CoGeo-WAM: Controllable Geometry Adaptation for World Action Models

Anonymous authors
Paper under double-blind review

Abstract

World action models (WAMs) have demonstrated impressive robotic manipulation capabilities by jointly modeling future visual dynamics and actions. However, their visual representations are largely learned from RGB videos without explicit geometric perception, creating a mismatch with the inherently 3D nature of physical interaction. Existing approaches can strengthen spatial awareness through explicit geometry modeling, but often introduce additional geometric inputs. Moreover, geometry-oriented adaptation can alter useful pretrained representations, while its utility varies across manipulation contexts, making uniform geometry adaptation potentially suboptimal. To this end, we propose **CoGeo-WAM**, a framework for controllable geometry adaptation in world action models that combines *implicit geometry alignment* with *adaptive geometry control*. We align intermediate WAM video representations with 3D-aware features from a frozen pretrained 3D vision model and capture the resulting geometry-aligned adaptation in a LoRA residual branch, requiring no explicit geometric inputs or additional geometry model at inference time. A context-guiden controller then dynamically modulates this branch, allowing the model to selectively exploit geometric adaptation and preserve its pretrained video-action prior in different manipulation contents. Extensive experiments across multiple simulation benchmarks, real-world manipulation tasks, and different WAM backbones demonstrate that CoGeo-WAM consistently improves manipulation performance and outperforms both the original world action models and static geometry adaptation.

1 Introduction

Robot manipulation requires understanding how actions will change the future physical world (Hou et al., 2026). World action models (WAMs) address this problem by jointly modeling future visual dynamics and robot actions, explicitly capturing object motion, interaction dynamics, and action-conditioned outcomes (Wang et al., 2026; Cheang et al., 2024; Zhu et al., 2025; Cen et al., 2025; Li et al., 2026b). Recent WAMs have demonstrated impressive manipulation capabilities, highlighting future prediction as a promising paradigm for robot control (Ye et al., 2026; Kim et al., 2026; Liao et al., 2026; Bi et al., 2026; Yuan et al., 2026; Liu et al., 2026).

Although WAMs model future dynamics, their visual representations are largely learned from RGB video observations without explicit geometric perception (Guo et al., 2026). Robot manipulation, in contrast, takes place in a 3D world and requires reasoning about spatial structure and physical contact (Ze et al., 2024; Goyal et al., 2024). Recent 3D-aware world models (WMs) and WAMs attempt to bridge this gap by incorporating explicit geometric inputs or introducing 3D reconstruction objectives during training (Lu et al., 2025; Huang et al., 2026; Guo et al., 2026). These approaches can strengthen spatial awareness but often require additional geometric supervision or computation, which may limit scalability and increase inference complexity. Moreover, geometry-oriented adaptation alters pretrained WAM representations, yet its utility varies across manipulation contexts. Figure 1(b) shows that the effect of geometry adaptation varies across manipulation contexts. It improves *Turn Switch*, where geometric reasoning is important, but hurts *Pick Diverse Bottles*, where the pretrained video-action prior is already effective. These drawbacks pose two key challenges for 3D-aware WAMs: *(1) equipping WAMs with geometric awareness without explicit geometry modeling at inference time; and (2) adaptively controlling the contribution of geometric adaptation according to the current context.*

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.

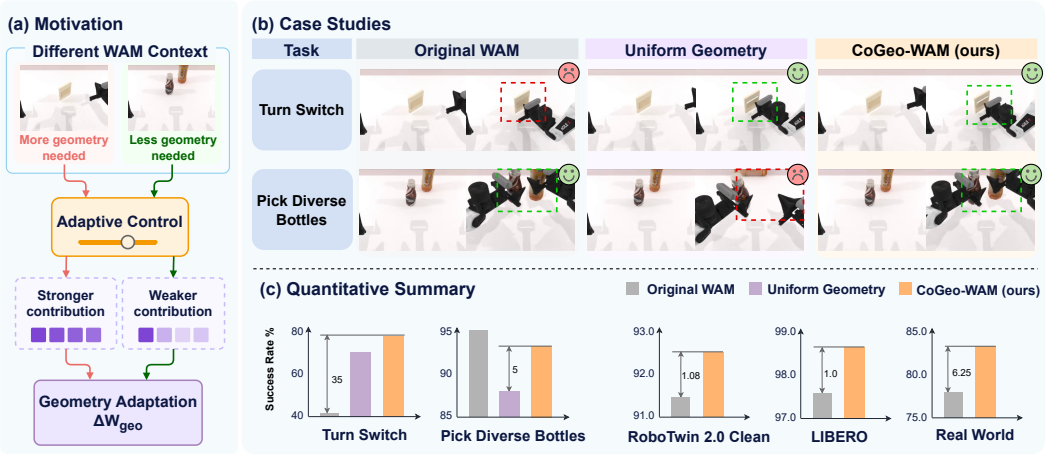

Figure 1: **Adaptive geometry control in CoGeo-WAM.** (a) Different WAM contexts require different strengths of geometry adaptation. (b) Case studies illustrate the contrasting effects of Uniform Geometry and the benefit of adaptive control. (c) CoGeo-WAM further improves the original WAMs on representative tasks, RoboTwin 2.0 Clean, LIBERO and real-world manipulation tasks.

This motivates regulating the contribution of geometry adaptation according to the current context rather than applying it with a fixed strength, as illustrated in Figure 1(a). Based on this insight, we propose CoGeo-WAM, a framework for controllable geometry adaptation in WAMs, which efficiently and adaptively inject geometry perception into WAMs considering different manipulation contexts. CoGeo-WAM consists of two components: *(1) implicit geometry alignment*, which transfers 3D-aware knowledge into intermediate WAM video representations and isolates the resulting adaptation in a low-rank adaptation (LoRA) (Hu et al., 2022) residual branch, and *(2) adaptive geometry control*, which regulates the contribution of this residual according to the current WAM context. Together, these components allow the model to incorporate geometric knowledge while preserving and selectively relying on its pretrained video-action prior. CoGeo-WAM requires neither explicit geometric inputs nor an external 3D vision model at inference.

Specifically, for implicit geometry alignment, we use a frozen pretrained 3D vision model (Wang et al., 2025) to extract 3D-aware reference representations from RGB videos, which are spatiotemporally matched to the WAM video-token grid. A lightweight projector maps intermediate WAM video features into the reference representation space, and LoRA is optimized to align these representations while the pretrained WAM parameters remain frozen. For adaptive geometry control, we introduce a learned context-guiden controller that predicts modulation weights for the geometry-aligned LoRA residual from the current WAM context. Rather than applying a fixed geometry adaptation uniformly, the controller regulates its contribution across tasks, flow times, and intermediate representations. We further investigate controllers with different control granularities and conditioning signals to characterize how controllable geometry utilization affects manipulation performance.

We conduct extensive evaluations on LIBERO and RoboTwin 2.0, together with real-world manipulation tasks, across two representative WAM backbones, FastWAM and LingBot-VA. CoGeo-WAM *consistently* improves manipulation performance across these diverse settings, demonstrating its effectiveness beyond any specific architecture or benchmark, see Figure 1(c). We further show that adaptive geometry control outperforms static geometry adaptation and conduct systematic ablations on different controller designs to analyze the effects of control granularity and conditioning information. In summary, the contributions of this paper are as follows:

- **Implicit geometry alignment for WAMs.** We introduce implicit geometry alignment into WAMs, transferring pretrained 3D-aware knowledge into intermediate video representations without requiring explicit geometry at inference time.
- **Adaptive geometry control.** We isolate the geometry-aligned adaptation in a LoRA residual branch and design a context-guiden controller to modulate its contribution, enabling context-guiden geometry adaptation while absorbing the advantage of the pretrained video-action prior.

- Extensive experiments across different WAM backbones and benchmarks demonstrate consistent improvements and establish new state-of-the-art performance.

2 Related Work

World Action Models for Robot Manipulation. WAMs combine visual dynamics modeling with action learning for robot manipulation. WorldVLA (Cen et al., 2025) integrates image generation and action prediction within an autoregressive framework, while Motus (Bi et al., 2026) unifies video and action modeling through latent action representations. Building on pretrained video models, DreamZero (Ye et al., 2026) jointly models video and actions to support generalization across tasks and embodiments. Cosmos Policy (Kim et al., 2026) represents actions, future states, and values as latent frames within a video model, supporting both control and planning. LingBot-VA (Li et al., 2026b) adopts causal video-action modeling to infer actions from predicted visual transitions. FastWAM (Yuan et al., 2026) further examines the role of test-time future generation, showing that video co-training can support strong action prediction even without explicit future-video generation at inference. However, these approaches primarily learn visual representations through video modeling without explicitly enforcing geometric awareness.

Geometry-Aware Representations for Robot Learning. Geometric representations provide spatial information for robot perception and control. 3D-VLA (Zhen et al., 2024) connects 3D perception and action with generative modeling of goal images and point clouds. SpatialVLA (Qu et al., 2025) introduces Ego3D position encoding to inject spatial information into visual representations. X-WAM (Guo et al., 2026) jointly predicts multi-view RGB-D videos and actions, using a dedicated depth branch for future geometric reconstruction.

An alternative is to transfer geometric knowledge through representation alignment. Spatial Forcing (Li et al., 2026a) aligns intermediate VLA visual embeddings with features from a pretrained 3D model, enabling spatial representation learning without explicit 3D inputs. Geometry Forcing (Wu et al., 2026) aligns intermediate video diffusion representations with geometric features to improve the geometric consistency of generated videos. In contrast, CoGeo-WAM combines implicit geometry alignment with context-guiden control of the learned adaptation, allowing its contribution to vary rather than applying it uniformly.

3 Preliminaries–World Action Models

We briefly describe the WAM components relevant to our method. WAMs commonly encode RGB observations using a pretrained video VAE, producing temporally and spatially compressed video latent tokens. A video Diffusion Transformer (DiT) processes these tokens together with task and robot conditions, while an action module models the corresponding action tokens. At DiT block l, we denote the intermediate video representation by $H_l^v = \{h_{l,i}\}_{i\in\Omega}$, where Ω indexes the spatiotemporal video-token grid.

WAMs may explicitly generate future observations before action prediction or jointly model video and action tokens in a shared generative process (Li et al., 2026b; Yuan et al., 2026). In either case, video modeling provides a representation-level learning signal that captures motion and interaction structure. However, video prediction alone does not explicitly enforce 3D geometric consistency, and the resulting representations may inadequately capture spatial relations needed for precise manipulation. CoGeo-WAM addresses this limitation by transferring 3D-aware knowledge into intermediate video representations during training and adaptively regulating its contribution, without introducing an explicit geometric interface at inference.

4 Method

4.1 Overview

As illustrated in Figure 2, our method equips a pretrained world-action model (WAM) with controllable geometric awareness through two stages: (I) *implicit geometry alignment* and (II) *adaptive geometry control*. Let W denote the parameters of a pretrained WAM, and let W_{geo} denote the pa-

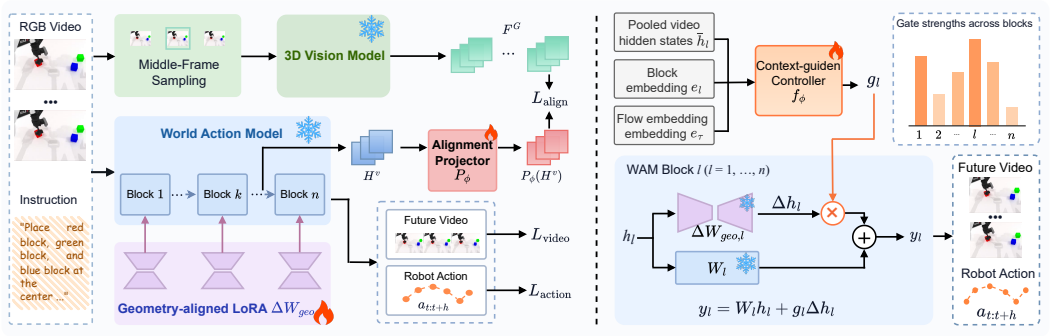

Figure 2: **Overview of CoGeo-WAM.** Stage I performs *implicit geometry alignment* to learn a geometry-aligned LoRA residual from 3D-aware supervision. Stage II freezes this residual and uses *adaptive geometry control* to modulate its contribution according to the current WAM context.

rameters after learning from 3D-aware supervision. We view their difference as a geometry-related parameter update $\Delta W_{\text{geo}} = W_{\text{geo}} - W$, Rather than treating geometry adaptation as an inseparable modification of the entire WAM, we explicitly isolate the geometry-related update ΔW_{geo}. Stage I learns this update by transferring 3D-aware knowledge from a frozen pretrained 3D vision model into intermediate video representations of the WAM. Stage II controls how strongly the learned update contributes under the current manipulation context.

We instantiate ΔW_{geo} using a LoRA residual branch. Specifically, for a frozen linear transformation $y = Wh$, LoRA learns low-rank factors A and B such that

$$y = Wh + \frac{\alpha}{r} BAh, \tag{1}$$

where r is the LoRA rank and α is a fixed scaling factor. Accordingly, the geometry-related update is parameterized as

$$\Delta W_{\text{geo}} = \frac{\alpha}{r} BA. \tag{2}$$

This low-rank parameterization provides an efficient interface for learning and controlling geometry adaptation without explicit 3D inputs or external 3D models at inference.

4.2 Implicit Geometry Alignment

We first learn ΔW_{geo} by aligning intermediate WAM video representations with 3D-aware reference features. The alignment is implicit: the WAM receives only its original video, action, and proprioceptive inputs, while the frozen 3D vision model provides supervision only during training. Consequently, geometric knowledge is transferred into the WAM representation without introducing an explicit geometric input interface at inference time.

3D-aware reference representations. Given an RGB video observation $V = \{I_t\}_{t=1}^{T}$, we use a frozen pretrained 3D vision model E_G, instantiated as VGGT (Wang et al., 2025), to construct 3D-aware reference representations. VGGT is a feed-forward model that infers camera parameters, point maps, depth maps, and 3D point tracks from one or more views. We use its dense intermediate features as geometry-aware references.

The temporal resolution of RGB observations differs from that of WAM video latents. For each temporally compressed latent frame, we associate its corresponding group of RGB frames according to the video tokenizer and select the middle frame as its geometry reference. The selected reference-frame sequence $\bar{V}$ is processed by E_G, yielding $F^G = \mathcal{M}\big(E_G(\bar{V})\big)$, where $\mathcal{M}$ denotes the spatial transformation to the WAM video-token grid. We denote the set of locations on this grid by Ω. For each location $i \in \Omega$, the corresponding 3D-aware reference feature is denoted by f_i^G.

Intermediate representation alignment. Let $H^v = \{h_i\}_{i\in\Omega}$ denote the intermediate WAM video representation, where h_i is the hidden feature at video-token location i. Intermediate features retain dense spatiotemporal information while integrating semantic context and predicted dynamics, making them an effective interface for geometry transfer.

Since h_i and f_i^G lie in different feature spaces, we introduce an alignment projector P_ϕ and obtain $\hat{f}_i^G = P_\phi(h_i)$. The projector consists of LayerNorm followed by a two-layer MLP. Its bottlenecked design makes it a lightweight readout from WAM features into the 3D-aware reference space, concentrating the geometry-alignment pressure on the intermediate WAM representation.

Before adapting the WAM, we calibrate the alignment projector using frozen WAM and frozen 3D vision model features. This calibration stage establishes a stable correspondence between the two feature spaces without modifying the pretrained WAM. We then use the calibrated projector to initialize joint geometry-alignment training and keep it trainable thereafter.

Geometry-aligned LoRA learning. We align each projected WAM feature with its corresponding 3D-aware reference using cosine similarity:

$$\mathcal{L}_{\text{align}} = \frac{1}{|\Omega|} \sum_{i \in \Omega} \left[1 - \cos\left(\hat{f}_i^G, f_i^G \right) \right]. \tag{3}$$

Minimizing $\mathcal{L}_{\text{align}}$ transfers the 3D-aware relational structure encoded by the frozen 3D vision model into intermediate WAM video features. We jointly optimize the geometry-alignment objective with the original WAM training losses:

$$\mathcal{L}_{\text{geo}} = \mathcal{L}_{\text{video}} + \mathcal{L}_{\text{action}} + \lambda_{\text{align}} \mathcal{L}_{\text{align}}, \tag{4}$$

where $\mathcal{L}_{\text{video}}$ and $\mathcal{L}_{\text{action}}$ are the video and action flow-matching losses, respectively, and λ_{align} balances the geometry-alignment objective. Only the LoRA factors and alignment projector are updated, while the pretrained WAM weights W remain frozen. Thus, implicit geometry alignment learns a geometry-aligned LoRA residual that is subsequently modulated by adaptive geometry control. Neither the 3D vision model nor the alignment projector is used at inference.

4.3 Adaptive Geometry Control

The implicit geometry alignment stage learns the geometry-aligned LoRA update ΔW_{geo} in Eq. 2. While this update introduces 3D-aware knowledge into the WAM, its contribution need not be equally beneficial across manipulation contexts, flow times, or intermediate representations. We therefore introduce *adaptive geometry control* to modulate the contribution of ΔW_{geo} according to the current WAM context. During this stage, we freeze the geometry-aligned LoRA factors and set $\lambda_{\text{align}} = 0$ in Eq. 4, since further feature alignment does not help determine when the learned residual should contribute. Consequently, neither the frozen 3D vision model nor the alignment projector is required during adaptive-control training.

Given the input feature h_l for DiT block l, we denote by Δy_l the residual output produced by its geometry-aligned LoRA branch. This residual is added to the corresponding frozen base transformation. To simplify notation, we suppress the fixed LoRA scaling factor α/r in the gate equations below. The controller predicts a positive gate through exponential mapping, such that a controller output near zero initially recovers the original geometry-aligned LoRA residual.

We use the DiT block as the basic unit for geometry control. We regard each DiT block as a natural unit of representation transformation, whose hidden states collectively characterize the representation state at a particular processing depth. We therefore use mean pooling to summarize the collective state of each block as $\bar{h}_l = \text{MeanPool}(h_l)$, rather than making the geometry strength depend on individual token states. The gate for block l is then computed as

$$g_l = \exp\left(f_\phi([\bar{h}_l, e_l, e_\tau]) \right),$$

where e_l is a learned block embedding and e_τ is the flow-time embedding. The gate modulates all geometry-aligned LoRA residuals within block l:

$$\Delta y_{l,n} = g_l \cdot B_l A_l h_{l,n}. \tag{5}$$

This formulation allows the model to regulate geometry adaptation at different representation depths while conditioning the control on the current WAM state and flow time.

Table 1: Results on RoboTwin 2.0 using LingBot-VA as backbone. Success rates (%) under clean and randomized settings.

Method	Clean	Rand.	Avg.
$\pi_{0.5}$	42.98	43.84	43.41
X-VLA	72.80	72.84	72.82
Motus	88.66	87.02	87.84
X-WAM	89.80	90.70	90.25
FastWAM-Joint	90.84	**91.32**	91.08
LingBot-VA	91.36	90.76	91.06
CoGeo-WAM (Ours)	**92.44**	90.58	**91.51**

Table 2: Results on LIBERO using FastWAM-Joint as backbone. Success rates (%) on four task suites.

Method	Spatial	Object	Goal	Long	Avg.
$\pi_{0.5}$	98.8	98.2	**98.0**	92.4	96.9
WorldVLA	87.6	96.2	83.4	60.0	81.8
X-VLA	98.2	98.6	97.8	97.6	98.1
Motus	96.8	**99.8**	96.6	97.6	97.7
LingBot-VA	98.5	99.6	97.2	**98.5**	98.5
FastWAM-Joint	98.0	99.4	97.6	95.6	97.7
CoGeo-WAM (Ours)	**99.2**	**99.8**	**98.0**	97.6	**98.7**

5 Experiments

5.1 Experimental Setup

Implementation details. We evaluate CoGeo-WAM on two representative WAMs backbones: LingBot-VA (Li et al., 2026b) and FastWAM-Joint (Yuan et al., 2026), applying the same controllable geometry-adaptation framework without changing the original backbone architectures or input modalities. For both backbones, we extract training-time geometry references using frozen VGGT (Wang et al., 2025) and align the video representation from the 20th DiT block with the spatiotemporally matched VGGT features. The alignment projector is a two-layer MLP with a 512-dimensional bottleneck and a 2048-dimensional output space. During joint geometry alignment, we set $\lambda_{\text{align}} = 0.005$. During adaptive-control training, we freeze the geometry-aligned LoRA factors, set $\lambda_{\text{align}} = 0$, and use the exponential function to map controller outputs to positive geometry gates.

Benchmarks. We evaluate CoGeo-WAM on two simulation benchmarks: (i) RoboTwin 2.0 (Chen et al., 2026), a scalable dual-arm manipulation benchmark with unified evaluation protocols across 50 tasks and diverse scene variations induced by structured domain randomization; and (ii) LIBERO (Liu et al., 2023), a lifelong robot-manipulation benchmark that provides high-quality human-teleoperated demonstrations and four task suites, *Spatial*, *Object*, *Goal*, and *Long*. These suites cover manipulation tasks with different spatial, object-centric, goal-conditioned, and long-horizon requirement. We additionally evaluate CoGeo-WAM on a Franka robot across four real-world tabletop tasks: *Cup Stacking*, *Block Stacking*, *Bread Placement*, and *Water Pouring*, illustrated in Figure 3. These tasks involve nesting one cup inside another, stacking three blocks, placing packaged bread onto a plate, and pouring water from one cup into another, respectively. They cover spatial alignment, sequential placement, and object orientation control. For real-world evaluation, we instantiate CoGeo-WAM on FastWAM-Joint and compare it with $\pi_{0.5}$ and the original FastWAM-Joint backbone. Each method is evaluated over 20 trials per task, and we report per-task success rates and their average.

Baselines. We compare CoGeo-WAM with representative VLA and world-action-model baselines. For VLA models, we include $\pi_{0.5}$ (Black et al., 2025), a generalist VLA trained through heterogeneous co-training, and X-VLA (Zheng et al., 2026), a flow-matching VLA that uses embodiment-specific soft prompts for cross-embodiment learning. For world-action models, we compare with WorldVLA (Cen et al., 2025), which jointly models autoregressive actions and future images; Motus (Bi et al., 2026), a unified latent action world model with understanding, video-generation, and action experts; and X-WAM (Guo et al., 2026), which predicts multi-view RGB-D observations for 4D world modeling. We further include FastWAM-Joint (Yuan et al., 2026), which jointly denoises future video and action tokens, and LingBot-VA (Li et al., 2026b), which jointly models visual dynamics and actions in a shared latent space.

5.2 Main Results

Simulation results. Tables 1 and 2 summarize the simulation results on RoboTwin 2.0 and LIBERO, respectively. We instantiate CoGeo-WAM on LingBot-VA for RoboTwin 2.0 and on FastWAM-Joint

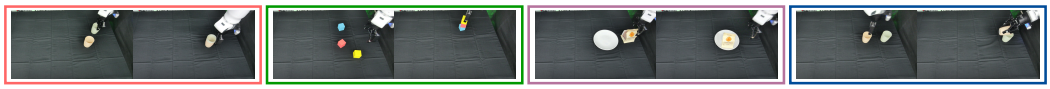

Figure 3: Real-world manipulation tasks on a Franka robot: (a) Cup Stacking, (b) Block Stacking, (c) Bread Placement, and (d) Water Pouring. Each task is illustrated by two frames.

Table 3: Real-world success rates (%) on a Franka robot, evaluated over 20 trials per task for each method. Avg. is the mean across the four tasks.

Method	Cup Stacking	Block Stacking	Bread Placement	Water Pouring	Avg.
$\pi_{0.5}$	90	65	85	55	73.75
FastWAM-Joint	90	60	**95**	65	77.50
CoGeo-WAM (Ours)	**95**	**75**	**95**	**70**	**83.75**

for LIBERO, thereby evaluating whether controllable geometry adaptation transfers across different WAM implementations and manipulation settings.

On RoboTwin 2.0, CoGeo-WAM achieves the highest average success rate of 91.51%, outperforming all compared baselines. Relative to the LingBot-VA backbone, it improves the clean-setting success rate by 1.08 points and the average success rate by 0.45 points. These results demonstrate that CoGeo-WAM improves a strong WAM backbone on diverse bimanual manipulation tasks.

On LIBERO, CoGeo-WAM achieves the best average success rate of 98.7%, improving the FastWAM-Joint backbone from 97.7% by 1.0 point. It achieves the best Spatial-suite result of 99.2% and tied-best results on the Object and Goal suites, reaching 99.8% and 98.0%, respectively. CoGeo-WAM also improves the Long-suite result from 95.6% to 97.6%. Together, these results support the applicability of CoGeo-WAM across different WAM backbones and manipulation benchmarks.

Real-world results. Table 3 summarizes the results on the four Franka manipulation tasks. CoGeo-WAM achieves the highest average success rate of 83.75%, exceeding FastWAM-Joint and $\pi_{0.5}$ by 6.25 and 10.00 percentage points, respectively. Compared with FastWAM-Joint, CoGeo-WAM improves success rates by 5, 15, and 5 percentage points on Cup Stacking, Block Stacking, and Water Pouring, respectively, while maintaining the same 95% success rate on Bread Placement. The largest gains occur in sequential block stacking and pouring, which require coordinated object positioning and orientation control. These results complement our simulation evaluation and support the effectiveness of controllable geometry adaptation on a physical robot, without explicit geometric inputs or an external 3D vision model at inference.

5.3 Analyses and Ablation Studies

Alignment-layer selection. We identify the alignment layer through a geometry-compatibility probe over intermediate DiT representations. For each candidate block, we freeze both the WAM and VGGT and train an alignment projector with the same architecture to map the block's video features into the spatiotemporally matched VGGT reference space. We report the best validation cosine similarity after projector convergence. This score measures the directional agreement between projected WAM features and their corresponding VGGT geometry references; a higher value indicates that geometry-relevant information can be more faithfully read out into the VGGT feature space.

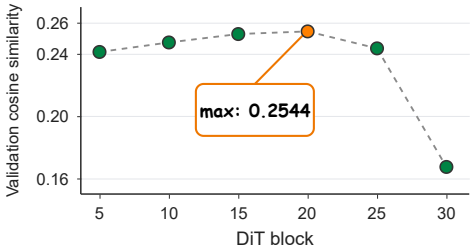

Figure 4: Geometry-compatibility probe across DiT blocks.

Figure 4 shows that cosine similarity increases from the 5th to the 20th DiT block, reaching a maximum of 0.2544, before declining in subsequent blocks and dropping substantially at the final block. We therefore select the 20th DiT block for implicit geometry alignment in all experiments. Under the same restricted-capacity projector, this result suggests that block 20 provides the most readily accessible geometry-relevant representation among the examined layers.

Table 4: Geometry-controller ablation on selected RoboTwin 2.0 tasks. We report task success rates (%) under clean and randomized settings. Uniform Geometry uses the Stage-I LoRA residual without control. Best results are highlighted in bold.

Variant	*Open Microwave*			*Dump Bin Bigbin*			*Place Bread Basket*			*Turn Switch*			Avg.
	Clean	Random	Avg.	Clean	Random	Avg.	Clean	Random	Avg.	Clean	Random	Avg.	
LingBot-VA	64	86	75.0	93	97	95.0	**97**	92	94.5	42	43	42.5	76.8
Uniform Geometry	80	**90**	85.0	90	91	90.5	95	89	92.0	71	**61**	66.0	83.4
Global	**89**	86	**87.5**	95	**98**	**96.5**	96	90	93.0	71	54	62.5	84.9
Block (Ours)	85	**90**	**87.5**	**96**	94	95.0	93	94	93.5	**77**	57	**67.0**	**85.8**
Token	82	87	84.5	93	97	95.0	94	93	93.5	70	55	62.5	83.9
Token-Rank	82	89	85.5	95	97	96.0	96	**95**	**95.5**	71	51	61.0	84.5
Context-Rank	85	84	84.5	92	97	94.5	96	93	94.5	65	60	62.5	84.0

Control granularity and conditioning ablation. We ablate our Block-Conditioned controller by varying the granularity of geometry modulation and the information used to predict its strength. Our default design predicts one scalar gate per DiT block from pooled hidden states, block identity, and flow time (Sec. 4.3). Let l and n index blocks and tokens, respectively, and let $c = [e_{\text{text}}, e_{\text{obs}}, e_{\text{state}}, e_\tau]$ denote pooled text, observation, and proprioceptive representations and the flow-time embedding. All variants use exponential mapping to produce positive gates.

Global Context. We replace block-specific modulation with a single scalar gate $g = \exp(f_\phi(c))$ shared across all blocks. The gate uses global context instead of block hidden states and identity, testing whether sample-level control is sufficient. *Token-Conditioned.* We remove hidden-state pooling and predict a scalar gate for each token, $g_{l,n} = \exp(f_\phi([h_{l,n}, e_l, e_\tau]))$. This retains block and flow-time conditioning while allowing geometry strength to vary within a block. *Token-wise Rank.* We further replace the token-level scalar with a rank-dimensional gate $g_{l,n} \in \mathbb{R}^r$, keeping the Token-Conditioned inputs. Inspired by TopLoRA (Li et al., 2025), the gate scales individual LoRA directions through $\Delta y_{l,n} = B_l[g_{l,n} \odot (A_l h_{l,n})]$, testing finer control within the low-rank update. *Context-Enriched Rank.* Starting from Token-wise Rank, we replace the explicit block embedding with global context and normalize the controller output: $q_{l,n} = f_\phi([h_{l,n}, c])$ and $g_{l,n} = \exp(\text{RMSNorm}(q_{l,n}))$. This tests a richer conditioning design for rank-wise modulation.

We evaluate these variants on four representative RoboTwin 2.0 tasks covering articulated-object operation (*Open Microwave*), container-to-container pouring (*Dump Bin Bigbin*), object-to-receptacle placement (*Place Bread Basket*), and localized point-contact actuation (*Turn Switch*). We compare against LingBot-VA and Uniform Geometry, which directly applies the Stage I geometry-aligned LoRA residual without a controller. Uniform Geometry improves *Open Microwave* and *Turn Switch* but reduces performance on the other two tasks relative to LingBot-VA, indicating that fixed geometry adaptation is not equally beneficial across tasks.

All five controllers outperform both baselines in average success rate. Their performance generally approaches the stronger baseline on each task and, in several cases, exceeds both, supporting the benefit of adaptively regulating geometry adaptation. The consistent average gains across all five variants suggest that this benefit is robust to the tested choices of control granularity and conditioning. Block-Conditioned achieves the highest average success rate of 85.8%, indicating that finer-grained modulation does not necessarily improve performance. We therefore adopt it as the default controller in all other experiments.

Stage-dependent geometry control. Figure 5 illustrates how geometry modulation changes across the execution stages of *Open Microwave*. Snapshots at control chunks 0, 3, and 8 show target localization, handle engagement, and articulated pulling, respectively. The curves report $100(g_{l,t} - \bar{g}_l)$, where $\bar{g}_l$ is each block's mean gate over the rollout. During target localization, identifying the handle and its spatial relationship to the gripper requires geometric reasoning. Consistent with this demand, gates in all four displayed blocks are generally above their respective rollout means. Once the handle is engaged, motion is increasingly constrained by the microwave's articulated structure, potentially reducing the need for additional geometry adaptation. The gates generally decrease toward or below

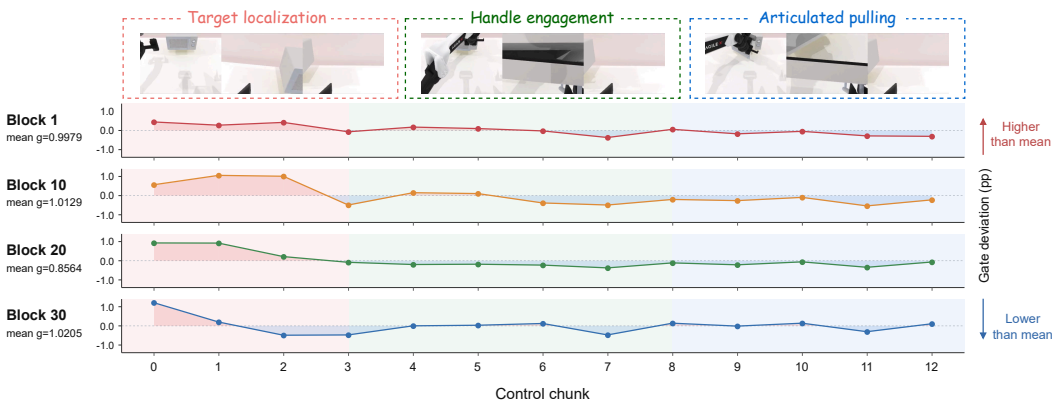

Figure 5: Stage-dependent geometry control on *Open Microwave*. Top: three-view observations at chunks 0, 3, and 8 (left to right), illustrating target localization, handle engagement, and articulated pulling. Bottom: gate deviations $100(g_{l,t} - \bar{g}_l)$, where $\bar{g}_l$ is the rollout mean for block l. Shaded regions indicate the corresponding task stages; positive and negative deviations indicate stronger and weaker geometry contributions relative to each block's rollout mean.

their means during handle engagement and articulated pulling. This pattern is consistent with the controller adjusting geometry strength to the changing demands of task execution.

The ablation results in Table 4 provide complementary quantitative evidence. Uniform Geometry increases the average success rate on *Open Microwave* from 75.0% to 85.0%, demonstrating the benefit of geometry adaptation. Adaptive control further improves it to 87.5%, supporting the value of regulating the learned adaptation during execution.

Video prediction quality. We investigate whether CoGeo-WAM improves video prediction alongside downstream control performance. Consistent with **Control granularity and conditioning ablation**, we compare CoGeo-WAM with the LingBot-VA baseline and evaluate uniform geometry adaptation to isolate the contribution of adaptive geometry control. We evaluate 100 videos per method from the same four tasks, yielding 300 videos in total. PSNR and SSIM measure pixel-level fidelity and structural similarity to corresponding simulator observations. We also report VBench's Subject Consistency and Dynamic Degree (Huang et al., 2024) to characterize temporal appearance consistency and motion activity. Together, these metrics provide complementary evidence to the control-success evaluation.

Table 5: Video prediction quality. Sub/Dyn: Subject Consistency/Dynamic Degree (%). PSNR: dB. Best/second-best: bold/underlined.

Method	PSNR ↑	SSIM ↑	Sub ↑	Dyn ↑
LingBot-VA	<u>26.11</u>	0.914	80.56	76
Uniform Geometry	25.79	<u>0.915</u>	**82.33**	<u>98</u>
CoGeo-WAM	**26.33**	**0.921**	<u>82.24</u>	**99**

Uniform geometry adaptation improves Subject Consistency from 80.56 to 82.33 and increases Dynamic Degree from 76 to 98. CoGeo-WAM achieves the highest PSNR (26.33) and SSIM (0.921), while maintaining comparable Subject Consistency (82.24) and a high Dynamic Degree (99). These results suggest that selectively regulating geometric information benefits prediction fidelity while preserving temporal appearance consistency and motion activity, complementing the improvements in downstream control.

6 Conclusion

We presented CoGeo-WAM, a framework for controllable geometry adaptation in WAMs. Implicit geometry alignment transfers 3D-aware knowledge into intermediate video representations through a LoRA residual branch, while adaptive geometry control regulates its contribution according to the current context. This design requires neither explicit geometric inputs nor an external 3D vision model at inference. Experiments across simulation and real-world tasks with different WAM backbones demonstrate the effectiveness of combining geometry acquisition with adaptive utilization for robot manipulation.

AI USE STATEMENT

In this work, we used generative AI tools, primarily OpenAI's GPT-5.6 Sol and GPT-5.5, for language polishing, translation, formatting, and supporting programming tasks such as data processing, debugging, and auxiliary script generation. All AI-assisted content was manually reviewed and verified by the authors, who take full responsibility for the final manuscript and research artifacts.

ETHICS STATEMENT

This work studies geometry-aware robot manipulation using simulation benchmarks and real-world tabletop experiments on a Franka robot. The intended application is to improve the reliability of robot interaction with physical objects. Nevertheless, improved benchmark performance does not establish safety in unrestricted environments. Errors in perception or action prediction can cause collisions or object damage, so deployment requires appropriate supervision, independent safety mechanisms, and validation for the intended operating conditions. The method also inherits potential biases and limitations from its pretrained models and training data.

REPRODUCIBILITY STATEMENT

We describe the two-stage training procedure of CoGeo-WAM, including geometry-reference construction, projector calibration, LoRA-based alignment, and adaptive geometry control. We specify the WAM backbones, alignment layer, projector architecture, and loss coefficients, together with the benchmarks and evaluation protocols for simulation and real-world experiments. We further report ablations on alignment-layer selection and controller designs to clarify the implementation choices. These details support independent reproduction and assessment of our method.

REFERENCES

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, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Conference on Robot Learning*, 2025. URL https://openreview.net/forum?id=vlhoswksBO.

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.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *International Conference on Machine Learning*, 2026.

Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. RVT-2: Learning precise manipulation from few demonstrations. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.055. URL https://www.roboticsproceedings.org/rss20/p055.html.

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.

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.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Wenlong Huang, Yu-Wei Chao, Arsalan Mousavian, Ming-Yu Liu, Dieter Fox, Kaichun Mo, and Li Fei-Fei. PointWorld: Scaling 3D world models for in-the-wild robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20765–20779, June 2026.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. In *International Conference on Learning Representations*, volume 2026, pp. 71531–71552, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/748becc400a57c0e31cfe6a2e7951467-Abstract-Conference.html.

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 *International Conference on Learning Representations*, volume 2026, pp. 132324–132345, 2026a.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. In *Proceedings of Robotics: Science and Systems*, 2026b. doi: 10.15607/RSS.2026.XXII.016. URL https://www.roboticsproceedings.org/rss22/p016.html.

Shiwei Li, Xiandi Luo, Haozhao Wang, Xing Tang, Ziqiang Cui, Dugang Liu, Yuhua Li, Xiuqiang He, and Ruixuan Li. Beyond higher rank: Token-wise input-output projections for efficient low-rank adaptation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 136414–136435, 2025. doi: 10.52202/085713-4552. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/c74429a930838b5bac03c681f0a0ce11-Abstract-Conference.html.

Yue Liao, Pengfei Zhou, Siyuan Huang, Donglin Yang, Shengcong Chen, Yuxin Jiang, Yue Hu, Si Liu, Jianlan Luo, Liliang Chen, et al. Genie envisioner: A unified world foundation platform for robotic manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 88446–88463, 2026.

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

Yushan Liu, Peibo Sun, Shoujie Li, Yifan Xie, Lingfeng Zhang, Xintao Chao, Shiyuan Dong, Fang Chen, Xiao-Ping Zhang, and Wenbo Ding. Oa-wam: Object-addressable world action model for robust robot manipulation. *arXiv preprint arXiv:2605.06481*, 2026.

Guanxing Lu, Baoxiong Jia, Puhao Li, Yixin Chen, Ziwei Wang, Yansong Tang, and Siyuan Huang. Gwm: Towards scalable gaussian world models for robotic manipulation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9263–9274. IEEE, 2025.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Jiayuan Gu, Zhigang Wang, Yan Ding, Bin Zhao, Dong Wang, and Xuelong Li. SpatialVLA: Exploring spatial representations for visual-language-action models. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.011. URL https://www.roboticsproceedings.org/rss21/p011.html.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025.

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*, 2026.

Haoyu Wu, Diankun Wu, Tianyu He, Junliang Guo, Yang Ye, Yueqi Duan, and Jiang Bian. Geometry forcing: Marrying video diffusion and 3d representation for consistent world modeling. In *International Conference on Learning Representations*, volume 2026, pp. 119186–119209, 2026.

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.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D diffusion policy: Generalizable visuomotor policy learning via simple 3D representations. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.067. URL https://www.roboticsproceedings.org/rss20/p067.html.

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3D-VLA: A 3D vision-language-action generative world model. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61229–61245. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhen24a.html.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 60580–60606, 2026.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.015. URL https://www.roboticsproceedings.org/rss21/p015.html.

APPENDIX

A QUALITATIVE COMPARISONS

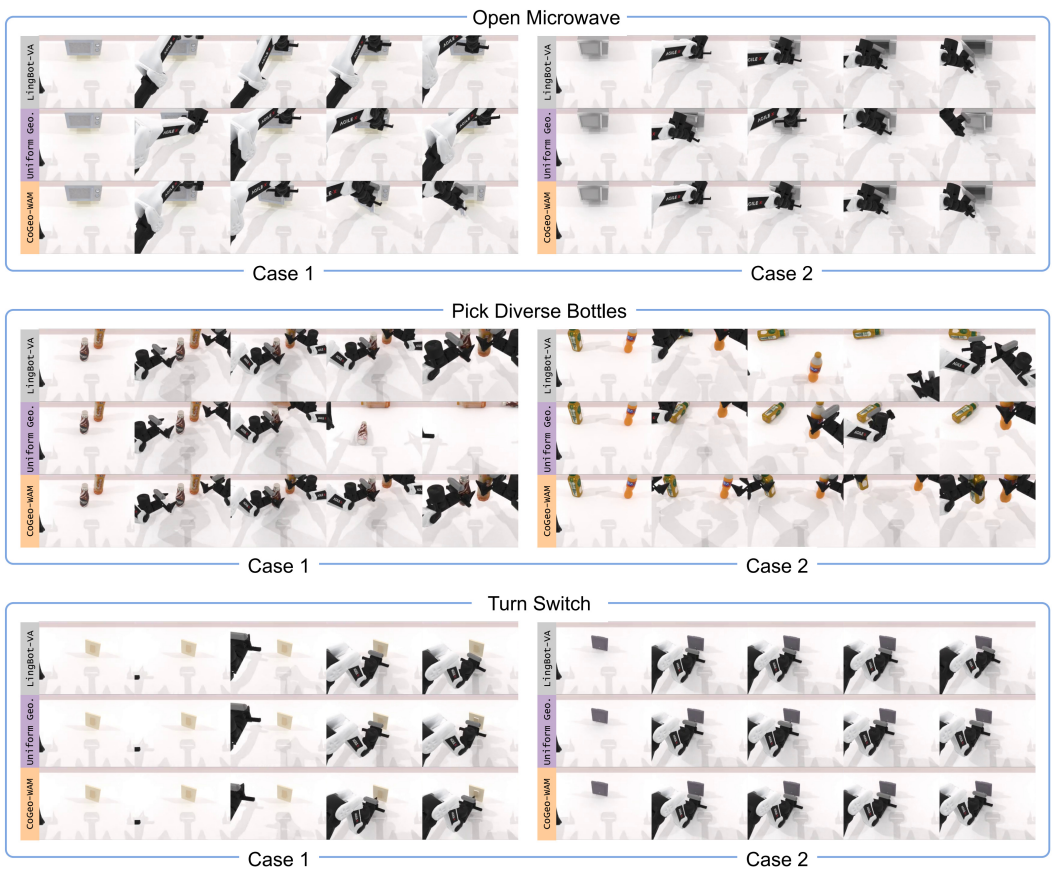

Figure 6: Qualitative comparisons on three RoboTwin 2.0 tasks, with two cases per task. Each case compares LingBot-VA, Uniform Geometry, and CoGeo-WAM in the same initial scene with the same objects and task. Third-person observations show execution progress from left to right; frames are selected independently for each rollout. Uniform Geometry uses the Stage-I LoRA residual without adaptive control.

Figure 6 compares LingBot-VA, Uniform Geometry, and CoGeo-WAM on *Open Microwave*, *Pick Diverse Bottles*, and *Turn Switch*. For each task, we present two cases, comparing all three methods in the same initial scene with the same objects and task. These comparisons illustrate how the utility of geometry adaptation depends on the interaction requirements and execution stage. Outcomes follow the original evaluation records.

Open Microwave. Opening a microwave involves distinct interaction stages. The robot must first localize the handle and position its gripper for effective engagement. Once contact is established, it must follow the door's hinge-constrained trajectory while maintaining engagement. Accurate spatial localization is particularly important during approach, whereas subsequent motion is more constrained by the articulated mechanism. The need for additional geometry adaptation may therefore decrease as the interaction progresses.

In Case 1, both LingBot-VA and Uniform Geometry fail to complete the opening operation, whereas CoGeo-WAM successfully pulls the door open. In Case 2, LingBot-VA succeeds but Uniform Geometry fails, while CoGeo-WAM retains successful execution. These contrasting outcomes suggest

that uniformly strengthening geometric representations is insufficient for coordinating the entire interaction. Adaptive control allows the geometry-aligned residual to vary as the policy transitions from handle localization to articulated pulling, consistent with the stage-dependent gate patterns discussed in the main text.

Pick Diverse Bottles. Picking up bottles requires coordinating object localization with gripper opening and approach direction. The gripper must provide sufficient clearance around each bottle before closing at a suitable grasping position. A positioning error or an insufficient opening can cause the fingers to collide with and topple the bottle, while an unstable grasp can lead to loss of the object during lifting. Successful execution thus requires spatial precision together with appropriate gripper control.

In Case 1, LingBot-VA and CoGeo-WAM successfully lift both bottles, whereas Uniform Geometry loses a bottle during execution. In Case 2, both comparison methods leave a bottle lying on the table, while CoGeo-WAM lifts both objects successfully. These failures highlight the importance of coordinating approach, gripper clearance, and grasp maintenance rather than strengthening geometric representations alone. The successful CoGeo-WAM rollouts illustrate how adaptive control can help integrate geometry-aligned information with useful pretrained grasping behavior.

Turn Switch. Switch activation requires precise localization of a small actuation region and contact in an appropriate direction. Reaching the vicinity of the switch is not sufficient: a small contact offset can leave it inactive even when the gripper appears close to its surface.

In Case 1, LingBot-VA approaches the switch but fails to activate it, whereas both Uniform Geometry and CoGeo-WAM succeed. This comparison illustrates the benefit of geometry-aligned adaptation for contact-sensitive manipulation. In Case 2, only CoGeo-WAM succeeds, showing that uniform adaptation does not resolve every contact configuration. The successful actuation is consistent with the intended role of adaptive control in adjusting the geometry-aligned residual to the current interaction context.

Together, these cases illustrate complementary requirements for precise contact, articulated motion, and stable grasping. They support adapting the contribution of geometric knowledge to the interaction context rather than applying it uniformly.

B Inference Memory Footprint

We compare the inference GPU memory footprint of LingBot-VA and its CoGeo-WAM counterpart in Table 6. CoGeo-WAM increases resident memory by only 0.70% and peak memory by 0.76%, remaining close to the original backbone. The frozen 3D vision model and alignment projector are not used at inference; only the learned LoRA residuals and controller are retained. These measurements indicate that controllable geometry adaptation introduces little additional inference memory usage in the evaluated configuration.

Table 6: Inference GPU memory usage of LingBot-VA and CoGeo-WAM. Both resident and peak memory increase by less than 1%.

Metric	LingBot-VA	CoGeo-WAM	Increase
Resident memory (MiB)	16675.82	16792.94	0.70%
Peak memory (MiB)	17294.23	17425.94	0.76%

C Per-Task Results on RoboTwin 2.0

Table 7 reports success rates on all 50 RoboTwin 2.0 tasks under clean and randomized settings. CoGeo-WAM uses LingBot-VA as its backbone. We report our evaluations of LingBot-VA and CoGeo-WAM alongside published results for the other methods.

Table 7: Per-task success rates (%) on all 50 RoboTwin 2.0 tasks under clean and randomized settings.

Task	$\pi_{0.5}$		X-VLA		Motus		X-WAM		LingBot-VA		CoGeo-WAM (Ours)	
	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.	Clean	Rand.
Adjust Bottle	79	83	**100**	**99**	89	93	**100**	**99**	97	96	97	97
Beat Block Hammer	63	50	92	88	95	88	**98**	96	**98**	**99**	96	97
Blocks Ranking RGB	43	35	83	83	**99**	97	**99**	95	97	**99**	98	95
Blocks Ranking Size	8	14	67	74	75	63	76	82	**98**	89	92	**91**
Click Alarmclock	97	93	99	99	**100**	**100**	98	99	**100**	**100**	**100**	99
Click Bell	75	76	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
Dump Bin Bigbin	30	42	79	77	95	91	90	96	93	**97**	**96**	94
Grab Roller	90	89	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
Handover Block	18	19	73	37	86	73	88	79	**100**	**94**	98	78
Handover Mic	28	18	0	0	78	63	88	89	88	87	**91**	**92**
Hanging Mug	3	3	23	27	38	38	**46**	**55**	25	32	41	34
Lift Pot	0	0	99	**100**	96	99	**100**	99	**100**	99	**100**	**100**
Move Can Pot	29	27	89	86	34	74	80	84	94	88	**97**	**92**
Move Pillbottle Pad	33	29	73	71	93	96	96	98	**100**	**100**	99	98
Move Playingcard Away	59	67	93	**98**	**100**	96	**100**	**98**	**100**	**98**	**100**	**98**
Move Stapler Pad	16	18	78	73	**83**	**85**	67	70	74	67	69	73
Open Laptop	19	35	93	**100**	95	91	96	97	**97**	95	96	98
Open Microwave	35	37	79	71	**95**	91	89	**92**	64	86	85	90
Pick Diverse Bottles	5	3	58	36	90	91	91	**92**	**95**	87	93	76
Pick Dual Bottles	10	6	47	36	96	90	99	**100**	99	**100**	**100**	92
Place A2B Left	62	60	48	49	88	79	90	87	**97**	**91**	96	89
Place A2B Right	62	57	36	36	91	87	92	89	**96**	90	93	**93**
Place Bread Basket	48	56	81	71	91	**94**	90	91	**97**	92	93	**94**
Place Bread Skillet	38	46	77	67	86	83	90	**96**	**96**	**96**	95	93
Place Burger Fries	66	70	94	94	**98**	98	97	**99**	96	97	93	96
Place Can Basket	19	25	49	52	81	76	84	82	**87**	**88**	82	81
Place Cans Plasticbox	40	47	97	98	98	94	99	98	99	**99**	**100**	**99**
Place Container Plate	71	78	97	95	98	99	98	**100**	98	97	**99**	98
Place Dual Shoes	12	7	79	88	**93**	87	83	81	87	81	**93**	**94**
Place Empty Cup	75	86	**100**	98	99	98	98	99	**100**	**100**	**100**	**100**
Place Fan	25	36	80	75	91	87	84	**92**	**93**	84	91	83
Place Mouse Pad	21	26	70	70	66	68	84	86	**98**	**95**	96	90
Place Object Basket	43	36	44	39	81	**87**	85	**87**	87	86	**89**	81
Place Object Scale	40	49	52	74	88	85	93	89	**99**	94	98	**95**
Place Object Stand	74	65	86	88	98	97	97	96	**100**	94	**100**	**99**
Place Phone Stand	49	53	88	87	87	86	75	80	**97**	**95**	92	**95**
Place Shoe	57	61	96	95	**99**	97	97	**99**	98	**99**	97	97
Press Stapler	80	70	92	**98**	93	**98**	**94**	90	77	88	83	87
Put Bottles Dustbin	12	9	74	77	81	79	**85**	**95**	65	81	67	87
Put Object Cabinet	24	15	46	48	**88**	71	66	76	83	**86**	81	79
Rotate QRcode	47	56	34	33	89	73	84	83	90	93	**94**	**94**
Scan Object	42	38	14	36	67	66	86	79	**96**	**92**	95	91
Shake Bottle	91	**100**	99	**100**	**100**	97	99	99	**100**	**100**	99	99
Shake Bottle Horizontally	96	**100**	**100**	**100**	**100**	98	**100**	99	**100**	99	99	98
Stack Blocks Three	15	16	6	10	91	95	97	95	99	96	**100**	**98**
Stack Blocks Two	48	56	92	87	**100**	98	**100**	**100**	**100**	96	**100**	**100**
Stack Bowls Three	33	35	76	86	79	**87**	**88**	82	78	84	79	80
Stack Bowls Two	78	66	96	93	**98**	**98**	**98**	**98**	97	97	94	**98**
Stamp Seal	36	23	76	82	93	92	93	**95**	97	92	**99**	90
Turn Switch	5	6	40	61	**84**	**78**	61	72	42	43	77	57

D REAL-WORLD EXPERIMENTAL PLATFORM

We conduct real-world experiments using a Franka Research 3 (FR3) manipulator with seven degrees of freedom, a rated payload of 3 kg, and a maximum reach of 855 mm. The arm incorporates torque sensors in all seven joints. Our evaluation covers four tabletop tasks: *Cup Stacking*, *Block Stacking*, *Bread Placement*, and *Water Pouring*, which involve object alignment, sequential placement, and orientation control. We implement CoGeo-WAM on FastWAM-Joint and compare it with the original backbone and $\pi_{0.5}$, evaluating each method over 20 trials per task.