GEOWRA: GEOMETRIC LATENT WORLD RESIDUAL ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

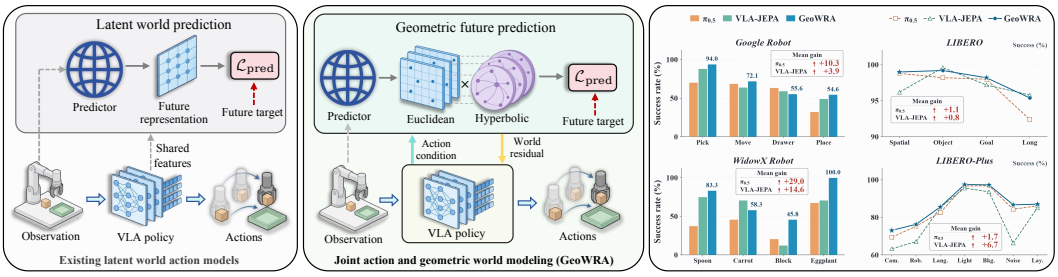
World models and vision-language-action (VLA) models provide complementary capabilities for robotic manipulation through predictive modeling and multimodal action generation. However, effectively integrating world models and VLAs into a unified architecture remains a key challenge, as existing approaches are structurally decoupled, lacking intrinsic interaction between world prediction and action generation. To address this challenge, we propose Geometric Latent World Residual Action Models (**GeoWRA**), a unified framework that couples latent world prediction with action generation. First, we establish a bidirectionally coupled architecture in which the VLA's latent actions condition future prediction, while predicted futures in turn guide action generation through bounded world residual feedback. Second, to capture the relational structure of embodied scenarios, we introduce a Euclidean-hyperbolic product representation that jointly models future features and their relationships by embedding predicted future states in hyperbolic space, providing a geometric inductive bias for learning task-relevant predictive representations. Third, we construct a joint action and geometric world objective that integrates action supervision with future prediction under hyperbolic geometric regularization, enabling geometrically consistent future representations to guide action generation. Extensive evaluations in both simulation and real-world settings show that GeoWRA delivers competitive performance across a diverse range of robotic manipulation tasks.

1 INTRODUCTION

World models (Kim et al., 2026; Guo et al., 2026; Lin et al., 2026) and vision-language-action (VLA) models (Zhang et al., 2026a; Song et al., 2026; Kim et al., 2025) provide complementary capabilities for robotic manipulation: world models anticipate future environmental changes, while VLAs translate visual observations and language instructions into executable actions (Hou et al., 2026; Ye et al., 2026; Wang et al., 2026b). Advances in predictive representation learning and robot pretraining have motivated frameworks that connect future prediction with action generation (Cen et al., 2025; Zheng et al., 2025b). This connection is particularly relevant to manipulation, where generating appropriate motions benefits from anticipating their effects on the scene. For example, placing an object involves local gripper-object interactions, spatial relationships with the destination, and their evolution over successive actions. Such futures are both *action-dependent* and *relational*, highlighting the importance of structured future prediction for action generation.

Despite progress in world action modeling (Tian et al., 2025; Shen et al., 2026), the integration of predictive learning and VLA action generation remains an architectural challenge (Lu et al., 2026; Zhu et al., 2026). Latent predictive approaches provide a promising basis by learning future representations alongside action policies (Sun et al., 2026; Zhang et al., 2025). By supervising scene dynamics without detailed image reconstruction, these approaches connect predictive information with policy learning. However, these approaches suffer from structural decoupling between world prediction and action generation: predicted futures mainly shape the learned representations rather than guiding action generation in a controlled manner, so the resulting policies often fail to generate precise actions. This motivates a central question: *how can world models and VLAs be effectively unified to exploit their complementary capabilities for robotic manipulation?*

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.



(a) GeoWRA versus existing latent world action models

(b) Performance across different benchmarks

Figure 1: **Bidirectionally coupled geometric World-VLA modeling and manipulation performance.** (a) Architectural comparison between existing latent world action models and GeoWRA, which integrates action-conditioned geometric prediction into action generation through bounded world residual feedback. (b) Performance across three manipulation benchmarks. GeoWRA achieves higher average success rates than $\pi_{0.5}$ and VLA-JEPA.

Building on these insights, we propose **GeoWRA**, a unified framework for latent world prediction and VLA action generation. As illustrated in Figure 1, we establish an architecture that bidirectionally couples latent world prediction and VLA action generation, forming the foundation of GeoWRA. Specifically, *from action to world*, the world model conditions latent future prediction on the VLA's latent actions. *From world to action*, these predicted futures are returned through bounded world residual feedback, allowing action representations and predicted consequences to jointly guide action generation. The bounded world residual feedback mechanism controls the magnitude of predictive feedback while retaining the underlying action expert and its generation schedule, without introducing additional action-generation iterations. Within this architecture, the content and organization of predicted future representations determine what information is conveyed through feedback to guide action generation.

Building upon this coupled framework, we introduce a Euclidean-hyperbolic product representation to capture the relational structure of embodied scenarios that individual feature alignment alone leaves implicit. It retains local future features in Euclidean space and embeds predicted future regions in hyperbolic space to explicitly model their structural relationships for subsequent action generation. This formulation provides a geometric inductive bias for learning task-relevant predictive representations. To translate this geometric inductive bias into explicit learning constraints, we formulate a joint action and geometric world objective that integrates action supervision with future prediction under hyperbolic geometric regularization. Specifically, hyperbolic point alignment and relational distance matching explicitly constrain the geometric structure of predicted futures. Together, these objectives integrate predictive fidelity and geometric consistency into policy learning, enabling the geometric structure of predicted futures to inform action generation. Our main contributions can be summarized as follows:

- We establish a unified framework for bidirectionally coupled World-VLA modeling, in which latent actions condition world prediction, and predicted futures guide action generation through bounded world residual feedback, while retaining the underlying action expert and its original generation schedule.
- Building upon this coupled architecture, we introduce a Euclidean-hyperbolic product representation that combines local future features with explicit relational structure among the predicted future regions, providing a geometric inductive bias for learning task-relevant predictive representations.
- We construct a joint action and geometric world objective that jointly optimizes latent world prediction alongside action generation by coupling action supervision with future feature alignment, hyperbolic point alignment, and relation matching, thereby learning predictive representations for control under explicit geometric constraints.
- We conduct comprehensive experiments on three simulation benchmarks and six real-world manipulation tasks, demonstrating GeoWRA's advantages over both world-model-based approaches and VLA-based policies in overall success rate and robustness across diverse tasks and environmental conditions.

2 Related Work

2.1 World Action Models

World Action Models (WAMs) (Kim et al., 2026; Ding et al., 2025; Intelligence et al., 2026) connect environmental prediction with planning, evaluation, or action generation. Planning-based approaches use predicted futures to select actions: V-JEPA 2-AC (Assran et al., 2025) learns latent dynamics over features from a frozen video encoder, whereas LeWorldModel (Maes et al., 2026) jointly learns an encoder and predictor from pixels. Beyond explicit planning, predictive information can also be incorporated into policy learning through joint modeling, predictive supervision, or representation transfer. WorldVLA (Cen et al., 2025) adopts an autoregressive framework for future image prediction and action generation, while VLA-JEPA (Sun et al., 2026) uses latent future prediction to learn action-relevant representations. JEPA-VLA (Miao et al., 2026) incorporates visual representations extracted by a frozen V-JEPA 2 encoder into the policy to guide action generation. In contrast, GeoWRA establishes a unified framework with explicit bidirectional predictive feedback: latent actions condition future prediction, while predicted futures are returned through bounded residuals to guide action generation.

2.2 Hyperbolic Representation Learning

Hyperbolic representation learning (Yang et al., 2022; Peng et al., 2026; Liu et al., 2026) exploits negatively curved spaces and their exponential volume growth to represent hierarchical structures. Poincaré embeddings (Nickel & Kiela, 2017) learn compact hierarchical representations, while hyperbolic graph neural networks (Liu et al., 2019) extend feature transformation and aggregation to curved spaces (Chami et al., 2019). In embodied learning, HMVLA (Wang et al., 2026a) performs hierarchical vision-language alignment, HyperMVP (Yang et al., 2026a) learns Lorentz-space multiview representations for visuomotor manipulation, and HERD (Dong et al., 2024) embeds coarse-to-fine robot morphologies for design optimization. For visual planning, GeoWorld (Zhang et al., 2026b) combines world model with geometric energy minimization over latent trajectories. In contrast, GeoWRA learns a Euclidean–hyperbolic product representation within a bidirectionally coupled VLA architecture, capturing future features and structural relationships among predicted regions. Its joint objective couples action supervision with future feature alignment and hyperbolic relation matching to guide policy learning.

3 Method

GeoWRA is a unified framework that coherently integrates three progressively connected designs: bidirectional coupling between world prediction and VLA action generation, a Euclidean-hyperbolic representation of predicted futures, and a joint objective for action and geometric learning, thereby enabling structured future predictions to guide action generation (Figure 2). First, a bidirectionally coupled World-VLA architecture enables latent action tokens to condition geometric future prediction and returns predicted futures through bounded world residual feedback to guide action generation. Building on this coupling, a Euclidean-hyperbolic product representation specifies the feedback content through local future features and their relational structure. Finally, a joint action and geometric world objective learns these representations through action supervision, Euclidean feature alignment, hyperbolic point alignment, and hyperbolic relation matching.

3.1 Preliminary

Flow-matching action expert. At physical time t, a VLA maps an observation o_t, instruction ℓ, and optional proprioceptive state s_t to an action chunk $a \in \mathbb{R}^{L \times d_a}$ of length L and action dimension d_a. A flow-matching action expert (Lipman et al., 2022; Intelligence et al., 2025) learns the velocity along the linear noise-to-demonstration path

$$a_\tau = (1-\tau)\epsilon + \tau a, \qquad u^{\mathrm{FM}} = \frac{\mathrm{d}a_\tau}{\mathrm{d}\tau} = a - \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \quad \tau \sim p_\tau, \tag{1}$$

where p_τ is a distribution on $[0, 1]$ and the flow time τ is distinct from the physical time t. The expert $v_\theta(a_\tau, \tau; C_t, s_t)$ regresses u^{FM} from action tokens C_t. At inference, integrating the learned velocity from $\bar{a}_0 = \epsilon$ to $\tau = 1$ yields the generated trajectory $\bar{a}_\tau$ and action chunk $\hat{a} = \bar{a}_1$.

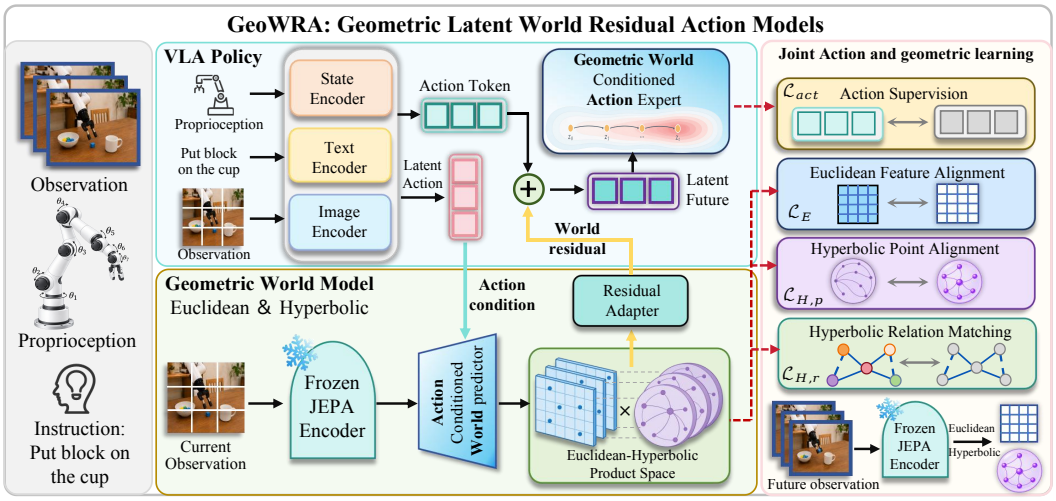


Figure 2: **Overall framework of GeoWRA.** We first establish a bidirectionally coupled World-VLA architecture that integrates latent future prediction with action generation through bounded world residual feedback. Building on this coupling, we introduce a Euclidean-hyperbolic product representation that combines local future features with explicit structural relations among the predicted future regions. We further construct a joint action and geometric world objective that integrates action supervision with future prediction under hyperbolic geometric regularization.

3.2 Bidirectionally Coupled Geometric World-VLA Architecture

Action-conditioned geometric future prediction. *From action to world*, let $\mathcal{F}_\vartheta$ denote the VLA backbone, E_{T} a frozen V-JEPA visual encoder, and W_ϕ a trainable geometric world predictor parameterized by ϕ. The backbone exposes latent-action tokens $q_t \in \mathbb{R}^{N_q \times d_q}$ and action tokens $C_t \in \mathbb{R}^{N_C \times d_C}$. Latent-action tokens q_t provide learned action-conditioning representations for the world predictor, whereas C_t conditions the action expert. The frozen encoder provides the current visual representation z_t. We write

$$(q_t, C_t) = \mathcal{F}_\vartheta(o_t, \ell, s_t), \qquad z_t = \mathrm{sg}[E_{\mathrm{T}}(\mathrm{Rep}(o_t))], \qquad (\widehat{E}_{t,\tau}, \widehat{H}_{t,\tau}) = W_\phi(z_t, q_t, a_\tau, \tau), \quad (2)$$

where Rep repeats o_t into a short clip and sg denotes stop-gradient. The flow time τ identifies the generation stage, and q_t supplies latent-action conditioning. The latent-action tokens q_t remain fixed within each action-generation call. The predictor additionally receives the evolving action a_τ and flow time τ; the connection from a_τ is omitted from Figure 2 for clarity. Bidirectional coupling combines action-conditioned prediction with predictive feedback to the action expert. The predictor outputs Euclidean and hyperbolic future token banks $\widehat{E}_{t,\tau}$ and $\widehat{H}_{t,\tau}$, whose construction is introduced in Section 3.3. Recorded future observations do not enter W_ϕ and are used only to construct training targets. Predictor input encodings and temporal layouts are detailed in Appendix C.1.

Geometric-world-conditioned action generation. *From world to action*, a residual adapter first reads the predicted future using action tokens as queries (the query connection is omitted in Figure 2 for clarity). Hyperbolic tokens are mapped token-wise to the origin tangent space, and both future banks are projected to a shared width d:

$$\begin{aligned} U_{t,\tau} &= \mathrm{Concat}\Big(\mathcal{P}_E(\widehat{E}_{t,\tau}), \mathcal{P}_H(\log_0^c(\widehat{H}_{t,\tau}))\Big), \\ R_{t,\tau} &= \mathcal{P}_O(\mathrm{MHA}(\mathcal{P}_Q(\mathrm{LN}_0(C_t)), U_{t,\tau}, U_{t,\tau})). \end{aligned} \quad (3)$$

Here Concat concatenates along the token dimension, $\log_0^c$ acts token-wise, and LN_0 is channel-wise layer normalization without affine parameters. In $\mathrm{MHA}(Q, K, V)$, Q supplies queries and (K, V) supply keys and values. Each $\mathcal{P}_X$ is a token-wise affine projection with weight P_X and bias b_X shared across tokens; $\mathcal{P}_E$, $\mathcal{P}_H$, and $\mathcal{P}_Q$ map to width d, and $\mathcal{P}_O$ maps to d_C; the projections and attention weights are collected in ψ.

The adapter output $R_{t,\tau}$ is injected into the action tokens through a bounded residual:

$$\begin{aligned} \widetilde{C}_{t,\tau} &= C_t + \Delta C_{t,\tau}, \qquad \Delta C_{t,\tau,n} = \alpha(\tau)\,\zeta(C_t)_n\,\tanh(R_{t,\tau,n}), \\ \alpha(\tau) &= \rho\, b(\tau), \qquad \zeta(C_t)_n = \mathrm{sg}[\max\{\mathrm{RMS}(C_{t,n}), \varepsilon_C\}]\,. \end{aligned} \tag{4}$$

Here $n = 1, \ldots, N_C$ indexes token rows, $\rho \geq 0$ controls feedback strength, $b : [0,1] \to [0,1]$ schedules it over flow time, and $\tanh$ acts elementwise. We use $\mathrm{RMS}(x) = \|x\|_2/\sqrt{d_C}$ for $x \in \mathbb{R}^{d_C}$, with $\varepsilon_C > 0$ setting a positive floor for the token scale. Each residual satisfies $\mathrm{RMS}(\Delta C_{t,\tau,n}) \leq \alpha(\tau)\zeta(C_t)_n$. The resulting geometry-conditioned action tokens $\widetilde{C}_{t,\tau}$ are termed the *Latent Future*. Zero-initialization details and full bounds are provided in Appendix A.1.

The action expert then conditions on the Latent Future:

$$v_\Theta(a_\tau, \tau; o_t, \ell, s_t) = v_\theta(a_\tau, \tau; \widetilde{C}_{t,\tau}, s_t), \quad \frac{\mathrm{d}\bar{a}_\tau}{\mathrm{d}\tau} = v_\Theta(\bar{a}_\tau, \tau; o_t, \ell, s_t), \quad \bar{a}_0 = \epsilon, \quad \hat{a} = \bar{a}_1. \tag{5}$$

Here $\Theta = (\vartheta, \theta, \phi, \psi)$. Action conditioning defines the action-to-world path, and bounded world residual feedback defines the world-to-action path while preserving the original flow schedule.

3.3 Euclidean-Hyperbolic Product Representation

Product representation. For J prediction horizons, let $\mathcal{R}_E$ be a fine regional grid and $\mathcal{R}_H$ collect regions from nested grids; the specific layout is given in Appendix C.1. With $N_E = J|\mathcal{R}_E|$ and $N_H = J|\mathcal{R}_H|$, we use the Poincaré ball of curvature $-c$ and the product space

$$\begin{aligned} \mathbb{B}_c^{d_H} &= \{x \in \mathbb{R}^{d_H} : c\|x\|_2^2 < 1\}, \qquad c > 0, \\ (\widehat{E}_{t,\tau}, \widehat{H}_{t,\tau}) &\in \mathcal{M} = (\mathbb{R}^{d_E})^{N_E} \times (\mathbb{B}_c^{d_H})^{N_H}. \end{aligned} \tag{6}$$

The Euclidean factor retains local future content, while the hyperbolic factor models relationships among predicted future regions. The product $\times$ is Cartesian; the two banks are fused only by the residual adapter in Equation 3. The Poincaré ball carries the metric $g_x^c = \lambda_c(x)^2 I_{d_H}$, where $\lambda_c(x) = 2/(1 - c\|x\|_2^2)$ and I_{d_H} is the identity matrix.

Scene-anchored future prediction. For a recorded clip $\mathcal{V}_t$, the frozen teacher provides target features $z_j^\star = \mathrm{sg}[E_{\mathrm{T}}(\mathcal{V}_t)_j]$ for $j = 1, \ldots, J$. Regional pooling, fixed normalization, and frozen projections produce Euclidean targets $e_{j,r}^\star$ for $r \in \mathcal{R}_E$ and tangent-space targets $u_{j,r}^\star$ for $r \in \mathcal{R}_H$. Applying the same transformations to z_t gives current-scene anchors $e_{t,r}$ and $u_{t,r}$. For $u \in T_0\mathbb{B}_c^{d_H} \cong \mathbb{R}^{d_H}$ and $x \in \mathbb{B}_c^{d_H}$, the origin maps are (Ganea et al., 2018)

$$\exp_0^c(u) = \frac{\tanh(\sqrt{c}\|u\|_2)}{\sqrt{c}\|u\|_2}u, \qquad \log_0^c(x) = \frac{\operatorname{artanh}(\sqrt{c}\|x\|_2)}{\sqrt{c}\|x\|_2}x, \tag{7}$$

with continuous extensions at zero. The predictor models changes relative to the current scene:

$$\begin{aligned} \hat{e}_{j,r} &= e_{t,r} + \Delta e_{j,r}, \qquad \hat{h}_{j,r} = \exp_0^c(\mathcal{T}(u_{t,r} + \Delta u_{j,r}))\,, \\ \mathcal{T}(u) &= \frac{R_{\max}}{\sqrt{d_H}}\tanh(u/\sigma_H), \qquad R_{\max}, \sigma_H > 0. \end{aligned} \tag{8}$$

Here $\Delta e_{j,r} \in \mathbb{R}^{d_E}$ and $\Delta u_{j,r} \in \mathbb{R}^{d_H}$ are predicted residuals; their dependence on (t, τ) is suppressed for readability. In the tangent-bounding transform $\mathcal{T}$, $\tanh$ is applied coordinatewise. The hyperbolic targets are $h_{j,r}^\star = \exp_0^c(\mathcal{T}(u_{j,r}^\star))$. Because $\|\mathcal{T}(u)\|_2 \leq R_{\max}$, predicted and target points stay in a compact subset strictly inside the ball, and $\|\log_0^c(\hat{h}_{j,r})\|_2 \leq R_{\max}$ (Appendix A.2). Target construction is detailed in Appendix C.2.

Hyperbolic relational structure. We index hyperbolic tokens by $i = (j, r)$ and write $\hat{h}_i$ and $h_i^\star$ for predicted and target points. Let r_0 denote the full-scene root region. The nested grids assign each non-root region r a unique parent $\mathrm{pa}(r)$ at the next coarser scale. Spatial and temporal edge sets define the supervised relational structure:

$$\begin{aligned} \mathcal{E}_S &= \{((j, r), (j, \mathrm{pa}(r))) : 1 \leq j \leq J,\ r \in \mathcal{R}_H \setminus \{r_0\}\}, \\ \mathcal{E}_T &= \{((j, r), (j+1, r)) : 1 \leq j < J,\ r \in \mathcal{R}_H\}. \end{aligned} \tag{9}$$

Spatial edges link nested regions within each prediction horizon, whereas temporal edges link the same fixed spatial region across adjacent horizons. Their lengths are measured by the Poincaré distance

$$d_c(x,y) = \frac{1}{\sqrt{c}} \operatorname{arcosh}\left(1 + \frac{2c\|x-y\|_2^2}{(1-c\|x\|_2^2)(1-c\|y\|_2^2)}\right). \tag{10}$$

The hierarchy is therefore encoded by the choice of supervised pairs rather than by an imposed radial ordering. Section 3.4 aligns the predicted edge lengths with their teacher counterparts.

3.4 Joint Action and Geometric World Objective

Action supervision and Euclidean feature alignment. Let $\mathcal{D}$ contain observations, instructions, optional states, demonstrated actions, recorded future clips, and validity masks. Expectations are taken over $\mathcal{D}$, independent noise ϵ, and flow time $\tau \sim p_\tau$. The action expert is trained by

$$\mathcal{L}_{\mathrm{act}} = \mathbb{E}\Big[\big\|v_\Theta(a_\tau, \tau; o_t, \ell, s_t) - u^{\mathrm{FM}}\big\|_\omega^2\Big], \qquad \|X\|_\omega^2 = \sum_{m=1}^{L}\sum_{k=1}^{d_a} \omega_{mk} X_{mk}^2, \tag{11}$$

where the nonnegative weights ω_{mk} incorporate action validity, coordinate weighting, and the configured normalization. Action supervision trains the action expert and can also update the residual adapter and world predictor through the Latent Future (Appendix A.1). Future observations supervise local predictive content through

$$\mathcal{L}_E = \mathbb{E}\big[\langle \kappa_{\beta_E}(\hat{e}_{j,r} - e^\star_{j,r})\rangle_{w^E}\big], \qquad \kappa_\beta(x) = \begin{cases} x^2/(2\beta), & |x| < \beta, \\ |x| - \beta/2, & |x| \geq \beta, \end{cases} \qquad \beta > 0. \tag{12}$$

Here $\beta_E > 0$ is the Euclidean loss threshold. For vector inputs, the smooth-L1 loss is applied coordinatewise and averaged over feature dimensions. For nonnegative weights, $\langle f\rangle_w = \sum_i w_i f_i / \sum_i w_i$ when $\sum_i w_i > 0$, and is zero otherwise; the index set is specified by each loss. The weights w^E combine validity and horizon weighting. All geometric targets are derived from the corresponding recorded future.

Hyperbolic point alignment and relation matching. The hyperbolic factor is supervised at both point and relation levels. Point alignment anchors each prediction to its teacher coordinate, while relation matching preserves the lengths of the spatial and temporal edges:

$$\begin{aligned} \mathcal{L}_{H,p} &= \mathbb{E}\Big[\langle d_c(\hat{h}_i, h_i^\star)^2\rangle_{w^H}\Big], \\ \mathcal{L}_{H,\nu} &= \mathbb{E}\Big[\big\langle \kappa_{\beta_R}\big(d_c(\hat{h}_i,\hat{h}_k) - d_c(h_i^\star, h_k^\star)\big)\big\rangle_{w^\nu}\Big], \quad \nu \in \{S,T\}, \\ \mathcal{L}_{H,r} &= \tfrac{1}{2}(\mathcal{L}_{H,S} + \mathcal{L}_{H,T}). \end{aligned} \tag{13}$$

The point average runs over valid tokens with horizon weights w^H, and each relation average runs over valid edges in $\mathcal{E}_\nu$ with weights w^ν; $\beta_R > 0$ is the relation-loss threshold. Spatial and temporal terms are normalized separately so that their relative contribution is not determined by edge counts. Point and relation supervision are complementary: exact point alignment implies exact alignment of supervised edge lengths, while relation matching directly penalizes finite structural distortion. Appendix A.3 formalizes this connection.

Joint optimization. The four supervision signals in Figure 2 define the joint objective

$$\mathcal{L}_{\mathrm{joint}} = \mathcal{L}_{\mathrm{act}} + \lambda_E \mathcal{L}_E + \lambda_{H,p}\mathcal{L}_{H,p} + \lambda_{H,r}\mathcal{L}_{H,r}, \tag{14}$$

where $\lambda_E, \lambda_{H,p}, \lambda_{H,r} \geq 0$ balance action supervision and the three geometric objectives. Specifically, $\mathcal{L}_{\mathrm{act}}$ supervises action generation, $\mathcal{L}_E$ aligns local future features, $\mathcal{L}_{H,p}$ anchors hyperbolic points, and $\mathcal{L}_{H,r}$ preserves spatial-temporal relations. The residual interface connects these predicted futures to control. Hyperbolic supervision is differentiated through the predictor's manifold maps and stabilized distance computations. Action supervision reaches the predictor through residual feedback, while the visual teacher and target transformations remain frozen. Together, these objectives couple action supervision with predictive fidelity and geometric consistency, allowing predicted futures to guide more accurate action generation.

Table 1: LIBERO success rates (%).

Method	Sp.	Obj.	Goal	Long	Avg.
UniVLA	96.5	96.8	95.6	92.0	95.2
OpenVLA	84.7	88.4	79.2	53.7	76.5
OpenVLA-OFT	97.6	98.4	97.9	94.5	97.1
WorldVLA	87.6	96.2	83.4	60.0	81.8
CogACT	97.2	98.0	90.2	88.8	93.6
MemoryVLA	98.4	98.4	96.4	93.4	96.7
ResVLA	96.0	**100.0**	97.4	92.8	96.6
Fast-WAM	98.2	**100.0**	97.0	95.2	97.6
π_0	96.8	98.8	95.8	85.2	94.2
$\pi_{0.5}$	98.8	98.2	98.0	92.4	96.9
JEPA-VLA	97.2	98.0	95.6	94.8	96.4
VLA-JEPA	96.2	99.6	97.2	95.8	97.2
Motus	96.8	99.8	96.6	**97.6**	97.7
GeoWRA	**99.0**	99.2	**98.2**	95.4	**98.0**

Table 2: LIBERO-Plus success rates (%).

Method	Cam.	Rob.	Lang.	Light	Bkg.	Noise	Lay.	Avg.
UniVLA	1.8	46.2	69.6	69.0	81.0	21.2	31.9	45.8
OpenVLA	0.8	3.5	23.0	8.1	34.8	15.2	28.5	16.3
OpenVLA-OFT	56.4	31.9	79.5	88.7	93.3	75.8	74.2	71.4
WorldVLA	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.6
RIPT-VLA	55.2	31.2	77.6	88.4	91.6	73.5	74.2	70.2
ResVLA	49.8	59.9	**88.5**	90.5	94.9	76.8	79.0	77.1
ABot-M0	60.4	67.9	86.4	96.2	91.6	86.4	82.6	81.6
π_0	13.8	6.0	58.8	85.0	81.4	79.0	68.9	56.1
π_0-Fast	65.1	21.6	61.0	73.2	73.2	74.4	68.8	62.5
JEPA-VLA	0.4	25.7	54.4	38.0	23.9	4.1	32.8	25.6
Cosmos-Policy	**75.8**	63.3	81.7	96.5	88.9	**92.7**	82.2	83.0
$\pi_{0.5}$	69.4	75.3	82.6	96.7	96.8	84.3	86.2	84.5
VLA-JEPA	63.3	67.1	85.4	95.6	93.6	66.3	85.1	79.5
GeoWRA	73.1	**76.3**	85.6	**97.5**	**97.3**	86.7	**87.0**	**86.2**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP AND RESULTS

Benchmarks. We evaluate GeoWRA on three simulation benchmarks, LIBERO (Liu et al., 2023), LIBERO-Plus (Fei et al., 2025), and SimplerEnv (Li et al., 2024c). LIBERO comprises four suites (Spatial, Object, Goal, and Long), totaling 2,000 episodes, while LIBERO-Plus assesses robustness across seven perturbation categories over 10,030 episodes. SimplerEnv covers four task families each for Google and WidowX with 864 and 96 evaluation episodes, respectively.

Baselines. We organize the comparison methods into two categories: *VLA-based policies* and *world-model-based methods*. The VLA-based category includes Octo-Base (Team et al., 2024), RT-1-X (O'Neill et al., 2024), OpenVLA (Kim et al., 2024), OpenVLA-OFT (Kim et al., 2025), π_0 (Black et al., 2024), π_0-Fast (Pertsch et al., 2025), $\pi_{0.5}$ (Intelligence et al., 2025), StarVLA (Community, 2026), UniVLA (Bu et al., 2025), villa-X (Chen et al., 2026), TraceVLA (Zheng et al., 2025a), Magma (Yang et al., 2025), RoboVLMs (Li et al., 2024b), CogACT (Li et al., 2024a), MemoryVLA (Shi et al., 2026), ResVLA (Zhong et al., 2026), RIPT-VLA (Tan et al., 2025), and ABot-M0 (Yang et al., 2026b). The world-model-based category comprises JEPA-VLA (Miao et al., 2026), WorldVLA (Cen et al., 2025), VLA-JEPA (Sun et al., 2026), Fast-WAM (Yuan et al., 2026), Motus (Bi et al., 2026), and Cosmos-Policy (Kim et al., 2026). To evaluate the generality of GeoWRA across different VLA backbones, we instantiate it with a Qwen-based VLA on SimplerEnv and with $\pi_{0.5}$ on LIBERO and LIBERO-Plus; implementation details are provided in Appendix C.

Performance in Simulated Environments. On LIBERO and LIBERO-Plus, GeoWRA achieves the highest average success rates among the compared methods, reaching 98.0% and 86.2%, respectively (Tables 1 and 2). *Compared with VLA-based policies*, GeoWRA outperforms $\pi_{0.5}$ and OpenVLA-OFT, with gains of 1.1 and 1.7 percentage points over $\pi_{0.5}$, respectively. *Compared with world-model-based approaches*, GeoWRA also surpasses WorldVLA and VLA-JEPA, exceeding the latter by 0.8 and 6.7 percentage points, respectively. GeoWRA achieves the best Spatial and Goal results on LIBERO and exceeds $\pi_{0.5}$ across all seven LIBERO-Plus perturbation categories, demonstrating strong manipulation performance and robustness to environmental variations.

On SimplerEnv, GeoWRA achieves 69.1% on Google and 71.9% on WidowX, with an overall average of 70.5%, the highest among the compared methods reporting all eight task families (Table 3). *GeoWRA surpasses VLA-based policies*, including $\pi_{0.5}$, OpenVLA-OFT, and StarVLA, with an overall gain of 14.1 percentage points over our matched-training StarVLA baseline. *Compared with world-model-based method* VLA-JEPA, GeoWRA improves success rates by 3.9 and 14.6 percentage points on Google and WidowX, respectively, and by 9.2 points overall. GeoWRA also achieves the best results on six of eight tasks. These advantages over both categories support the effectiveness of geometric world-action modeling for robotic manipulation.

Table 3: Success rates (%) in SimplerEnv. Overall is the arithmetic mean of the two reported robot-level averages. Best and second-best results are highlighted in bold and underlined, respectively.

	Google Robot					WidowX Robot					Overall
Method	Pick	Move	Drawer	Place	Avg.	Spoon	Carrot	Block	Eggplant	Avg.	
Octo-Base	17.0	4.2	22.7	0.0	11.0	12.5	8.3	0.0	43.1	16.0	13.5
OpenVLA	18.0	56.3	63.0	0.0	34.3	4.2	0.0	0.0	12.5	4.2	19.3
RT-1-X	56.7	31.7	59.7	21.3	42.4	0.0	4.2	0.0	0.0	1.1	21.8
villa-X	81.7	55.4	38.4	4.2	44.9	48.3	24.2	19.2	71.7	40.9	42.9
TraceVLA	45.0	63.8	63.1	11.1	45.8	12.5	16.6	16.6	65.0	27.7	36.8
Magma	75.0	53.0	58.9	8.3	48.8	37.5	29.2	20.8	91.7	44.8	46.8
RoboVLMs	77.3	61.7	43.5	24.1	51.7	45.8	20.8	4.2	79.2	37.5	44.6
StarVLA	80.0	62.1	48.6	39.8	57.6	70.8	45.8	20.8	83.3	55.2	56.4
π_0	72.7	65.3	38.3	-	58.8	29.1	0.0	16.6	62.5	27.1	43.0
$\pi_{0.5}$	70.3	68.8	**63.5**	32.4	58.8	37.5	45.8	20.8	67.5	42.9	50.9
OpenVLA-OFT	72.3	69.6	47.2	-	63.0	34.2	30.0	30.0	72.5	41.7	52.4
VLA-JEPA	88.3	64.1	59.3	49.1	65.2	75.0	**70.8**	12.5	70.8	57.3	61.3
GeoWRA (ours)	**94.0**	**72.1**	55.6	**54.6**	**69.1**	**83.3**	58.3	**45.8**	**100.0**	**71.9**	**70.5**

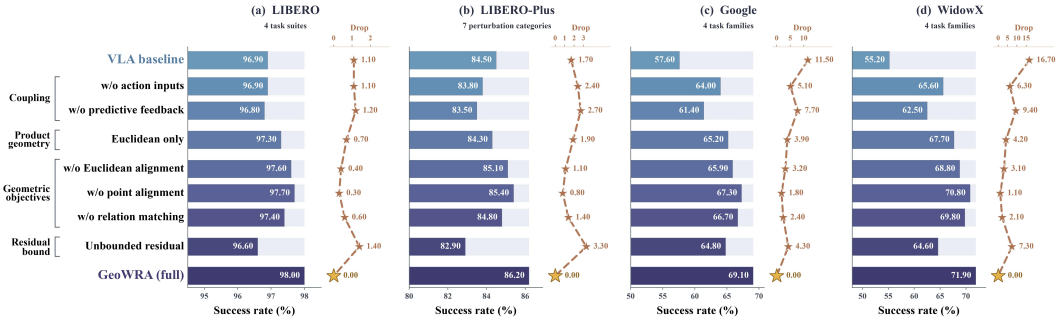


Figure 3: Component ablations on LIBERO, LIBERO-Plus, Google, and WidowX. Colored bars show success rates, gray bars mark the full-model reference, and stars indicate drops in percentage points. Rows show separate ablations; success-rate axes are truncated.

4.2 Ablation Study

In-depth analysis of the GeoWRA architecture. We conduct an analysis of GeoWRA's components, including bidirectional coupling, geometric representation and supervision, and bounded world residual feedback. As shown in Figure 3, the ablations support three observations.
Observation 1: Both directions of coupling contribute to policy performance. Removing action condition inputs from the world predictor reduces average success rates by 1.75 percentage points across LIBERO and LIBERO-Plus and by 5.70 points on SimplerEnv, while removing predictive feedback yields corresponding drops of 1.95 and 8.55 points. The larger drops without feedback highlight the importance of directly translating predicted consequences into action guidance.
Observation 2: Geometric structure and alignment improve policy performance. The Euclidean-only variant lowers performance by 1.30 and 4.05 points, respectively. Removing any geometric objective also degrades performance, suggesting that product geometry and explicit alignment help learn predictive representations that better support action generation.
Observation 3: Predictive feedback is more effective when its magnitude is bounded. Removing the residual bound causes drops of 2.35 and 5.80 points. This suggests that effective integration depends on regulating how world predictions influence the policy, supporting bounded world residual feedback as a mechanism for incorporating predictive information through controlled action corrections.

Sensitivity to geometric loss weights. The geometric loss weights $(\lambda_E, \lambda_{H,p}, \lambda_{H,r})$ are set to $(0.05, 0.02, 0.01)$ by default. Figure 4 visualizes a $3 \times 3 \times 4$ parameter grid on SimplerEnv, with $\lambda_E \in \{0.025, 0.05, 0.10\}$, $\lambda_{H,p} \in \{0.01, 0.02, 0.04\}$, and $\lambda_{H,r} \in \{0, 0.005, 0.01, 0.02\}$. This grid includes the shared default and six configurations obtained by halving or doubling one coefficient while fixing the other two. The outlined default uses the main-result reference of 70.5%, computed

Table 4: Real-world manipulation results, reported as successful trials out of 40 per task. Total aggregates all six tasks (240 trials). Bold numerators indicate the best results.

Method	Moving object interception	Unstable stack transfer	Obstacle aware placement	Large on small stacking	Order aware bowl packing	Bowl to bowl block pouring	**Total**
OpenVLA	9/40	7/40	14/40	5/40	11/40	6/40	52/240
π_0	13/40	10/40	17/40	9/40	14/40	10/40	73/240
$\pi_{0.5}$	18/40	15/40	22/40	12/40	17/40	13/40	97/240
VLA-JEPA	23/40	18/40	21/40	17/40	19/40	16/40	114/240
GeoWRA (ours)	**27**/40	**25**/40	**26**/40	**21**/40	**22**/40	**21**/40	**142**/240

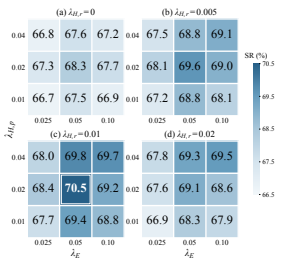


Figure 4: Joint sensitivity analysis of geometric loss coefficients on SimplerEnv benchmark.

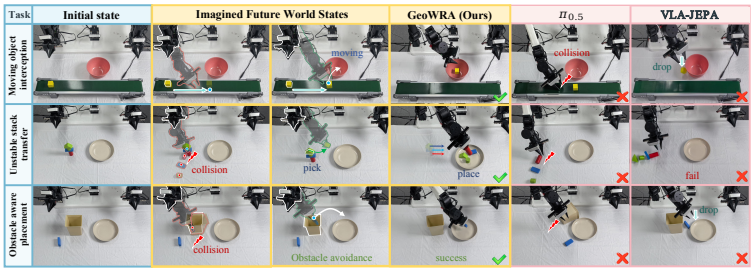


Figure 5: Real-world task comparisons. Each row shows the initial state, imagined future world states, and representative successful and unsuccessful executions. The schematic outcomes illustrate task consequences rather than decoded model outputs.

as the arithmetic mean of the Google Robot and WidowX Robot averages of 69.1% and 71.9%, respectively (Table 3). Within the illustrated one-factor subset, success rates are 0.7-2.1 percentage points below the default reference, with the largest difference occurring at $\lambda_E = 0.025$.

4.3 Real-World Experiments

We evaluate GeoWRA on the Cobot Mobile ALOHA platform (Fu et al., 2024), which comprises two 6-DoF robotic arms, two wrist-mounted RGB cameras, and one third-person RGB camera. The evaluation covers six tasks: moving-object interception, unstable-stack transfer, obstacle-aware placement, large-on-small stacking, order-aware bowl packing, and bowl-to-bowl block pouring. We compare GeoWRA with OpenVLA, π_0, $\pi_{0.5}$, and VLA-JEPA, evaluating each policy over 40 trials per task. As shown in Table 4, GeoWRA achieves the highest success count on all six tasks, completing 142 of 240 trials. This corresponds to an average success rate of 59.2%, exceeding VLA-JEPA by 11.7 percentage points. Figure 5 summarizes qualitative comparisons for the different tasks by showing the initial states, imagined future world states, successful GeoWRA executions, and representative baseline failures. See Appendix C.3 for more details.

5 Conclusion

This paper presents GeoWRA, a unified framework that couples latent world prediction with VLA action generation. GeoWRA establishes a bidirectionally coupled architecture in which latent actions condition future prediction and predicted futures guide action generation through bounded world residual feedback. A Euclidean-hyperbolic product representation models local future features and their relational structure, while a joint action and geometric world objective integrates action supervision with feature, point, and relation alignment. Experiments in simulation and real-world settings demonstrate competitive performance and robustness across diverse manipulation tasks. We hope GeoWRA encourages further research on unified world modeling and action generation for embodied intelligence.

AI USE STATEMENT

We used generative AI tools to assist with language editing, code implementation, and consistency checks of mathematical notation and derivations. The authors reviewed and revised the AI-assisted text and code and checked the resulting changes for correctness and consistency. We take full responsibility for the final content of this work, including its text, code, mathematical claims, and AI-assisted artifacts.

REFERENCES

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, 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.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*, 2024.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

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

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32, 2019.

Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, et al. Villa-x: enhancing latent action modeling in vision-language-action models. In *International Conference on Learning Representations*, volume 2026, pp. 70673–70703, 2026.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Jingtao Ding, Yunke Zhang, Yu Shang, Yuheng Zhang, Zefang Zong, Jie Feng, Yuan Yuan, Hongyuan Su, Nian Li, Nicholas Sukiennik, et al. Understanding world or predicting future? a comprehensive survey of world models. *ACM Computing Surveys*, 58(3):1–38, 2025.

Heng Dong, Junyu Zhang, and Chongjie Zhang. Leveraging hyperbolic embeddings for coarse-to-fine robot design. In *International Conference on Learning Representations*, volume 2024, pp. 5160–5187, 2024.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018.

Yanjiang Guo, Lucy Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 6121–6138, 2026.

Bohan Hou, Gen Li, Jindou Jia, Tuo An, Xinying Guo, Sicong Leng, Haoran Geng, Yanjie Ze, Tatsuya Harada, Philip Torr, et al. World model for robot learning: A comprehensive survey. *arXiv preprint arXiv:2605.00080*, 2026.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

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

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, et al. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024a.

Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 3(7):8, 2024b.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024c.

Yihan Lin, Jiawei He, Shifeng Bao, Chen Zhao, Yang Li, Xiaobo Wang, Yan Wang, Cheng Chi, and Jing Zhang. Jepa-wam: Learning vision-language-action policies with joint-embedding world modeling. *arXiv preprint arXiv:2608.09381*, 2026.

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

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. *Advances in neural information processing systems*, 32, 2019.

Yuwen Liu, Lianyong Qi, Xingyuan Mao, Weiming Liu, Xuhui Fan, Qiang Ni, Xuyun Zhang, Yang Zhang, Yuan Tian, and Amin Beheshti. Hyperbolic-enhanced mixture-of-experts mamba for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15403–15411, 2026.

Hongbo Lu, Liang Yao, Chenghao He, Haoyu Wang, Xiang Gu, Xianfei Li, Wenlong Liao, Tao He, and Pai Peng. The dawn of world-action interactive models. *arXiv preprint arXiv:2605.11550*, 2026.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Shangchen Miao, Ningya Feng, Jialong Wu, Ye Lin, Xu He, Dong Li, and Mingsheng Long. Jepa-vla: Video predictive embedding is needed for vla models. *arXiv preprint arXiv:2602.11832*, 2026.

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

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

Zelin Peng, Zhengqin Xu, Qingyang Liu, Xiaokang Yang, and Wei Shen. Hyperet: Efficient training in hyperbolic space for multi-modal large language models. *Advances in neural information processing systems*, 38:8025–8048, 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. *arXiv preprint arXiv:2501.09747*, 2025.

Yichao Shen, Fangyun Wei, Zhiying Du, Yaobo Liang, Yan Lu, Jiaolong Yang, Nanning Zheng, and Baining Guo. Videovla: Video generators can be generalizable robot manipulators. *Advances in neural information processing systems*, 38:95597–95621, 2026.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 18567–18602, 2026.

Mingchen Song, Xiang Deng, Jie Wei, Dongmei Jiang, Liqiang Nie, and Weili Guan. Ensemblevla: Ensemble learning for vision-language action models. In *Forty-third International Conference on Machine Learning*, 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 *European Conference on Computer Vision*, pp. 478–497. Springer, 2026.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models. *arXiv preprint arXiv:2505.17016*, 2025.

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

Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 92033–92052, 2025.

Kun Wang, Xiao Feng, Mingcheng Qu, and Tonghua Su. Hmvla: Hyperbolic multimodal fusion for vision-language-action models. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 12557–12561. IEEE, 2026a.

Siyin Wang, Junhao Shi, Zhaoyang Fu, Xinzhe He, Feihong Liu, Chenchen Yang, Yikang Zhou, Zhaoye Fei, Jingjing Gong, Jinlan Fu, et al. World action models: The next frontier in embodied ai. *arXiv preprint arXiv:2605.12090*, 2026b.

Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, et al. Magma: A foundation model for multimodal ai agents. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14203–14214. IEEE, 2025.

Jin Yang, Ping Wei, Yixin Chen, and Nanning Zheng. Hyperbolic multiview pretraining for robotic manipulation. *arXiv preprint arXiv:2603.04848*, 2026a.

Menglin Yang, Min Zhou, Tong Zhang, Jiahong Liu, Zhihao Li, Lujia Pan, Hui Xiong, and Irwin King. Hyperbolic graph neural networks: A review of methods and applications. *arXiv preprint arXiv:2202.13852*, 2022.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, et al. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026b.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 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. *arXiv preprint arXiv:2510.10642*, 2025.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *Advances in Neural Information Processing Systems*, 38: 24195–24228, 2026a.

Zeyu Zhang, Danning Li, Ian Reid, and Richard Hartley. Geoworld: Geometric world models. *arXiv preprint arXiv:2602.23058*, 2026b.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *International Conference on Learning Representations*, volume 2025, pp. 54277–54296, 2025a.

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

Yiming Zhong, Yaoyu He, Zemin Yang, Pengfei Tian, Yifan Huang, Qingqiu Huang, Xinge Zhu, and Yuexin Ma. From noise to intent: Anchoring generative vla policies with residual bridges. *arXiv preprint arXiv:2604.21391*, 2026.

Fangqi Zhu, Zhengyang Yan, Zicong Hong, Quanxin Shou, Xiao Ma, and Song Guo. Wmpo: World model-based policy optimization for vision-language-action models. In *International Conference on Learning Representations*, volume 2026, pp. 62486–62502, 2026.

APPENDIX

This appendix provides theoretical support, supplementary experiments, and implementation details for the main paper. The following guide summarizes the purpose of each part.

1. Properties of Geometric Residual Coupling: establishes feedback bounds and the compatibility of geometric supervision.
2. Choice of Geometric Factors: evaluates the composition of the prediction branches.
3. Sensitivity to Prediction Layout: evaluates spatial scales and prediction horizons.
4. Alternative Feature Fusion: compares interfaces for combining the two prediction banks.
5. Residual Adapters and Future Content: examines how the policy reads and uses predicted futures.
6. Additional VLA Backbone: evaluates geometric feedback with OpenVLA-OFT.
7. Action-Conditioned Future Prediction: evaluates the contribution of action conditioning to prediction accuracy.
8. Architecture and Residual Mechanism: specifies the prediction architecture and residual feedback.
9. Teacher Targets and Numerical Geometry: explains how feature and relation supervision are constructed.
10. Real-World Evaluation: documents the platform, task criteria, and experimental outcomes.
11. Training and Evaluation Settings: summarizes training configurations, hardware, and evaluation protocols.

A PROPERTIES OF GEOMETRIC RESIDUAL COUPLING

This section supplements the architecture, representation, and objectives in Section 3. Appendix A.1 analyzes the residual interface and flow-matching generation; Appendix A.2 establishes properties of the bounded hyperbolic factor; and Appendix A.3 relates point and relation supervision. Model parameters, current inputs, and the teacher reference are fixed throughout each comparison. Geometric statements concern the exact maps; Appendix C.2 describes numerical evaluation.

A.1 BOUNDED FEEDBACK AND ACTION-FLOW PERTURBATIONS

Proposition A.1 (Scale-controlled conditioning feedback)**.** Let $C, R \in \mathbb{R}^{N_C \times d_C}$, $\alpha \geq 0$, and $\varepsilon_C > 0$. Using the scale in Equation 4, define

$$\begin{aligned} \zeta(C)_n &= \mathrm{sg}[\max\{\mathrm{RMS}(C_n), \varepsilon_C\}], \\ S_{\varepsilon_C}(C) &= \left(d_C \sum_{n=1}^{N_C} \zeta(C)_n^2 \right)^{1/2}. \end{aligned} \tag{15}$$

For $\Delta C_n = \alpha \zeta(C)_n \tanh(R_n)$,

$$\mathrm{RMS}(\Delta C_n) \leq \alpha \zeta(C)_n, \qquad \|\Delta C\|_F \leq \alpha S_{\varepsilon_C}(C). \tag{16}$$

When the scale floor is inactive for every token, $S_{\varepsilon_C}(C) = \|C\|_F$. For the adapter in Equation 3, $P_O = 0$ and $b_O = 0$ imply $\Delta C = 0$, recovering the same action expert evaluated with the original action tokens.

Proof. Using $|\tanh z| \leq 1$ gives

$$\mathrm{RMS}(\Delta C_n)^2 = \frac{\alpha^2 \zeta(C)_n^2}{d_C} \sum_{k=1}^{d_C} \tanh^2(R_{nk}) \leq \alpha^2 \zeta(C)_n^2. \tag{17}$$

Multiplying by d_C and summing over n gives the Frobenius bound. If the floor is inactive, then $d_C \zeta(C)_n^2 = \|C_n\|_2^2$, yielding $S_{\varepsilon_C}(C) = \|C\|_F$. If $P_O = 0$ and $b_O = 0$, the adapter output is $R = 0$; hence $\widetilde{C} = C$. Stop-gradient affects derivatives but leaves these forward identities unchanged. □

The selected configuration uses $\rho = 0.2$, $b(\tau) = 0.25 + 0.75\tau$, and $\varepsilon_C = 10^{-3}$. Thus $\|\Delta C_{t,\tau}\|_F \leq 0.2 S_{\varepsilon_C}(C_t)$, controlling the magnitude of the conditioning perturbation.

The following comparison uses the flow-matching formulation of Section 3.1 (Lipman et al., 2022). It concerns flow-matching action experts; the feedback bound above does not require a particular action-generation solver.

Proposition A.2 (Conditional perturbation bound for the action flow)**.** Fix the action tokens $C = C_t$, action-expert parameters θ, initial noise ϵ, and optional proprioceptive state s_t. We suppress the fixed state argument and write $v_\theta(x, \tau; C)$ for $v_\theta(x, \tau; C, s_t)$. Let the uncorrected and corrected flows be absolutely continuous on $[0, 1]$ and satisfy the following differential equations almost everywhere:

$$\begin{aligned} \dot{a}^0_\tau &= v_\theta(a^0_\tau, \tau; C), \\ \dot{a}^G_\tau &= v_\theta(a^G_\tau, \tau; C + \Delta C_\tau), \\ a^0_0 &= a^G_0 = \epsilon. \end{aligned} \tag{18}$$

Here $\Delta C_\tau = \Delta C_{t,\tau}(a^G_\tau)$ denotes the residual evaluated along the corrected action trajectory. Assume that the expert is Lipschitz in its action and conditioning arguments on a domain containing the two trajectories and their evaluated conditions: for nonnegative integrable $L_a(\tau)$ and $L_C(\tau)$, the following inequality holds for almost every $\tau \in [0, 1]$:

$$\|v_\theta(x, \tau; C_1) - v_\theta(y, \tau; C_2)\|_F \leq L_a(\tau)\|x - y\|_F + L_C(\tau)\|C_1 - C_2\|_F. \tag{19}$$

If $\|\Delta C_\tau\|_F \leq \alpha(\tau) S_{\varepsilon_C}(C)$ for almost every τ and $L_C\alpha$ is integrable, then

$$\|a^G_\tau - a^0_\tau\|_F \leq S_{\varepsilon_C}(C) \int_0^\tau \exp\left(\int_s^\tau L_a(u)\,\mathrm{d}u\right) L_C(s)\alpha(s)\,\mathrm{d}s. \tag{20}$$

Proof. Let $e(\tau) = \|a^G_\tau - a^0_\tau\|_F$. As the norm of an absolutely continuous function, e is absolutely continuous and satisfies $e'(\tau) \leq \|\dot{a}^G_\tau - \dot{a}^0_\tau\|_F$ for almost every τ. Adding and subtracting $v_\theta(a^G_\tau, \tau; C)$ in the velocity difference, Equation 19 and Proposition A.1 imply, for almost every $\tau \in [0, 1]$,

$$e'(\tau) \leq L_a(\tau)e(\tau) + L_C(\tau)\alpha(\tau)S_{\varepsilon_C}(C), \qquad e(0) = 0. \tag{21}$$

Multiplying by the integrating factor $\exp(-\int_0^\tau L_a(u)\,\mathrm{d}u)$ and integrating yields Equation 20. □

Under the stated regularity assumptions, this result bounds the deviation induced by residual feedback during a single action-generation call, relative to the same action expert without feedback. Its effect on task performance is assessed in the experiments.

A discrete integration counterpart. For explicit Euler steps $0 = \tau_0 < \cdots < \tau_K = 1$, let $h_k = \tau_{k+1} - \tau_k$ and $e_k = \|a^G_k - a^0_k\|_F$. Assume that the residual bound and Equation 19 also hold at every discrete flow time and evaluated condition of the two trajectories. At step k, the corrected update uses ΔC_{τ_k}. Applying the same decomposition to the two discrete updates gives

$$\begin{aligned} e_{k+1} &\leq (1 + h_k L_a(\tau_k))e_k + h_k L_C(\tau_k)\alpha(\tau_k)S_{\varepsilon_C}(C), \\ e_K &\leq S_{\varepsilon_C}(C) \sum_{k=0}^{K-1} h_k L_C(\tau_k)\alpha(\tau_k) \prod_{j=k+1}^{K-1} (1 + h_j L_a(\tau_j)). \end{aligned} \tag{22}$$

The empty product equals one. This follows by induction from $e_0 = 0$ and is an Euler-specific result; other solvers require their own discrete analysis.

The gradient pathway. Let $Z_{\phi,t,\tau} = U_{t,\tau}$ denote the projected geometric context in Equation 3, including the affine projections $\mathcal{P}_E$ and $\mathcal{P}_H$. At fixed inputs $(a_\tau, \tau, o_t, \ell, s_t)$ and fixed parameters $(\vartheta, \theta, \psi)$, the world-predictor parameters ϕ influence the action expert through the Latent Future:

$$\frac{\partial v_\Theta}{\partial \phi} = \frac{\partial v_\theta}{\partial \widetilde{C}_{t,\tau}} \frac{\partial \Delta C_{t,\tau}}{\partial Z_{\phi,t,\tau}} \frac{\partial Z_{\phi,t,\tau}}{\partial \phi}. \tag{23}$$

Jacobians are taken after vectorizing their matrix arguments. This is the derivative of a single velocity evaluation with respect to the world-predictor parameters. At initialization, $P_O = 0$ and $b_O = 0$, so this pathway is initially zero. Geometric supervision still trains the predictor, and the action-gradient pathway can become nonzero as P_O learns.

A.2 A Finite Geometric Domain and Bounded Readout

We use the Poincaré-ball metric underlying Poincaré embeddings (Nickel & Kiela, 2017) and the exponential and logarithmic maps used in hyperbolic neural networks (Ganea et al., 2018). The curvature and metric normalization follow Section 3.3. The next proposition derives bounds for the specific transform $\mathcal{T}$ used by GeoWRA.

Proposition A.3 (Compact hyperbolic predictions and tangent readout)**.** Let $c, R_{\max}, \sigma_H > 0$ and define $h(u) = \exp_0^c(\mathcal{T}(u))$ using Equation 8. Set

$$r_c = \frac{\tanh(\sqrt{c}R_{\max})}{\sqrt{c}}, \qquad \mathcal{K}_c = \{x \in \mathbb{R}^{d_H} : \|x\|_2 \le r_c\}. \tag{24}$$

Every prediction and teacher target of this form belongs to the compact set $\mathcal{K}_c \subset \mathbb{B}_c^{d_H}$. For $x, y \in \mathcal{K}_c$,

$$\begin{gathered} 1 - c\|x\|_2^2 \ge \operatorname{sech}^2(\sqrt{c}R_{\max}) > 0, \qquad \|\log_0^c(x)\|_2 \le R_{\max}, \\ d_c(0,x) \le 2R_{\max}, \qquad d_c(x,y) \le 4R_{\max}. \end{gathered} \tag{25}$$

Here $\operatorname{sech}(z) = 1/\cosh(z)$. Furthermore,

$$2\|x-y\|_2 \le d_c(x,y) \le 2\cosh^2(\sqrt{c}R_{\max})\|x-y\|_2. \tag{26}$$

Proof. Elementwise boundedness of $\tanh$ yields $\|\mathcal{T}(u)\|_2^2 \le R_{\max}^2$. Taking the norm of the exponential map gives

$$\sqrt{c}\|h(u)\|_2 = \tanh(\sqrt{c}\|\mathcal{T}(u)\|_2) \le \tanh(\sqrt{c}R_{\max}) < 1. \tag{27}$$

Thus $h(u) \in \mathcal{K}_c$, and $\mathcal{K}_c$ is closed and bounded with radius strictly below $1/\sqrt{c}$. The identity $1 - \tanh^2 z = \operatorname{sech}^2 z$ proves the boundary margin. The logarithmic-map norm is $\operatorname{artanh}(\sqrt{c}\|x\|_2)/\sqrt{c}$, giving the tangent bound. Under the metric convention in Section 3.3,

$$d_c(0, \exp_0^c(v)) = 2\|v\|_2, \qquad \log_0^c(\exp_0^c(v)) = v. \tag{28}$$

The origin-distance bound follows, and the triangle inequality yields the diameter bound. Finally, the conformal factor satisfies $\lambda_c(z) \ge 2$ throughout the ball, so every path from x to y has length at least $2\|x-y\|_2$. The Euclidean segment between x and y stays in the convex set $\mathcal{K}_c$, where $\lambda_c(z) \le 2/(1-cr_c^2) = 2\cosh^2(\sqrt{c}R_{\max})$. Integrating its metric length gives the upper bound in Equation 26. □

The result links the geometric domain to feedback: logarithmic readouts have a bounded input scale, and distance denominators stay away from zero. Since $\log_0^c(\hat{h}_{j,r}) = \mathcal{T}(u_{t,r} + \Delta u_{j,r})$, the adapter reads tangent-space features; hyperbolic geometry shapes these features through point and relation supervision. It concerns the hyperbolic factor, not the entire product space, whose Euclidean factor is unrestricted. Numerical clipping and stabilized evaluations are still required in finite precision. In particular, the unsquared distance is nondifferentiable at coincident points, and the implementation uses stabilized distance computations, while squared point distance has a smooth geometric extension there.

A.3 Compatibility of Point and Relation Supervision

Proposition A.4 (Endpoint control of relational distortion)**.** Let $\delta_i = d_c(\hat{h}_i, h_i^\star)$ and define the signed edge discrepancy $\xi_{ik} = d_c(\hat{h}_i, \hat{h}_k) - d_c(h_i^\star, h_k^\star)$. For every spatial or temporal edge,

$$|\xi_{ik}| \le \delta_i + \delta_k. \tag{29}$$

For a nonempty valid edge set $\mathcal{E}$ with normalized nonnegative weights p_{ik}, define the weighted incidence $\gamma_i^{\mathcal{E}} = \sum_{k:(i,k)\in\mathcal{E}} p_{ik} + \sum_{k:(k,i)\in\mathcal{E}} p_{ki}$. Then the smooth-L1 convention of Equation 12 satisfies

$$\sum_{(i,k)\in\mathcal{E}} p_{ik}\kappa_\beta(\xi_{ik}) \le \frac{1}{\beta}\sum_i \gamma_i^{\mathcal{E}}\delta_i^2, \qquad \beta > 0. \tag{30}$$

Proof. Applying the triangle inequality through the two target points gives

$$d_c(\hat{h}_i, \hat{h}_k) \le \delta_i + d_c(h_i^\star, h_k^\star) + \delta_k. \tag{31}$$

Exchanging predicted and target points yields the reverse difference and proves Equation 29. For all real z, $\kappa_\beta(z) \le z^2/(2\beta)$: this is equality in the quadratic regime, while outside it the gap equals $(|z| - \beta)^2/(2\beta) \ge 0$. Therefore

$$\kappa_\beta(\xi_{ik}) \le \frac{(\delta_i + \delta_k)^2}{2\beta} \le \frac{\delta_i^2 + \delta_k^2}{\beta}. \tag{32}$$

Multiplying by p_{ik} and summing proves Equation 30. If there are no valid edges, the loss is defined to be zero. Applying this argument separately to the spatial and temporal sets preserves the averaging convention of Equation 13. □

Exact point alignment implies exact relation alignment on every supervised edge. At finite error, relation supervision directly penalizes edge-length discrepancies, with each endpoint contributing through its weighted incidence. These incidence weights need not equal the point-loss weights w^H; the bound establishes compatibility without identifying the two objectives. The converse does not hold: a common sign reversal preserves hyperbolic distances and remains realizable because $\mathcal{T}$ and $\exp_0^c$ are odd maps, while it may change the target coordinates. Point supervision therefore fixes the teacher reference, while relation supervision constrains the structure of connected predictions. The task-level effects of point and relation supervision are evaluated empirically in Figure 3.

B Supplementary Experimental Analyses

This section provides additional analyses of geometric representations, prediction layouts, and predictive feedback, together with evaluations of backbone integration and computational cost. These studies complement the main experiments and examine the design choices underlying GeoWRA.

B.1 Choice of Geometric Factors

Table 5 compares prediction-branch configurations. The Euclidean-only variant removes the hyperbolic branch and its associated losses, retaining Euclidean predictions for residual feedback. Both the Euclidean-Euclidean and hyperbolic-hyperbolic variants retain the two prediction banks with 48 and 63 tokens, their spatial layouts, three prediction horizons, 64-dimensional factors, and predictor widths and depths. The Euclidean-Euclidean variant leaves the first branch unchanged and replaces the hyperbolic branch with bounded Euclidean coordinates $\hat{v}_{j,r} = \mathcal{T}(u_{t,r} + \Delta u_{j,r})$ and $v_{j,r}^\star = \mathcal{T}(u_{j,r}^\star)$, retaining the frozen projection A_H and transform $\mathcal{T}$. Its point and relation objectives replace hyperbolic distances with $d_{\mathrm{flat}}(v, w) = 2\|v - w\|_2$, consistent with the $c \to 0$ limit of our metric convention, while preserving the supervised edges, loss forms, validity weights, and normalization. The adapter reads $\mathcal{P}_H(\hat{v}_{j,r})$, matching the full model's tangent-space readout. The hyperbolic-hyperbolic variant uses targets and losses consistent with each factor, with curvature -0.5 and tangent-space readout. All variants use the same training data and training budget. GeoWRA achieves the highest average success rate of 70.5%, exceeding the Euclidean-only, Euclidean-Euclidean, and hyperbolic-hyperbolic variants by 4.05, 2.70, and 1.50 percentage points, respectively. These results demonstrate the effectiveness of joint Euclidean-hyperbolic modeling.

B.2 Sensitivity to Prediction Layout

We examine the contribution of intermediate spatial and temporal predictions by removing either the 2×2 hyperbolic grid or the middle prediction horizon from the default layout. The fewer-scales variant links each 4×4 region directly to the full-scene region. The fewer-horizons variant retains the original earliest and latest targets and their action summaries $\bar{q}_j$ and $\bar{q}_{\le j}$ from Equation 33. As shown in Table 6, these variants achieve average success rates of 69.0% and 69.9%, respectively, compared with 70.5% for the default layout. The corresponding decreases of 1.55 and 0.65 percentage points, with lower success rates on both robots, support retaining intermediate spatial and temporal predictions for action generation.

Table 5: Geometric representations results.

Representation	Google	WidowX	Avg.
Euclidean-only	65.2	67.7	66.5
Euclidean-Euclidean	66.8	68.8	67.8
Hyperbolic-Hyperbolic	67.2	70.8	69.0
GeoWRA	**69.1**	**71.9**	**70.5**

Table 6: Prediction layouts: success rates (%).

Layout	Google	WidowX	Avg.
StarVLA (baseline)	57.6	55.2	56.4
Fewer scales	68.1	69.8	69.0
Fewer horizons	68.9	70.8	69.9
Default	**69.1**	**71.9**	**70.5**

Table 7: Feature-fusion: success rates (%).

Fusion	Google	WidowX	Avg.
Concatenation	67.3	68.8	68.1
Averaging	65.3	66.7	66.0
Gated addition	68.1	70.8	69.5
Multiplication	64.9	67.7	66.3
Native attention	**69.1**	**71.9**	**70.5**

Table 8: Residual adapters: success rates (%).

Adapter	Google	WidowX	Avg.
Pooled MLP	65.4	67.7	66.6
Token-mixing MLP	67.0	66.7	66.9
Single-head	68.2	68.8	68.5
Four-head	**69.1**	**71.9**	**70.5**

B.3 Alternative Feature Fusion

We compare complete feature-fusion interfaces to examine how Euclidean and hyperbolic predictions support action generation. The default adapter concatenates Euclidean tokens and tangent-space hyperbolic tokens along the token dimension and reads them through attention. Alternative interfaces resample each bank to 16 slots per prediction horizon, giving $U_E, U_H \in \mathbb{R}^{48\times 256}$, and apply channel concatenation, averaging $(U_E+U_H)/2$, gated addition $\gamma\odot U_E+(1-\gamma)\odot U_H$, or elementwise multiplication $U_E\odot U_H$, where $\gamma=\mathrm{sigmoid}(G([U_E\|U_H]))$ and G is a learned mapping. These alternatives share normalization, output width, and training budget, with bottleneck layers used to reduce parameter-count differences. As shown in Table 7, the default interface achieves the highest average success rate of 70.5%, exceeding the strongest alternative, gated addition, by 1.05 percentage points. These results support the default attention interface for incorporating the two prediction banks into action generation.

B.4 Residual Adapters and Future-Content Interventions

Residual Adapter Comparison. We evaluate how different residual adapters transfer predicted future information to action generation, using the same predictor setup, feedback location, residual bound, gain schedule, and zero-initialized output projection. The pooled MLP aggregates context tokens into a shared vector, while the token-mixing MLP maps them to action-query positions; both combine the resulting context with projected queries. Their bottleneck widths are selected to reduce parameter-count differences, and both attention variants use a width of 256. As shown in Table 8, four-head attention achieves the highest average success rate of 70.5%, outperforming both MLP variants and exceeding single-head attention by 2.0 percentage points. These results support the multi-head residual adapter as an effective interface for using geometric future features to guide action generation.

Effect of Predictive Feedback Content. We examine the role of predictive content by fixing the trained checkpoint and replacing its feedback features with either current-observation features or mismatched future predictions. Current features are repeated across the future-token layout, while mismatched futures are sampled from other trajectories matched by task, episode-time quartile, and generation step. Both substitutions preserve the geometric mappings and token layouts, match each bank's input RMS, and keep the predictor active. With paired initial conditions and sampling seeds, current-feature and mismatched-future substitutions reduce average success by 8.70 and 11.75 percentage points, respectively, as shown in Table 9. These decreases support the contribution of predicted future content to action generation, while the larger drop with mismatched futures highlights the importance of its correspondence with the current condition.

Table 9: Feedback-content substitutions: success rates (%).

Feedback	Google	WidowX	Avg.
Current features	62.1	61.5	61.8
Mismatched future	59.2	58.3	58.8
Predicted future	**69.1**	**71.9**	**70.5**

Table 10: OpenVLA-OFT success rates (%); gains in percentage points.

Benchmark	OpenVLA-OFT	+GeoWRA	Gain
LIBERO Avg.	97.1	**97.7**	+0.6
LIBERO-Plus	71.4	**73.9**	+2.5

Table 11: Prediction NRMSE (↓) at horizons $j = 1, 2, 3$.

Predictor	$j=1$	$j=2$	$j=3$
Demonstration trajectories			
Persistence	0.319	0.470	0.588
Without q_t	0.264	0.370	0.448
Without a_τ	0.251	0.346	0.423
Full predictor	**0.221**	**0.292**	**0.353**
Policy rollouts			
Persistence	0.373	0.492	0.619
Without q_t	0.301	0.396	0.495
Without a_τ	0.283	0.379	0.463
Full predictor	**0.249**	**0.338**	**0.393**

B.5 Integration with an Additional VLA Backbone

To evaluate the applicability of GeoWRA across VLA architectures, we integrate it with OpenVLA-OFT (Kim et al., 2025). The modular world-residual formulation can, in principle, accommodate any VLA that exposes intermediate action representations and supports residual feature injection. For OpenVLA-OFT, backbone action representations and an initial action proposal condition geometric prediction, while projections matched to the backbone width inject bounded feedback before action readout. This integration preserves the native continuous action head, action normalization, and chunk structure, and is trained and evaluated separately from the Qwen configuration. As shown in Table 10, GeoWRA improves average success from 97.1% to 97.7% on LIBERO and from 71.4% to 73.9% on LIBERO-Plus, yielding gains of 0.6 and 2.5 percentage points, respectively. These results demonstrate compatibility with an additional continuous-action backbone and support the broader applicability of the world-residual formulation.

B.6 Action-Conditioned Future Prediction

To assess action-to-world coupling, we examine the contributions of backbone context q_t and the current action sample a_τ to future-feature prediction. We compare current-feature persistence, two retrained variants that separately remove q_t or a_τ from the world predictor, and the full predictor. Both ablations retain flow-time conditioning and the action expert's inputs. The comparison uses common demonstration targets and matched rollout targets across predictors. We evaluate at $\tau = 1$, where $a_\tau = a$ is the recorded action chunk, using the normalized root mean squared error (NRMSE):

$$e_j = \frac{\mathrm{RMS}(\widehat{E}_j - E_j^\star)}{\max\{\mathrm{RMS}(E_j^\star), 10^{-6}\}}.$$

Here $\widehat{E}_j = \{\hat{e}_{j,r}\}_{r\in\mathcal{R}_E}$ and $E_j^\star = \{e_{j,r}^\star\}_{r\in\mathcal{R}_E}$ contain the predicted and target Euclidean tokens at horizon j, respectively, and RMS is computed over all entries. Errors are averaged across task families separately for each horizon. All predictors use the same target construction; rollout comparisons additionally match starting states and executed actions. As shown in Table 11, the full predictor achieves the lowest NRMSE at all three horizons on both demonstrations and rollouts, supporting the contributions of backbone context and action-sample conditioning to future prediction. Together with the feedback-content interventions in Appendix B.4, they examine the action-to-world and world-to-action paths of the coupled architecture.

C Implementation Details

C.1 Architecture and Residual Mechanism

Prediction layout and action conditioning. GeoWRA uses the same geometric representation and residual-feedback formulation across benchmarks. The default configuration uses $J = 3$ pre-

diction horizons ordered in time. Each prediction horizon corresponds to one future target block in the frozen teacher's temporal output.

At each horizon, $\mathcal{R}_E$ is a 4×4 grid, while $\mathcal{R}_H$ contains the regions from nested 1×1, 2×2, and 4×4 grids. Thus, each horizon contains 16 Euclidean tokens and 21 hyperbolic tokens, giving $N_E = 48$ and $N_H = 63$ in total. Each non-root region is linked to its containing region at the next coarser resolution, yielding $4 + 16 = 20$ spatial edges per horizon and $|\mathcal{E}_S| = 60$ in total. Temporal edges link the same spatial region at adjacent horizons, giving $|\mathcal{E}_T| = (J-1)|\mathcal{R}_H| = 42$. Latent-action tokens q_t provide the prediction condition, while action tokens C_t query the future banks and receive residual feedback. In the default configuration, we divide q_t into J equal, nonempty groups $\mathcal{G}_j$ and compute

$$\bar{q}_j = \mathrm{LN}_0\left(\frac{1}{|\mathcal{G}_j|}\sum_{i\in\mathcal{G}_j} q_{t,i}\right), \qquad \bar{q}_{\le j} = \frac{1}{j}\sum_{k=1}^{j}\bar{q}_k. \tag{33}$$

Here LN_0 is channel-wise layer normalization without learned affine parameters. The Euclidean branch uses $\bar{q}_j$, while the hyperbolic branch uses $[\bar{q}_j \| \bar{q}_{\le j}]$. An MLP jointly encodes the flattened action sample a_τ and $[\tau, \sin(\pi\tau), \cos(\pi\tau)]$ to condition both prediction branches. The Qwen hyperbolic predictor contains two width-256 Transformer layers with attention across all prediction horizons and no additional causal mask.

Bounded residual interface. For Qwen, LN_0 normalizes 32 action tokens of width 2048. These tokens and the width-64 future features are projected to width 256. Four-head attention reads 111 context tokens with head width 64 and zero dropout. The external maps $\mathcal{P}_E$, $\mathcal{P}_H$, $\mathcal{P}_Q$, and $\mathcal{P}_O$ are distinct from the internal attention projections. The output map restores width 2048 with zero-initialized weights and bias. Equation 4 uses $\rho = 0.2$, $b(\tau) = 0.25 + 0.75\tau$, and $\varepsilon_C = 10^{-3}$, with stop-gradient on the RMS scale. The resulting $\widetilde{C}_{t,\tau}$ conditions the action expert. Action gradients pass through the adapter and predictor, while the teacher remains frozen. Appendix A.1 gives the feedback bounds.

Module cost. In the Qwen configuration, the geometric predictor and residual adapter add 5.34M trainable parameters and require 1.28 GFLOPs per action chunk, summed over the four action-generation steps. This FLOP count excludes the frozen visual encoder, which processes the current observations once per action-generation call; its features are cached and reused across all four steps. However, despite this modest increase in trainable parameters, GeoWRA improves the overall SimplerEnv success rate from 56.4% to 70.5%, achieving a gain of 14.1 percentage points.

C.2 Teacher Targets and Numerical Geometry

Teacher feature targets. The frozen V-JEPA encoder processes repeated current images to obtain context features. For supervision, it processes an eight-frame recorded clip $\mathcal{V}_t$ to obtain target features for the $J = 3$ prediction horizons. We denote the target features at horizon j by $z_j^\star = \mathrm{sg}[E_{\mathrm{T}}(\mathcal{V}_t)_j]$. Regional pooling and frozen projections give

$$e_{j,r}^\star = A_E\,\mathrm{LN}_0(\mathrm{Pool}_r\, z_j^\star), \qquad u_{j,r}^\star = A_H\,\mathrm{LN}_0(\mathrm{Pool}_r\, z_j^\star). \tag{34}$$

Here Pool_r averages features within region r, and each pooled feature is treated as a column vector of width d_T. The frozen projections $A_E, A_H \in \mathbb{R}^{64\times d_T}$ have orthonormal rows. They are distinct from the learned readout maps $\mathcal{P}_E$ and $\mathcal{P}_H$ in Equation 3. Euclidean targets use the 4×4 grid, and hyperbolic targets use all three nested grids. Applying the same transformations to the current features gives the reference banks in Equation 8.

Hyperbolic point and relation targets. For the token index $i = (j, r)$, let $u_i^\star = u_{j,r}^\star$. The target point and the reference distance for each supervised edge are

$$\begin{aligned} h_i^\star &= \exp_0^c(\mathcal{T}(u_i^\star))\,, \\ d_{ik}^\star &= d_c(h_i^\star, h_k^\star), \qquad (i,k)\in\mathcal{E}_S\cup\mathcal{E}_T. \end{aligned} \tag{35}$$

Equation 9 selects nested region pairs within each horizon and pairs of the same spatial region at adjacent horizons. Their reference distances are computed from recorded future clips using the

frozen teacher and target transformations. The relation objective in Equation 13 matches $d_c(\hat{h}_i, \hat{h}_k)$ to $d^\star_{ik}$. These teacher-derived distances require no additional relation annotations and are fixed for each recorded clip.

Numerical settings and supervision. We use curvature $-c = -0.5$, $R_{\max} = 1.25$, and $\sigma_H = 2$. The manifold maps, distances, and logarithmic readout are computed in float32 with boundary clipping and numerical stabilization. Appendix A.2 analyzes the corresponding exact maps. Future observations enter only the target branch. The joint objective in Equation 14 uses $\lambda_E = 0.05$, $\lambda_{H,p} = 0.02$, and $\lambda_{H,r} = 0.01$. Validity-weighted averages follow Section 3.4, with spatial and temporal relation losses normalized separately. The Qwen and $\pi_{0.5}$ configurations share these supervision and gradient pathways.

C.3 Real-World Experimental Setup and Tasks

Platform and sensing. We conduct physical experiments on Cobot Mobile ALOHA, which has two 6-DoF robotic arms with parallel grippers. A wrist-mounted RGB camera on each arm captures local interactions, while a third-person RGB camera provides a view of the tabletop workspace. All policies receive the same three camera views and task instruction, combining local manipulation cues with the positions of objects and targets in the full scene.

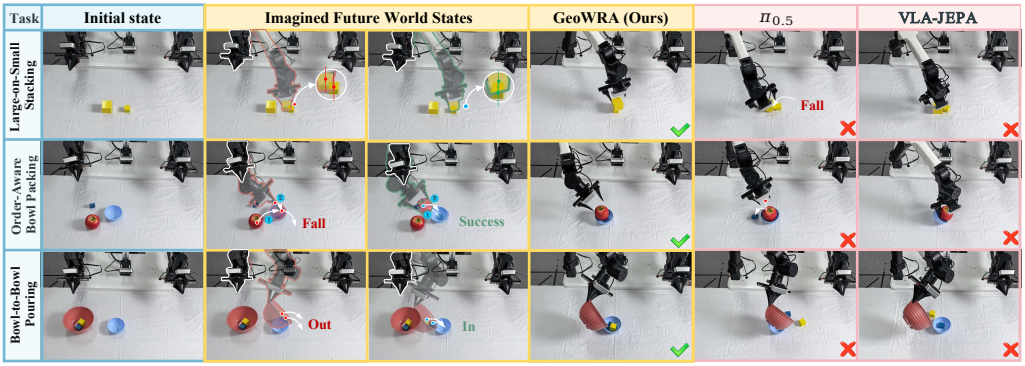


Figure 6: Real-world comparisons for three real world tasks, shown from top to bottom. Each row presents the initial state, Imagined Future World States, a successful GeoWRA execution, and representative $\pi_{0.5}$ and VLA-JEPA failures.

Task definitions. The six tasks cover moving objects, unstable contacts, obstacles, partial occlusion, ordering constraints, and pouring. They provide different settings in which anticipating the effects of an action can support manipulation.

1. **Moving-object interception** requires grasping a block moving on a conveyor belt and placing it in the target container. The task tests the timing of the grasp relative to the block's motion.
2. **Unstable-stack transfer** requires moving an unstable stack of blocks onto the target plate while preserving its support structure. The stack must remain intact throughout pickup, transport, and placement.
3. **Obstacle-aware placement** requires transporting a block to the target plate without contacting or displacing a trash bin that obstructs the direct path.
4. **Large-on-small stacking** requires placing a large block stably on a smaller block. The large block partly occludes the supporting surface during placement, making alignment more difficult.
5. **Order-aware bowl packing** requires placing a block and a piece of fruit in a compact bowl. In the evaluated setup, the block must be placed first, since placing the fruit first leaves insufficient space for the block.

Table 12: Real-world successes with 40 trials per task. Bold numerators indicate better results.

Task	GeoWRA	VLA-JEPA
Moving-object interception	**27**/40	23/40
Unstable-stack transfer	**25**/40	18/40
Obstacle-aware placement	**26**/40	21/40
Large-on-small stacking	**21**/40	17/40
Order-aware bowl packing	**22**/40	19/40
Bowl-to-bowl block pouring	**21**/40	16/40
Total	**142**/240	114/240

6. **Bowl-to-bowl block pouring** requires grasping and tilting a source bowl to transfer all contained blocks into the target bowl without spilling.

Quantitative analysis. GeoWRA achieves the highest observed success rate on all six real-world tasks (Table 4), completing 142 of 240 trials (59.2%) compared with 114 of 240 trials (47.5%) for VLA-JEPA. This corresponds to 28 additional successful trials and an overall improvement of 11.7 percentage points. The largest task-level gain is 17.5 percentage points on unstable-stack transfer, highlighting improved performance when handling unstable objects. Large-on-small stacking and bowl-to-bowl block pouring yield the lowest GeoWRA success rates, both 52.5%, while still exceeding the baseline results. These consistent improvements across tasks support the effectiveness of GeoWRA in physical manipulation. Detailed success counts are provided in Table 12.

Qualitative analysis. Figures 5 and 6 complement the quantitative results with successful GeoWRA executions and representative baseline failures, alongside schematic illustrations of possible task outcomes. The successful executions demonstrate that GeoWRA satisfies the corresponding manipulation requirements, whereas the selected baseline failures involve mistimed grasps, stack collapse, obstacle contact, incorrect operation order, or spillage. These examples illustrate successful handling of requirements involving grasp timing, object stability, obstacle avoidance, and coordinated manipulation, providing qualitative support for the performance gains observed across the six tasks.

C.4 Training and Evaluation Settings

Training and implementation details. We initialize GeoWRA from a pretrained VLA policy and train it using eight NVIDIA A800 GPUs. We use AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$ and weight decay 0.01. The peak learning rates are 2×10^{-6} for the last eight language layers, 10^{-5} for the action expert, 4×10^{-6} for the geometric predictor, and 1.5×10^{-6} for the residual adapter. The visual backbone and teacher encoder remain frozen. For LIBERO and LIBERO-Plus, we use $\pi_{0.5}$ as the base VLA policy. For SimplerEnv, we use Qwen3-VL-2B-Instruct as the vision-language backbone and a GR00T-style flow-matching action expert. The StarVLA baseline in Table 3 is trained by us using the same backbone, action expert, training data, batch size, and number of training updates as GeoWRA. Shared modules use the same initialization, trainable layers, and optimization settings. The baseline is trained with action supervision alone, without the geometric predictor, residual adapter, or geometric objectives. Both models follow the same checkpoint-selection and evaluation protocols.

Evaluation Settings. For simulation models evaluated in this work, we report the mean success rate over three evaluation seeds using the same trained checkpoint for each model. For each seed, we evaluate 2,000 episodes on LIBERO, 10,030 on LIBERO-Plus, and 864 and 96 for the Google and WidowX robots in SimplerEnv, respectively. Specifically, benchmark averages are computed as unweighted means across the four task suites in LIBERO, the seven perturbation categories in LIBERO-Plus, and the evaluated task families within each robot in SimplerEnv. Google and WidowX receive equal weight in the overall SimplerEnv score. Real-world evaluation uses 40 trials per task across six tasks, totaling 240 trials per policy, and the average success rate is the unweighted mean across the six tasks.