
`Trash`	58	56.8 s	ZED 2i	Pick up bottle (0.25), pick up paper (0.25), throw bottle in bin (0.25), throw paper in bin (0.25)
`Fruit`	57	83.7 s	ZED 2i	Pick up fruit plate (0.25), open fridge door (0.25), place plate inside (0.25), close door (0.25)

Table C.2. Subgoal completion distributions on the Unitree G1. Entries give trial counts (out of 10) completing $(0, 1, 2)$ subgoals for two-stage tasks and $(0, 1, 2, 3, 4)$ subgoals for four-stage tasks.

Method	`Dog` $(0,1,2)$	`Puffs` $(0,1,2)$	`Chair` $(0,1,2)$	`Book` $(0,1,2)$	`Fruit` $(0,1,2,3,4)$	`Trash` $(0,1,2,3,4)$
Fast-WAM (Yuan et al., 2026)	$(3,1,6)$	$(0,4,6)$	$(4,0,6)$	$(2,3,5)$	$(4,1,0,0,5)$	$(2,1,2,1,4)$
$\pi_{0.5}$ (Black et al., 2025a)	$(3,0,7)$	$(2,3,5)$	$(6,0,4)$	$(9,1,0)$	$(9,1,0,0,0)$	$(9,0,1,0,0)$
GR00T N1.7 (NVIDIA, 2026)	$(1,0,9)$	$(4,3,3)$	$(5,2,3)$	$(4,5,1)$	$(4,3,0,0,3)$	$(2,3,2,0,3)$
Ψ_0 (Wei et al., 2026a)	$(2,1,7)$	$(3,2,5)$	$(4,0,6)$	$(4,3,3)$	$(4,5,0,0,1)$	$(2,2,4,0,2)$
$\mathcal{Z}$-sight (ours)	$(0,1,9)$	$(2,2,6)$	$(1,0,9)$	$(1,3,6)$	$(1,2,0,0,7)$	$(0,1,1,1,7)$

on G1 (5.12M samples) and 50,000 steps on SIMPLE (12.8M samples), evaluating the final checkpoints under the shared multi-task protocols.

$\pi_{0.5}$. We fine-tune the released $\pi_{0.5}$ base checkpoint with all parameters trainable and official OpenPI preprocessing for 20k steps at global batch size 128, using 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 $\mathcal{Z}$-sight, with action projections re-initialized for the 78-D G1 action space.

GR00T N1.7 and unfrozen-VLM control. Following its official adaptation recipe, we fine-tune the released GR00T N1.7 checkpoint by updating the embodiment projector, action diffusion transformer, and vision-language layer normalization while keeping the vision tower and language model frozen. On G1, we train for 20k steps at batch size 128 with a cosine learning rate from 10^{-4} to 0, 500 warm-up steps, state dropout 0.2, percentile normalization, and official image augmentation. Because $\mathcal{Z}$-sight updates its vision and language backbones during adaptation, we additionally train an unfrozen GR00T N1.7 control on SIMPLE with its vision tower, language model, projector, diffusion transformer, and state and action encoders all trainable under the same $50\text{k} \times 256$ budget (Table D.2). Unfreezing the VLM raises GR00T N1.7's average success from 72.3% to 73.9%, while $\mathcal{Z}$-sight remains higher at every difficulty level with an 8.0-point overall advantage (81.9%).

Fast-WAM and training compute. We evaluate the Fast-WAM architecture without test-time video generation, coupling a Wan2.2-TI2V-5B video diffusion transformer with a 1024-wide action diffusion transformer trained jointly on 33-frame clips at 224×320 resolution with min-max normalization (action horizon 32, cosine learning rate from 10^{-4}, weight decay 0.01, and global batch size 192). Because Fast-WAM backpropagates through a 5B video diffusion backbone over 33-frame video clips at every step, its per-sample training cost is substantially higher than that of a VLA policy, so fewer sample presentations correspond to comparable wall-clock GPU compute. Accordingly, we set Fast-WAM's step budgets to match $\mathcal{Z}$-sight's training GPU-hours on both real-world G1 ($7.5\text{k} \times 192 = 1.44\text{M}$ samples) and SIMPLE ($35\text{k} \times 192 = 6.72\text{M}$ samples). Training losses converge in both settings, and in our preliminary results extending training further did not improve downstream performance.

D.2 Deployment Latency Profiling

Table 8 measures in-process action-chunk computation on one RTX 4090 with 480×640 inputs, 100 warm-up calls, 100 timed calls, and CUDA synchronization. GR00T N1.7 uses four denoising steps, whereas $\pi_{0.5}$, Fast-WAM, Ψ_0, and $\mathcal{Z}$-sight use ten. These deployment-level measurements isolate model compute and exclude image acquisition, network transport, and low-level whole-body control. Disabling $\mathcal{Z}$-sight's foresight stream reduces model computation by approximately 45 ms per chunk, measuring the incremental cost of online latent foresight directly.

Table D.1. Adaptation budgets on SIMPLE and real-world G1. Budgets are shown as optimizer steps × global batch size = sample presentations. Trainable modules list the default recipe.

Method	Budget (steps × batch = samples)		Default trainable modules
	SIMPLE	Real-world G1	
$\pi_{0.5}$ (Black et al., 2025a)	–	20k × 128 = 2.56M	VLM, action expert, and projections
GR00T N1.7 (NVIDIA, 2026)	50k × 256 = 12.8M	20k × 128 = 2.56M	Projector, action transformer, VLM layer norm
Fast-WAM (Yuan et al., 2026)	35k × 192 = 6.72M	7.5k × 192 = 1.44M	Video and action transformers, state encoder
Ψ_0 (Wei et al., 2026a)	50k × 256 = 12.8M	40k × 128 = 5.12M	VLM, action expert, and projections
$\mathcal{Z}$-sight (ours)	50k × 256 = 12.8M	20k × 128 = 2.56M	VLM, foresight and action experts, projections

Table D.2. GR00T N1.7 trainable-backbone control on SIMPLE. Success rates (%) averaged over three evaluation passes.

Model	L0	L1	L2	Avg.
GR00T N1.7 (official, frozen VLM)	78.1	71.9	66.9	72.3
GR00T N1.7 (unfrozen VLM)	78.3	79.2	64.2	73.9
$\mathcal{Z}$-sight (ours)	**86.7**	**85.8**	**73.1**	**81.9**

Table E.1. Full SIMPLE benchmark summary. Success rates (%) averaged over the 12 Teleop tasks, 5 MP tasks, and all 17 tasks.

Method	Teleop (12)	MP (5)	All 17
GR00T N1.7 (NVIDIA, 2026)	72.3	30.0	59.9
Fast-WAM (Yuan et al., 2026)	66.0	25.3	54.1
Ψ_0 (Wei et al., 2026a)	65.8	20.0	52.4
$\mathcal{Z}$-sight (ours)	**81.9**	**52.0**	**73.1**

For the joint-versus-sequential comparison in Table 7, we time the same SIMPLE checkpoint on one idle RTX 4090 with three-frame 360 × 640 clips and 30-step action chunks (20 warm-ups and 100 CUDA-synchronized calls per mode in separate processes, caching the VLM prefix). Joint denoising (10 shared steps) averages 114.8 ± 0.1 ms (median 114.8 ms), whereas sequential denoising (10 future steps followed by 10 action steps) averages 208.1 ± 0.3 ms (median 208.0 ms).

E Simulation Benchmark Protocols and Detailed Results

E.1 SIMPLE Whole-Body Loco-Manipulation

SIMPLE (Wei et al., 2026b) evaluates whole-body humanoid loco-manipulation through a 50 Hz decoupled whole-body controller (36-D actions, 32-D proprioceptive state). All methods train on the same 17-task dataset and are evaluated across three levels of domain randomization, where L0 varies materials and distractors, L1 adds lighting variation, and L2 (i.e., Hard) varies target initial positions (10 episodes per task and level per pass). The dataset comprises 12 teleoperation (Teleop) tasks and 5 motion-planned (MP) tasks. Across all trained policies, we observe a substantial performance gap between the Teleop and MP tasks, which we attribute to their distinct demonstration distributions. For a meaningful comparison, our main experiments focus on the 12-task Teleop suite, averaged over three evaluation passes of one checkpoint per method (1,080 episodes total, measuring evaluation variability). For computational tractability across multiple trained variants, the matched mechanism ablations and interventions on SIMPLE in Tables 3, 4, 6, and 7 evaluate one pass on the 12 Teleop tasks at L2 (10 episodes per task, 120 episodes per entry). For completeness, Tables E.1 and E.2 report aggregate and per-task success rates across both Teleop and MP suites for all 17 tasks.

Table E.2. Per-task SIMPLE results across all 17 tasks. Success rates (%) averaged over L0, L1, and L2 (10 episodes per level).

Suite	Task	$\mathcal{Z}$-sight (ours)	GR00T N1.7 (NVIDIA, 2026)	Fast-WAM (Yuan et al., 2026)	Ψ_0 (Wei et al., 2026a)
Teleop	BendHandoverTeleop	78.9	83.3	**85.6**	66.7
Teleop	BendPickAndPlaceTeleop	**72.2**	66.7	40.0	52.2
Teleop	BendPickTeleop	**87.8**	85.6	75.6	72.2
Teleop	CloseDoorTeleop	**98.9**	95.6	97.8	**98.9**
Teleop	HandoverTeleop	**94.4**	64.4	**94.4**	71.1
Teleop	LocomotionPickBetweenTablesTeleop	**84.4**	53.3	70.0	73.3
Teleop	OpenFaucetTeleop	**71.1**	58.9	10.0	53.3
Teleop	OpenOvenTeleop	**84.4**	83.3	63.3	67.8
Teleop	OpenTrashCanTeleop	**84.4**	76.7	72.2	48.9
Teleop	PickAndPlaceAndHugContainerTeleop	**53.3**	37.8	32.2	38.9
Teleop	PushOfficeChairTeleop	**78.9**	70.0	63.3	68.9
Teleop	XMovePickTeleop	**93.3**	92.2	87.8	77.8
MP	BendPickAndPlaceOnSofaMP	**30.0**	10.0	3.3	3.3
MP	BendPickMP	**90.0**	76.7	43.3	13.3
MP	PickAndBendPlaceMP	26.7	**56.7**	0.0	20.0
MP	TabletopGraspMP	**80.0**	3.3	56.7	60.0
MP	TabletopHandoverMP	**33.3**	3.3	23.3	3.3
Teleop	Average (12 tasks)	**81.9**	72.3	66.0	65.8
MP	Average (5 tasks)	**52.0**	30.0	25.3	20.0
All	Average (17 tasks)	**73.1**	59.9	54.1	52.4

Table F.1. Offline action-chunk error under controlled future-feature inputs. Normalized MSE against held-out G1 demonstration chunks with shared initial action noise.

Future-feature input to action expert	Normalized action MSE ↓	Relative change
Model's own prediction α ($\sigma = \tau$)	0.0495	Reference
Ground-truth future y, noised to $\sigma = \tau$	**0.0436**	-11.9%
Clean present frame β ($\sigma = 0$)	0.0635	$+28.3\%$

E.2 LIBERO and RoboTwin 2.0 Tabletop Protocols

LIBERO. We jointly train a single $\mathcal{Z}$-sight policy across all 40 tasks from the four LIBERO suites (`Spatial`, `Object`, `Goal`, and `Long`) for 80k steps at global batch size 128 (Table A.3). Evaluation uses 50 episodes per task (500 per suite, 2,000 episodes per variant).

RoboTwin 2.0. Following the official clean-only RGB protocol, we jointly train a single $\mathcal{Z}$-sight policy across all 50 bimanual tasks using 50 clean demonstrations per task (2,500 total) without measured depth, executing 25 actions before replanning. The main benchmark comparison in Table 2 evaluates 100 episodes per task under clean (C2C) and randomized (C2R) conditions (5,000 episodes per condition). For computational tractability across multiple trained variants, the matched mechanism ablations and interventions in Tables 3, 4, and 6 evaluate 20 episodes per task and condition (1,000 episodes per condition), where the full model scores 82.40% on C2C and 40.10% on C2R.

F Extended Foresight Quality and Intervention Analyses

F.1 Held-Out Prediction Quality and Offline Action-Error Protocol

For the offline evaluation in Table 5, we test the deployed physical G1 checkpoint and its EMA target encoder on episode-level held-out trajectories across all six G1 tasks. Out of 12,480 candidate observation-to-future windows, 804 extend past the end of an episode without valid future frames and are excluded, leaving 11,676 valid windows. Across individual tasks, the predicted-over-present similarity gain $\cos(\alpha, y) - \cos(\beta, y)$ is positive on every one of the six tasks, with $\alpha - \beta$ capturing both object displacement and egocentric head-camera motion.

To isolate how future-token accuracy affects action generation on the same held-out windows, we compute normalized open-loop action-chunk MSE against demonstration chunks while holding the initial Gaussian action-noise sample fixed across conditions and varying only the future-feature stream supplied to the action expert (Table F.1). Noising ground-truth EMA future tokens y to the shared denoising level $\sigma = \tau$ reduces action MSE on every task individually, whereas supplying clean present-frame tokens β ($\sigma = 0$) raises action MSE by 28.3%.

Table F.2. Additional test-time interventions on physical G1 and LIBERO. Scores (%) with fixed model weights on physical G1 `Dog` (10 trials) and LIBERO (2,000 episodes across four suites, including 500 on `Long`).

	Real-world G1 `Dog`		LIBERO	
Variant	SR	Progress	Avg. SR	`Long`
Model prediction ($\sigma = \tau$)	**90**	**95.0**	**97.70**	**96.6**
Gaussian noise ($\sigma = 1$)	60	60.0	96.50	95.4
Gaussian noise ($\sigma = \tau$)	–	–	75.15	50.8
Present frame (Copy, $\sigma = \tau$)	0	0.0	7.05	0.0

F.2 Additional Inference Interventions on Physical G1 and LIBERO

Complementing the SIMPLE Hard and RoboTwin 2.0 interventions in Table 6, Table F.2 reports test-time interventions on physical G1 `Dog` and LIBERO with model weights held fixed (on `LIBERO-Long`, sequential denoising in Table 7 scores 94.8% vs. 96.6% for joint denoising). Replacing the predicted stream with Gaussian noise under the shared $\sigma = \tau$ schedule lowers LIBERO average success from 97.70% to 75.15%, with a 45.8-point drop on long-horizon `LIBERO-Long` (96.6% to 50.8%). Declaring Gaussian noise unreliable via $\sigma = 1$ preserves 96.50% average success on LIBERO and retains 60% on physical G1 `Dog`, whereas asserting a static future (**Copy**, $\sigma = \tau$) collapses success to 0% on G1 `Dog` and 7.05% on LIBERO.

G Discussion of Related Predictive Robot Policies

While recent robotic policies increasingly incorporate future visual signals beyond pixel reconstruction, they differ fundamentally in how future states are represented and coupled to action generation. Implicit world-modeling methods such as FLARE (Zheng et al., 2025) (adopted in GR00T N1.5 (NVIDIA, 2025)) align intermediate activations of an action diffusion transformer toward compact action-aware embeddings via cosine loss. This shapes internal representations during training without constructing an explicit spatial future stream that participates in action denoising. Conversely, video-based WAMs (Ye et al., 2026; Bi et al., 2026; Yuan et al., 2026) and codec-based unified policies such as F1 (Lv et al., 2025) and InternVLA-A1 (Cai et al., 2026a) generate discrete tokens or continuous latents of a separate pre-trained image or video tokenizer. Even when explicit pixel decoding is omitted along the action path at test time, future generation operates in a separate generative codec space rather than the visual representation space that grounds VLM perception.

Among policies that explicitly predict latent visual features, existing designs rely on frozen target encoders. DreamVLA (Zhang et al., 2026b) supervises world queries with dynamic regions, depth, and semantic features from frozen external teachers and omits its prediction decoders at deployment, while VLA-JEPA (Sun et al., 2026) predicts latent states against a frozen external V-JEPA 2 video encoder. Closer in architecture is UniJEPA (Zhang et al., 2026a), which also attaches a feature-prediction expert and an action expert to a PaliGemma VLM in a mixture-of-transformers, yet differs from $\mathcal{Z}$-sight across target space, prediction objective, and action coupling. First, UniJEPA predicts future features in the space of a frozen target encoder (an initial SigLIP copy or DINOv3), while fine-tuning the policy's own vision encoder on robot data. Its action expert therefore reads current observations and predicted futures from two different spaces that drift apart during training. Second, its predictor uses deterministic MSE regression, collapsing multimodal futures toward their conditional mean rather than modeling their distribution. Third, its noise-free foresight stream is computed once in a prefix pass and cached prior to action denoising. $\mathcal{Z}$-sight differs on all three axes: an online EMA of the policy's own causal video encoder keeps current observations and predicted futures in one co-evolving feature space, foresight tokens are generated via conditional flow matching, and foresight and continuous actions are denoised jointly across shared Euler steps.