GeoScale: Inference-Time Scaling of Geometry-Aware Video World Models for Robot Manipulation

Anonymous authors
Paper under double-blind review

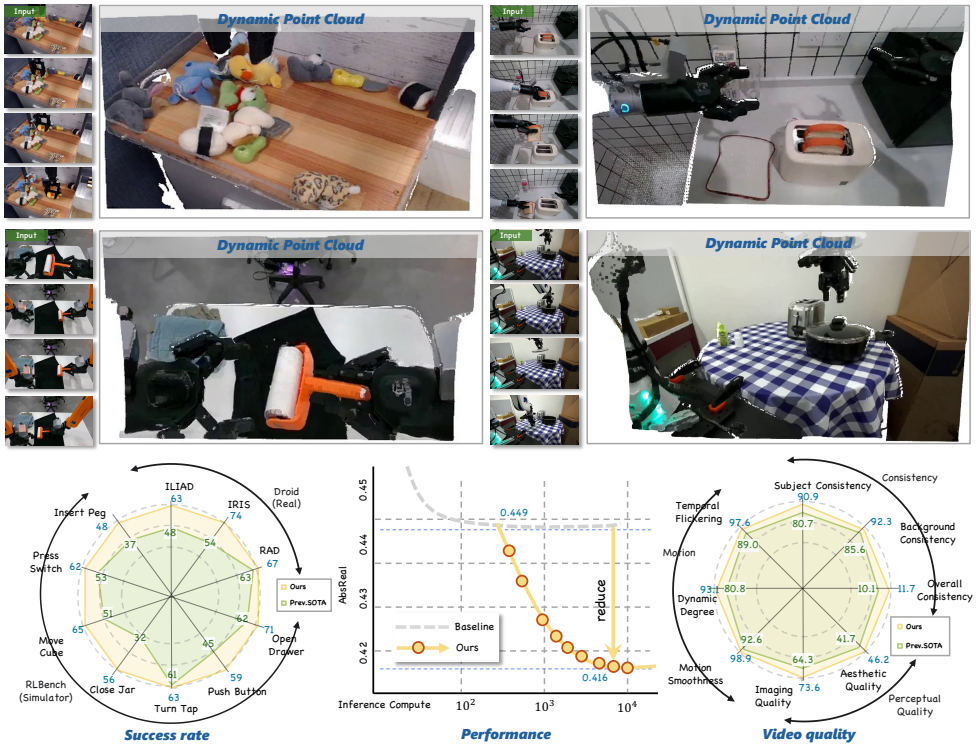

Figure 1: **Showcase of GeoScale.** Given a first-frame observation and a language instruction, GeoScale improves video world-model rollouts at inference time by scoring geometric consistency in 4D. The selected rollout preserves coherent depth, point correspondence, and object motion, enabling more reliable recovery of executable robot trajectories.

Abstract

Video world models provide a promising interface for robot manipulation by predicting future visual states that can be converted into actions. Geometric consistency and motion plausibility in these predictions are critical for recovering executable action trajectories, as robots must carry out these actions in the physical world. We observe that different random seeds at inference time produce rollouts with substantially different geometric quality, suggesting that better predictions can be obtained by searching over seeds. We therefore propose **GeoScale**, an inference-time scaling framework that searches for seeds yielding geometrically stronger rollouts. GeoScale uses feed-forward 4D reconstruction as geometric feedback for both candidate selection and guidance during denoising. It progressively scores and prunes candidate rollouts to concentrate computation on promising candidates, while applying differentiable geometric guidance to refine the surviving candidates at selected denoising steps. Experiments on real-world Droid and simulated RLBench show consistent gains in depth accuracy, 3D reconstruction, and point correspondence while preserving visual quality. GeoScale also

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.

improves human-rated task completion and simulator execution success, demonstrating more actionable rollouts for robot manipulation.

1 INTRODUCTION

Video world models (Yang et al., 2025b; Wan et al., 2025; Xing et al., 2024; Shi et al., 2024) have demonstrated strong capabilities in predicting realistic future visual states conditioned on current observations and language instructions. These visual predictions can be converted into robot actions through inverse dynamics or pose estimation (Du et al., 2023; Zhou et al., 2026), supporting robot manipulation planning. By drawing on the transferable priors and instruction-following capabilities of pretrained video world models, this paradigm can reduce dependence on extensive robot data collection and improve generalization to unseen tasks and environments (Ye et al., 2026).

However, visually plausible predictions do not guarantee accurate action recovery for reliable execution in physical environments. Objects may deform, contacts may drift, depth may change discontinuously, and pixels corresponding to the same 3D surface point may become inconsistent across time. Although these errors can be subtle in RGB space, they can severely corrupt the 6-DoF object or end-effector trajectories recovered from a generated rollout (Du et al., 2023; Zhou et al., 2026). To improve geometric consistency, some methods incorporate 3D scene flow into their predictions (Zhi et al., 2025), while others distill 4D correspondence representations into video generation backbones (Zhou et al., 2026). These approaches demonstrate the importance of geometry, but require geometry-aware data or teacher supervision and must be retrained when the generator changes. In parallel, inference-time methods seek to better exploit the capabilities of pretrained video generators without updating their parameters. These methods improve generation through ***seed search*** (Ma et al., 2025), which selects promising candidates initialized with different random seeds, or ***reward-guided sampling***, which steers the denoising process using reward feedback (Yuan et al., 2026).

In this work, we aim to improve the geometric consistency of video world models without additional training. Toward this goal, our analysis reveals two key observations regarding seed selection and the timing of geometric guidance, as illustrated in Fig. 2:

(1) ***Seed-dependent geometry***: Given the same initial observation and language instruction, changing only the initial noise can produce rollouts with markedly different depth consistency and 3D reconstruction quality. This variation indicates that geometrically strong futures already exist within the frozen generator's sampling distribution, motivating a search for random seeds that yield more geometrically consistent rollouts.

(2) ***Early-stage inefficiency***: Early predicted-clean videos contain insufficiently stable scene structure for reliable 4D feedback. Applying expensive decoding, reconstruction, and gradient guidance at these stages adds substantial overhead for only marginal geometric gains. Guidance should therefore focus on later, more informative denoising stages.

Motivated by these observations, we introduce **GeoScale**, an inference-time scaling framework that improves the geometric consistency of rollouts from a frozen video generator through progressive seed search and selective geometric guidance for robotic manipulation. Specifically, to exploit seed-dependent geometry, we introduce a 4D consistency reward built on feed-forward 4D reconstruction to evaluate static reprojection consistency, dynamic trajectory smoothness, and global motion regularity, together with a progressive Best-of-N strategy that retains promising candidates across denoising stages. To avoid unproductive early optimization, geometric guidance applies differentiable 4D feedback only at informative denoising stages and only to surviving trajectories. Together, search and selective guidance allocate test-time compute across candidates and timesteps, yielding more geometrically coherent rollouts for downstream action recovery without fine-tuning.

We evaluate GeoScale on real-world Droid (Khazatsky et al., 2024) manipulation sequences and simulated RLBench (James et al., 2020) tasks. We measure 4D scene generation quality, including RGB fidelity, depth accuracy, 3D reconstruction, and point correspondence. We further evaluate human-perceived task completion on Droid and simulator execution success after converting generated videos into 6-DoF trajectories with a FoundationPose-based (Wen et al., 2024) video-to-action pipeline on RLBench. Overall, GeoScale improves geometric consistency, human-rated task completion, and simulated execution success over embodied video-generation baselines.

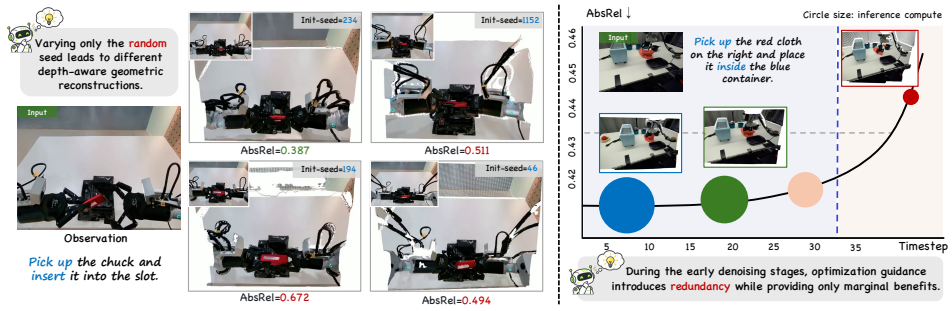

Figure 2: **Motivation for inference-time geometric scaling.** ***Left:*** Given the same initial observation and language instruction, different random seeds produce substantially different geometric reconstructions. This variation motivates searching for seeds that produce geometrically stronger rollouts from the same frozen generator. ***Right:*** Early guidance adds computation with marginal gains, motivating guidance only when predictions provide useful 4D feedback.

Our contributions are summarized as follows:

- We analyze the properties of geometric consistency in video world models for robotic manipulation and propose **GeoScale**, an inference-time scaling framework that improves the geometric consistency of frozen video generators without additional training.
- We introduce a set of geometry-aware inference mechanisms: a *4D consistency reward* for scoring reconstruction quality, *progressive Best-of-N search* for using test-time sampling budget efficiently, and *geometric optimization guidance* for injecting differentiable 4D feedback.
- We perform end-to-end validation from video world modeling to robot control, outperforming the strongest baseline across all evaluated Droid categories and RLBench tasks, with additional qualitative validation on a physical robot.

2 Related Work

Video World Models for Embodied Control. Video world models support embodied decision-making by predicting task-conditioned future observations that can be used for planning or policy extraction. Prior systems recover actions from generated visual rollouts (Du et al., 2023) or use generated interaction futures for downstream policy learning (Yang et al., 2023; Bruce et al., 2024), while recent large video generators provide strong open-domain visual priors (Yang et al., 2025b; Wan et al., 2025; Xing et al., 2024; Shi et al., 2024). However, these models often optimize visual plausibility without enforcing 3D structure or inter-frame correspondence. Recent geometry-aware models address this gap by predicting geometric outputs (Zhen et al., 2025; Zhi et al., 2025) or distilling 4D correspondence into the video backbone (Zhou et al., 2026; Wu et al., 2026). In contrast, GeoScale keeps the video generator frozen and injects geometry only at inference time, using feed-forward 4D reconstruction as a reward signal rather than as a training target.

Inference-Time Alignment for Diffusion Models. Inference-time alignment steers a pretrained diffusion model toward task-specific objectives without updating model parameters (Uehara et al., 2025). Existing methods typically use derivative-free candidate search, such as Best-of-N or resampling-based selection (Ma et al., 2025; Singhal et al., 2025; Li et al., 2024), or gradient-based guidance from differentiable objectives (Bansal et al., 2023; Ye et al., 2024). WMReward (Yuan et al., 2026) applies this idea to video dynamics by rewarding consistency under a latent predictive world model. GeoScale instead evaluates generated videos with explicit 4D geometric reconstruction and correspondence signals, making inference-time alignment directly target the spatial consistency needed for robot execution.

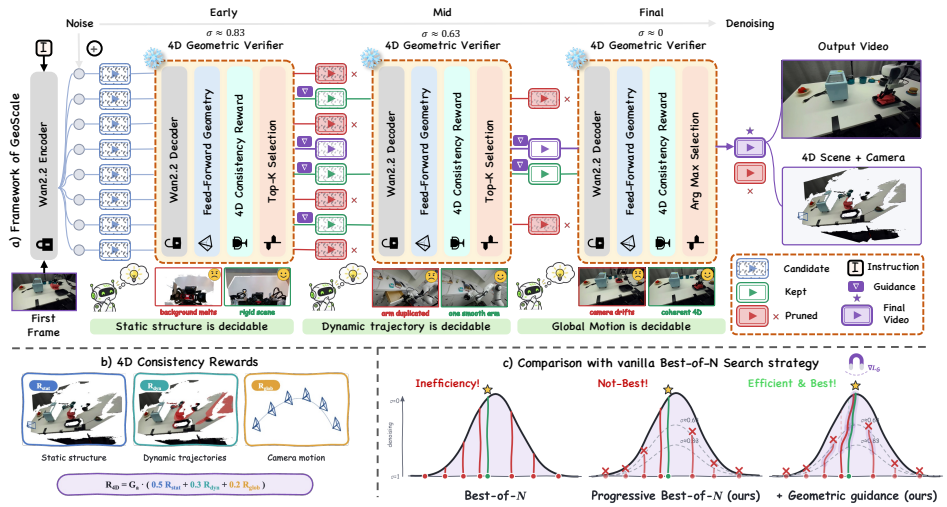

Figure 3: **Overview of GeoScale.** (a) A frozen video generator denoises N candidate futures. The predicted clean videos are scored by a frozen feed-forward 4D verifier, the population is halved ($8 \to 4 \to 2 \to 1$), and survivors are refined by gradient-based geometric guidance (∇). Checkpoints sit where static structure, then dynamic trajectories, then the full 4D scene become decidable. (b) The 4D consistency reward combines static, dynamic and camera terms gated by first-frame geometry. (c) Compared with vanilla Best-of-N, progressive pruning stops denoising discarded candidates early, and guidance pulls the finalists toward the mode of 4D-consistent rollouts.

3 METHOD

Given an initial observation I_0 and a language instruction ℓ, our goal is to generate a future video $\mathbf{V} \in \mathbb{R}^{T\times H\times W\times 3}$ that follows the instruction and admits a geometrically consistent 4D interpretation. We introduce **GeoScale**, an inference-time scaling framework that uses reconstructed 4D geometry to select and refine candidate futures from frozen generators at inference time. Rather than updating model parameters, GeoScale uses geometric feedback to allocate computation across candidates and refine individual sampling trajectories for downstream robot manipulation.

The framework addresses two questions: how to evaluate the geometric consistency of a generated video, and how to use this evaluation during generation. We first introduce a *4D Consistency Reward* (Sec. 3.1) that evaluates static structure, dynamic trajectories, and camera motion. We then use this reward in two complementary ways. *Progressive Best-of-N Search* (Sec. 3.2) prunes candidates during denoising, while *Geometric Optimization Guidance* (Sec. 3.3) refines sampling trajectories using a differentiable geometric loss. The video generator and geometric verifier remain frozen throughout inference. Implementation details are provided in App. B.

3.1 4D CONSISTENCY REWARD

A geometrically consistent video must distinguish persistent scene structure from object and camera motion. Background points should remain stationary in world coordinates, whereas foreground objects may undergo task-dependent motion. We therefore evaluate static structure, dynamic trajectories, and viewpoint evolution separately to avoid conflating scene deformation with legitimate movement.

Geometric Verifier and Notation. We apply a frozen feed-forward 4D reconstruction model, 4RC (Luo et al., 2026), to uniformly sampled frames of each candidate video. For each frame t, the verifier returns world-space points $\mathbf{x}_t(u)$, trajectories $\mathbf{y}_t(u)$ tracking first-frame pixels u, and camera rotations $\mathbf{R}_t$ and centers $\mathbf{c}_t$, all in a common world frame. We normalize translational residuals by a scene scale s estimated from the first frame. Verifier confidence only selects valid observations and does not enter the score. Trajectories with noticeable displacement from their initial positions define the dynamic set Ω_{d}; the remainder define the static set Ω_{s}. We write $\langle\cdot\rangle_\Omega$ for robust aggregation over

the corresponding valid observations. For uniformly sampled frames, Δ^2 denotes the second-order temporal difference, which measures velocity changes between consecutive intervals.

Static Structure Consistency. A static point reconstructed in one frame should agree with the geometry independently reconstructed in another. For a frame pair (i, j), let $d_{i\to j}(u)$ denote the depth of $\mathbf{x}_i(u)$ in camera j, and let $d_j(\pi_j(\mathbf{x}_i(u)))$ denote the reconstructed depth at its reprojected location. We measure their disagreement by:

$$E_{\mathrm{s}} = \left\langle \left| \log\left(d_{i\to j}(u) / d_j(\pi_j(\mathbf{x}_i(u))) \right) \right| \right\rangle_{\Omega_{\mathrm{s}}}. \tag{1}$$

The log-depth ratio is invariant to a common reconstruction scale. It tests whether cross-frame observations support a single static structure rather than plausible but inconsistent per-frame geometry.

Dynamic trajectory consistency. Moving foreground points should not be constrained to fixed world-space positions. Instead, we penalize abrupt changes in their reconstructed velocities:

$$E_{\mathrm{d}} = \left\langle \|\Delta^2 \mathbf{y}_t(u)\| / s \right\rangle_{\Omega_{\mathrm{d}}}. \tag{2}$$

This term discourages trajectory jitter without penalizing motion magnitude. It is a temporal regularizer rather than a dynamics model: legitimate accelerations also produce nonzero residuals.

Global motion regularity. We additionally regularize the recovered camera trajectory. Let $\theta_t = \angle(\mathbf{R}_{t+1}\mathbf{R}_t^\top)$ denote the rotation angle between consecutive cameras. We measure translation acceleration and changes in rotation-step magnitude as

$$E_{\mathrm{c}} = \left\langle \|\Delta^2 \mathbf{c}_t\| / s \right\rangle_t, \qquad E_{\mathrm{r}} = \left\langle |\theta_t - \theta_{t-1}| \right\rangle_t. \tag{3}$$

Unlike a stationary-camera constraint, these terms allow viewpoint changes while discouraging abrupt translation and rotation-speed variations.

Combined reward. We map the geometric errors to scores in $[0, 1]$:

$$R_{\mathrm{s}} = e^{-E_{\mathrm{s}}/\tau_{\mathrm{s}}}, \quad R_{\mathrm{d}} = e^{-E_{\mathrm{d}}/\tau_{\mathrm{d}}}, \quad R_{\mathrm{g}} = e^{-E_{\mathrm{c}}/\tau_{\mathrm{c}} - E_{\mathrm{r}}/\tau_{\mathrm{r}}}. \tag{4}$$

The positive temperatures τ_{s}, τ_{d}, τ_{c}, and τ_{r} control sensitivity to each error.

Let $\widetilde{R}_{\mathrm{s}}$, $\widetilde{R}_{\mathrm{d}}$, and $\widetilde{R}_{\mathrm{g}}$ denote the component scores after validity processing. The final reward is

$$R_{4\mathrm{D}}(\mathbf{V}) = G_{\mathrm{a}}\Big(\underbrace{\lambda_{\mathrm{s}}\widetilde{R}_{\mathrm{s}}}_{\text{static structure}} + \underbrace{\lambda_{\mathrm{d}}\widetilde{R}_{\mathrm{d}}}_{\text{dynamic tracks}} + \underbrace{\lambda_{\mathrm{g}}\widetilde{R}_{\mathrm{g}}}_{\text{camera motion}} \Big), \tag{5}$$

(Cross-frame depth agreement; Smooth viewpoint; 4D consistency reward; Trajectory smoothness)

where the nonnegative weights λ_{s}, λ_{d}, and λ_{g} control the component contributions, and $G_{\mathrm{a}} \in [0, 1]$ is a first-frame validity gate that suppresses degenerate reconstructions.

Component-specific validity processing suppresses under-observed, motionless, or teleporting reconstructions, as applicable. When a component has too few valid observations, it returns a neutral score rather than the maximum. Thus, an empty observation set cannot receive the best possible consistency score. The gates and fallback rules are specified in App. B. Higher rewards indicate lower geometric inconsistency under the verifier. The reward requires no ground-truth future geometry and does not measure instruction fulfillment or task success. Semantic conditioning is supplied by the video generator; the reward evaluates the geometry of its predictions.

3.2 Progressive Best-of-N Search

Standard Best-of-N sampling denoises all candidates to completion before comparison. GeoScale instead evaluates candidates during denoising and stops computation for trajectories with unreliable geometry. At denoising step k, candidate n has latent $z_k^{(n)}$, noise level σ_k, and predicted velocity $v_k^{(n)}$. Its predicted clean video is

$$\hat{z}_{0,k}^{(n)} = z_k^{(n)} - \sigma_k v_k^{(n)}, \qquad \widehat{\mathbf{V}}_k^{(n)} = \mathcal{D}(\hat{z}_{0,k}^{(n)}), \tag{6}$$

where $\mathcal{D}$ is the frozen video decoder. This provides an intermediate estimate of the geometry represented by the current sampling trajectory. Starting from N independent candidates, $\mathcal{A}_0 = \{1, \dots, N\}$, we evaluate all active candidates at each checkpoint k_j and retain the highest-scoring ones:

$$\mathcal{A}_j = \mathrm{TopK}_{n_j} \left\{ R_{4\mathrm{D}}\big(\widehat{\mathbf{V}}_{k_j}^{(n)}\big) \;\middle|\; n \in \mathcal{A}_{j-1} \right\}. \tag{7}$$

Candidates outside $\mathcal{A}_j$ are removed, while each survivor continues from its current latent and solver state. Because intermediate predictions are unreliable at high noise levels, pruning begins only after sufficient scene structure has emerged. Our default schedule progressively halves the active population. Checkpoint locations, survivor counts, and tie handling are described in App. B. After denoising, we re-evaluate the completed survivors and return:

$$\mathbf{V}^\star = \arg\max_{n \in \mathcal{A}_J} R_{4\mathrm{D}}(\mathbf{V}^{(n)}). \tag{8}$$

Progressive search therefore concentrates denoising computation on geometrically promising trajectories while accounting for the overhead of intermediate decoding and verification.

3.3 Geometric Optimization Guidance

Progressive search selects promising trajectories but does not directly improve them. We therefore use differentiable geometric feedback to refine each retained trajectory during denoising. The ranking reward contains robust aggregation and discrete validity operations that are unsuitable for optimization. We construct a smooth surrogate loss:

$$\mathcal{L}_{\mathrm{G}} = \omega_{\mathrm{s}} \mathcal{L}_{\mathrm{stat}} + \omega_{\mathrm{d}} \mathcal{L}_{\mathrm{dyn}} + \omega_{\mathrm{a}} \mathcal{L}_{\mathrm{anchor}}, \tag{9}$$

where $\mathcal{L}_{\mathrm{stat}}$ and $\mathcal{L}_{\mathrm{dyn}}$ are differentiable versions of the static and dynamic errors, and $\mathcal{L}_{\mathrm{anchor}}$ ties the first-frame reconstruction to the geometry of the conditioning image. Thus, the reward and guidance loss share the same geometric principles but serve different purposes: the former ranks candidates, while the latter provides stable local gradients.

At denoising step k, we differentiate the geometric loss with respect to the predicted clean latent:

$$\mathbf{g}_k = \nabla_{\hat{z}_{0,k}} \mathcal{L}_{\mathrm{G}} \left(\mathcal{D}(\hat{z}_{0,k}) \right). \tag{10}$$

We then update the predicted velocity as:

$$v_k^+ = v_k + \eta_k \frac{\|v_k\|}{\|\mathbf{g}_k\| + \varepsilon} \mathbf{g}_k, \tag{11}$$

where η_k controls the guidance strength. Since $\hat{z}_{0,k} = z_k - \sigma_k v_k$, this update changes the clean prediction by:

$$\hat{z}_{0,k}^+ - \hat{z}_{0,k} = -\sigma_k \eta_k \frac{\|v_k\|}{\|\mathbf{g}_k\| + \varepsilon} \mathbf{g}_k, \tag{12}$$

corresponding to a local descent step on the geometric loss.

Guidance is applied only within an intermediate noise window: early predictions provide unreliable geometry, while late corrections have limited influence and may introduce artifacts. Gradients pass through the frozen decoder and verifier to $\hat{z}_{0,k}$, but not through the denoising network. Discrete validity masks and geometric assignments are held fixed during each update. When combined, we rank and prune candidates using $R_{4\mathrm{D}}$, then apply guidance only to retained trajectories. Progressive search allocates computation across alternative futures, whereas geometric guidance locally refines each survivor. Final candidates are always selected using the original reward $R_{4\mathrm{D}}$.

4 Experiments

4.1 Implementation Details

We build our system on Wan2.2-I2V-14B (Wan et al., 2025) with all parameters frozen during inference. Videos are generated at 480×832 resolution with 81 frames using 40 FlowUniPC (Zhao et al., 2023) denoising steps and a noise-schedule shift of 5.0. We employ 4RC (Luo et al., 2026)



Figure 4: **Qualitative comparison with previous methods.** Each row shows one method, and each pair of columns shows an early and a late frame of the same scenario. While the baselines distort the arm and objects and let the background drift, GeoScale stays coherent.

as the geometry backbone, using 12 uniformly sampled frames at the first pruning checkpoint, 20 at later reward evaluations, and 8 for guidance; frames are resized to a long side of 518 pixels and center-cropped to a multiple of the 14-pixel patch size. For Best-of-N selection (N=8), we adopt progressive elimination with checkpoints at $\sigma\approx0.83$ and $\sigma\approx0.63$, eliminating the bottom 50% of candidates at each stage so that discarded trajectories are not fully denoised. Geometric guidance uses step size η=0.001. All model inference is conducted on 4× NVIDIA A800 80GB GPUs, with the DiT, VAE decoder, and 4RC distributed across devices to enable end-to-end differentiable guidance without model offloading bottlenecks. More details are provided in App. B.

4.2 Comparison with Baselines

We compare GeoScale with representative video world models for embodied control, including CogVideoX (Yang et al., 2025b), Wan2.2-I2V-14B (Wan et al., 2025), TesserAct (Zhen et al., 2025), and Geometry-Forcing (Wu et al., 2026). We exclude GEM-4D (Zhou et al., 2026) because its implementation is not publicly available. GeoScale operates purely at inference time on a frozen backbone and requires no generator retraining.

Qualitative Comparison. Figure 4 presents qualitative comparisons across three representative manipulation scenarios. Existing methods often produce incomplete or distorted geometry around manipulated objects, yielding unstable object shapes and spatial relationships during interaction. In contrast, GeoScale better preserves object boundaries, scene structure, and robot-object consistency, consistent with the depth, Chamfer, and correspondence improvements in Table 1.

Quantitative Comparison. We evaluate on two domains: (1) 50 real-world manipulation sequences from Droid (Khazatsky et al., 2024) covering grasping, placing, and tool-use tasks, and (2) 50 synthetic sequences rendered in RLBench (James et al., 2020) with exact ground-truth geometry. For each first-frame and action instruction pair, every method produces 20 independently seeded outputs; each GeoScale output is selected from an independent group of N=8 candidates. Per-video metrics are averaged over the 20 outputs and then over test cases, whereas FVD is computed once from the full generated and reference feature sets. Following prior work, we evaluate four dimensions: (a) *Video quality*, measured by FVD (Unterthiner et al., 2018), SSIM (Wang et al., 2004), and PSNR against ground truth; (b) *Depth accuracy*, measured by the standard AbsRel, δ_1, and δ_2 metrics (Eigen et al., 2014) using DA3 (Lin et al., 2025) as the evaluator; (c) *3D reconstruction*, measured by L1 Chamfer Distance (Fan et al., 2017) between predicted and ground-truth point clouds; and (d) *Correspondence*, measured by the TAP-Vid $\delta^{\text{vis}}_{\text{avg}}$ metric (Doersch et al., 2022), the percentage of visible points tracked within $\{1, 2, 4, 8, 16\}$ pixels, averaged over thresholds.

Table 1: **Comparison with state-of-the-art video world models.** ↑ and ↓ indicate that higher and lower values are better, respectively. **Red** and **Blue** denote the best and second-best results, respectively. D denotes dataset; **R** denotes real-world and **S** denotes synthetic.

D	Method	RGB			Depth			Points	Tracking
		FVD↓	SSIM (%)↑	PSNR↑	AbsRel↓	δ_1↑	δ_2↑	Chamfer↓	$\delta^{\text{vis}}_{\text{avg}}$ (%)↑
R	CogVideoX (Yang et al., 2025b)	77.50	58.11	14.02	0.463	0.579	0.768	0.238	61.72
	TesserAct (Zhen et al., 2025)	69.08	53.87	12.68	0.451	0.614	0.779	0.212	63.03
	Wan2.2-I2V-14B (Wan et al., 2025)	**52.43**	51.34	11.96	0.446	0.623	0.784	**0.205**	**63.68**
	Geometry-Forcing (Wu et al., 2026)	75.27	**58.96**	**14.17**	**0.443**	**0.626**	**0.793**	0.224	60.17
	Ours	**47.36**	**63.28**	**14.25**	**0.416**	**0.668**	**0.824**	**0.198**	**71.61**
S	CogVideoX (Yang et al., 2025b)	73.89	61.86	13.62	0.244	0.736	0.894	0.217	62.54
	TesserAct (Zhen et al., 2025)	68.37	62.53	14.04	0.219	0.769	0.899	0.224	65.31
	Wan2.2-I2V-14B (Wan et al., 2025)	62.16	65.71	**16.23**	**0.211**	0.795	**0.924**	**0.147**	**66.09**
	Geometry-Forcing (Wu et al., 2026)	**60.55**	**68.59**	16.11	0.215	**0.801**	0.913	0.165	64.27
	Ours	**56.74**	**71.34**	**18.77**	**0.204**	**0.825**	**0.927**	**0.101**	**72.71**

Table 2: **Task success rates (%) on Droid and RLBench.** Droid is human-rated; RLBench is simulator-executed. **Red** and **Blue** denote the best and second-best results.

Method	Droid			RLBench						
	ILIAD	IRIS	RAD	Open Drawer	Push Button	Turn Tap	Close Jar	Move Cube	Press Switch	Insert Peg
CogVideoX (Yang et al., 2025b)	18	45	22	32	29	17	19	25	21	11
TesserAct (Zhen et al., 2025)	21	53	35	44	37	24	25	33	34	19
Geometry-Forcing (Wu et al., 2026)	34	49	48	56	**45**	**61**	27	**51**	46	24
Wan2.2-I2V-14B (Wan et al., 2025)	**48**	**54**	**63**	**62**	43	57	**32**	38	**53**	**37**
Ours	**63**	**74**	**67**	**71**	**59**	**63**	**56**	**65**	**62**	**48**

As shown in Table 1, our method consistently improves geometric metrics (Depth, Chamfer, $\delta^{\text{vis}}_{\text{avg}}$) over the vanilla Wan2.2-I2V-14B backbone while preserving RGB fidelity. Compared with training-based methods (TesserAct, Geometry-Forcing), our approach achieves competitive or superior geometric quality *without any model fine-tuning*, demonstrating that inference-time geometric selection provides an effective training-free path to geometry-consistent video generation.

Embodied Action Planning Evaluation. To evaluate the downstream utility of improved geometric consistency, we assess generated rollouts on real-world Droid (Khazatsky et al., 2024) sequences and simulated RLBench (James et al., 2020) tasks (Table 2). For Droid, 20 participants assess perceived task completion, object and robot-arm deformation, and instruction following; Table 2 reports task completion, while the other judgments are complementary quality checks. For RLBench, following prior work (Zhou et al., 2026), we convert each generated video into a robot trajectory by estimating object and end-effector 6D poses with FoundationPose (Wen et al., 2024), recover a temporally consistent pose sequence, solve for actions, and execute them under the environment's success predicates. GeoScale improves both human-rated task completion on Droid and execution success on RLBench, suggesting that geometrically coherent rollouts are more actionable.

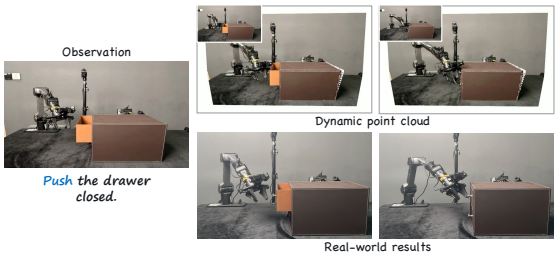

Figure 5: **Qualitative real-robot results.** Physical execution guided by a generated rollout.

Qualitative Real-Robot Experiment. We conduct five manipulation tasks on an AgileX robotic arm in unseen tabletop scenes. These tasks involve diverse objects and spatial arrangements. Given an initial RGB observation and a language instruction, GeoScale generates multiple rollouts and selects the one with the highest 4D consistency reward. FoundationPose (Wen et al., 2024) estimates the per-frame 6D poses of the manipulated object and end effector from the selected rollout. After temporal filtering, the recovered trajectory is converted into waypoints and joint-space commands via inverse kinematics for execution on the physical robot (Fig. 5); see App. D.3 for details.

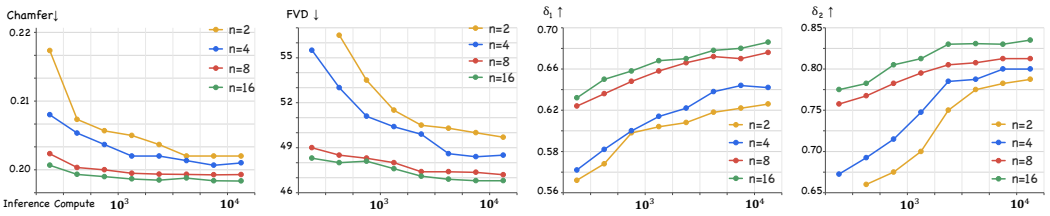

Figure 6: **Ablation study of candidate budget.** Chamfer distance and FVD (lower is better) and δ_1 and δ_2 (higher is better) versus inference compute for $N \in \{2, 4, 8, 16\}$. All metrics improve consistently with N, but the cost grows rapidly (Table 5), so we adopt N=8.

4.3 Ablation Studies.

Figure 7 and Table 3 present the qualitative and quantitative ablation results, respectively. More ablation studies are reported in App. F.

Effectiveness of Candidate Budget. Figure 6 studies the effect of the candidate budget N. Increasing N consistently improves all metrics by exploring more candidate trajectories, but also incurs higher inference cost, as shown in Table 5. We therefore use N=8 in our experiments to balance generation quality and computational efficiency.

Effectiveness of the 4D Consistency Reward (4DC-Reward). Removing 4DC-Reward degrades all appearance and geometry metrics, e.g., FVD rises from 47.36 to 50.36 and δ_2 drops from 0.824 to 0.797, demonstrating the value of the reward for selecting higher-quality rollouts.

Effectiveness of Geometric Optimization Guidance. Disabling guidance while keeping reward-based selection degrades all six metrics, confirming that guidance complements candidate ranking by refining the denoising trajectory.

Efficiency of Progressive Best-of-N (PBoN). We replace PBoN with conventional BoN, which fully denoises all N=8 candidates before selection. Conventional BoN and PBoN yield comparable quality, while PBoN halves the runtime from 4188 to 2094 seconds, demonstrating that progressive pruning removes substantial redundant computation.

Table 3: **Ablation study on the real-world Droid benchmark.** All variants follow the evaluation protocol of Table 1. "w/o PBoN" uses conventional BoN. **Red** and **Blue** denote the best and second-best results.

Method	RGB (Appearance)			Depth (Geometry)			Runtime (s) ↓
	FVD ↓	SSIM (%) ↑	PSNR ↑	AbsRel ↓	δ_1 ↑	δ_2 ↑	
w/o 4DC-Reward	50.36	57.92	13.01	0.427	0.635	0.797	**1973**
w/o Guidance	49.08	60.87	13.72	0.423	0.651	0.810	**1926**
w/o PBoN	**46.98**	**62.84**	**14.31**	**0.414**	**0.662**	**0.819**	4478
Ours	**47.36**	**63.28**	**14.25**	**0.416**	**0.668**	**0.824**	2094

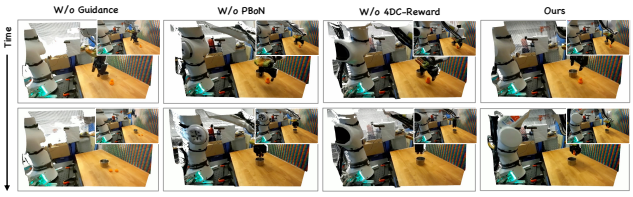

Figure 7: **Qualitative ablation study.** Comparison of variants without guidance or 4DC-Reward, and with PBoN replaced by conventional BoN.

5 Conclusion

We presented **GeoScale**, an inference-time scaling framework for narrowing the geometric actionability gap in video world models. Without retraining, GeoScale uses feed-forward 4D reconstruction as geometric feedback through a 4D consistency reward, progressive Best-of-N search, and geometric optimization guidance. Experiments on real-world Droid and simulated RLBench demonstrate consistent improvements in depth, 3D reconstruction, and point correspondence, alongside higher human-rated task completion and simulator execution success with recovered robot trajectories. Allocating test-time computation across candidate rollouts and informative denoising stages makes frozen video generators more geometrically coherent and useful for manipulation. GeoScale depends on the accuracy of the 4D reconstruction backbone and incurs additional inference cost; improving feedback robustness and compute efficiency remains an important future direction.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing, brainstorming, and limited code assistance. We did not use generative AI tools to generate experimental results, fabricate data, or make final scientific claims. All AI-assisted text, code, and suggestions were manually reviewed, revised, and verified by the authors. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies geometry-aware video generation for robot manipulation. While it may benefit robot planning and data generation, realistic synthetic videos could be misused to fabricate robot behavior or overstate system capabilities, and inaccurate generated rollouts could lead to unsafe actions if deployed without validation. We therefore recommend clearly identifying generated content and avoiding deployment in safety-critical settings without human oversight and real-world verification. Our human study evaluates generated rollouts and does not collect sensitive personal information. We will follow the licenses of all source datasets and models and document the intended use, limitations, and potential misuse risks in any public release.

REPRODUCIBILITY STATEMENT

We support reproducibility by providing a complete description of the GeoScale design, reward construction, candidate selection, guidance procedure, and evaluation protocol in the main paper and supplementary material. Section 3 specifies the 4D consistency reward, progressive Best-of-N search, and geometric optimization guidance, while Section 4 and the supplementary material describe the baselines, metrics, hyperparameters, computational resources, and evaluation settings. We will release the inference code, evaluation scripts, configurations, and example inputs and outputs needed to reproduce the reported results.

REFERENCES

Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 843–852, 2023.

Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *International Conference on Machine Learning*, 2024.

Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Easi3r: Estimating disentangled motion from dust3r without training. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9158–9168. IEEE, 2025.

Carl Doersch, Ankush Gupta, Larisa Markeeva, Adrià Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 605–613, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European Conference on Computer Vision*, 2024.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pp. 71–91, 2024.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, et al. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Yihang Luo, Shangchen Zhou, Yushi Lan, Xingang Pan, and Chen Change Loy. 4rc: 4d reconstruction via conditional querying anytime and anywhere. *arXiv preprint arXiv:2602.10094*, 2026.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025.

Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Motion-I2V: Consistent and controllable image-to-video generation with explicit motion modeling. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review. *arXiv preprint arXiv:2501.09685*, 2025.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Rapha"el Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric and challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10510–10522, 2025.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17868–17879, 2024.

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.

Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *European Conference on Computer Vision*, pp. 399–417, 2024.

Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21924–21935. IEEE, 2025a.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025b.

Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Y. Zou, and Stefano Ermon. Tfg: Unified training-free guidance for diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 22370–22417, 2024.

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. URL https://arxiv.org/abs/2602.15922.

Jianhao Yuan, Xiaofeng Zhang, Felix Friedrich, Nicolas Beltran-Velez, Melissa Hall, Reyhane Askari-Hemmat, Xiaochuang Han, Nicolas Ballas, Michal Drozdzal, and Adriana Romero-Soriano. Inference-time physics alignment of video generative models with latent world models. *arXiv preprint arXiv:2601.10553*, 2026.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. In *International Conference on Learning Representations*, 2025.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. Tesseract: Learning 4d embodied world models. *arXiv preprint arXiv:2504.20995*, 2025.

Hongyan Zhi, Peihao Chen, Siyuan Zhou, Yubo Dong, Quanxi Wu, Lei Han, and Mingkui Tan. 3dflowaction: Learning cross-embodiment manipulation from 3d flow world model. *arXiv preprint arXiv:2506.06199*, 2025.

Kaichen Zhou, Yuhan Wang, Grace Chen, Xinhai Chang, Gaspard Beaudouin, Fangneng Zhan, Paul Pu Liang, and Mengyu Wang. Page-4d: Disentangled pose and geometry estimation for 4d perception. *arXiv e-prints*, pp. arXiv–2510, 2025.

Kaichen Zhou, Yuzhen Chen, Fangneng Zhan, Hang Hua, Grace Chen, Xinhai Chang, Ao Qu, Yilun Du, Zhuang Liu, Paul Pu Liang, and Mengyu Wang. Gem-4d: Geometry-enhanced video world models for robot manipulation. *arXiv preprint arXiv:2605.22882*, 2026.

APPENDIX

A Additional Related Work **S1**

B Detailed Implementation **S2**

B.1 Progressive Search Schedule S2

B.2 Reward Validity Rules S2

B.3 Geometric Guidance Details S3

C Computational Resources and Inference Cost **S4**

D Evaluation Details **S4**

D.1 Datasets and Generation Protocol S4

D.2 Evaluation Metrics S4

D.3 Embodied Evaluation Protocols S5

E Additional Qualitative Results **S5**

F Additional Ablation Settings **S8**

G Limitations **S8**

H Potential Social Impact **S9**

Supplementary video. Please see the accompanying `demo.mp4` for additional video results.

A ADDITIONAL RELATED WORK

Video World Models for Embodied Control. Video world models provide a planning substrate by predicting how a scene may evolve under a language instruction or action command. UniPi (Du et al., 2023) generates future visual trajectories with video diffusion and recovers actions through inverse dynamics, while Uni-Sim (Yang et al., 2023) and Genie (Bruce et al., 2024) use generative world models to simulate interactive environments and support downstream policy learning. These approaches motivate using video prediction for embodied control, but they do not explicitly enforce the 3D geometry and temporal correspondences required for reliable action recovery. TesserAct (Zhen et al., 2025) jointly models RGB, depth, and surface normals, 3DFlowAction (Zhi et al., 2025) represents manipulation through 3D flow fields, and GEM-4D (Zhou et al., 2026) distills dense 4D correspondence representations into the video backbone. GeoScale differs from these training-based strategies by using a frozen video generator and applying geometry-aware selection or guidance only at inference time.

Feed-Forward 3D and 4D Geometry Models. Recent advances in feed-forward geometry estimation provide an efficient alternative to scene-specific optimization for recovering 3D structure from visual observations. DUSt3R (Wang et al., 2024) formulates reconstruction as direct prediction of pixel-aligned pointmaps from image pairs, and MASt3R (Leroy et al., 2024) improves geometric matching through learned local features. Fast3R (Yang et al., 2025a) scales this paradigm to multi-view reconstruction via global feature fusion. Scaling feed-forward reconstruction to dynamic scenes is particularly challenging because camera ego-motion and object motion must be modeled jointly. MonST3R (Zhang et al., 2025) estimates temporally consistent geometry in the presence of motion, while CUT3R (Wang et al., 2025) introduces a persistent-state formulation for dynamic 3D perception. Easi3R (Chen et al., 2025) explores training-free motion disentanglement on top of

DUSt3R, and PAGE-4D (Zhou et al., 2025) enhances dynamic reconstruction by explicitly separating static and dynamic components. Unlike per-scene optimization, these models recover geometry in a feed-forward manner, making them well-suited as geometric priors for video generation and world modeling.

Inference-Time Alignment for Diffusion Models. Inference-time alignment can be implemented either as search over multiple denoising trajectories or as gradient guidance during sampling. Derivative-free methods explore candidate generations and resample them according to a reward signal, including Best-of-N selection (Ma et al., 2025), sequential Monte Carlo sampling (Singhal et al., 2025), and local reward-aware resampling such as SVDD (Li et al., 2024). Gradient-based methods directly steer the diffusion score using differentiable objectives (Bansal et al., 2023; Ye et al., 2024). WMReward (Yuan et al., 2026) combines these ideas for video generation by using the predictive surprise of V-JEPA 2 (Assran et al., 2025) as a dynamics reward. GeoScale follows the same inference-time alignment philosophy but replaces latent predictive surprise with explicit 4D scene consistency, including static reprojection, dynamic trajectory smoothness, and camera-motion regularity.

B Detailed Implementation

This section specifies the pruning schedule, validity rules, and guidance surrogates that Sec. 3 defers to the appendix. All numerical settings are collected in Table 4; the backbone, resolution, and hardware configuration are given in Sec. 4.1. Throughout, med denotes the median.

B.1 Progressive Search Schedule

Candidates share the text and image conditioning and differ only in their initial noise. Each candidate keeps an independent FlowUniPC (Zhao et al., 2023) solver instance so that multi-step solver history is preserved across pruning. At checkpoint k_j, the clean prediction in Eq. (6) is formed from the pre-step latent and the classifier-free-guidance-merged velocity (Ho & Salimans, 2022). The survivor count in Eq. (7) is

$$n_j = \max\big(n_{\min},\ |\mathcal{A}_{j-1}| - \lfloor r\,|\mathcal{A}_{j-1}|\rfloor\big), \tag{13}$$

where r is the elimination ratio and $n_{\min}$ the minimum population. One additional candidate is retained when the reward gap across the pruning boundary is below the tie threshold ϵ. The same reward is used at every checkpoint; only the number of verifier frames K differs.

B.2 Reward Validity Rules

Verifier Inputs and Partition. The verifier receives K uniformly sampled frames, always including the first frame, resized as listed in Table 4. In addition to the quantities in Sec. 3.1, it returns per-pixel confidences for points and tracks and intrinsics $\mathbf{K}_t$. The world frame coincides with the first camera, and the scene scale s is the median first-frame depth. A point or track is valid when its confidence exceeds a per-video quantile. A pixel belongs to Ω_{d} when the maximum displacement of its track from the first frame exceeds a threshold relative to s; using the maximum rather than the final displacement captures "move-then-return" motion.

Aggregation. The robust aggregation $\langle\cdot\rangle$ in Eqs. (1)–(3) is the median over valid observations. For E_{s}, static points of frame i are projected into camera j, and $d_j(\pi_j(\mathbf{x}_i(u)))$ is obtained by bilinear sampling of frame j's reconstructed depth map. The median is taken over pixels that are valid in both frames, project inside the image with positive depth, and pass an occlusion test with the listed margin, and is then averaged over the frame pairs $(i, i{+}\delta)$ given by the listed strides δ. E_{d} takes the median over dynamic pixels and time, and E_{c}, E_{r} over time.

Validity Processing and Fallbacks. A frame pair is skipped for E_{s} if fewer than 100 static pixels are valid in both frames, fewer than 50 project inside the image, or fewer than 20 survive the occlusion test. $\widetilde{R}_{\mathrm{s}}$ and $\widetilde{R}_{\mathrm{d}}$ down-weight R_{s} and R_{d} by the fraction of static (resp. dynamic) pixels that remain valid (*under-observed*); $\widetilde{R}_{\mathrm{d}}$ also penalizes a 95th-percentile track speed far above the median (*teleporting*); and $\widetilde{R}_{\mathrm{g}}$ gates R_{g} on the median dynamic displacement (*motionless*) and on the fraction of per-frame dynamic displacements exceeding $0.5\,s$ (*teleporting*). A component with no remaining

observations returns the neutral score 0.5. The first-frame gate is $G_{\mathrm{a}} = \mathrm{sigmoid}\big((\nu_{\mathrm{a}} - c_{\mathrm{a}})/w_{\mathrm{a}}\big)$, with ν_{a} the fraction of first-frame static pixels with finite positive depth and $(c_{\mathrm{a}}, w_{\mathrm{a}})$ the listed center and width.

B.3 Geometric Guidance Details

Surrogate Loss. The camera term is not used for guidance. Let Ω_{s}^{i} denote static pixels that are valid in frames i and $i{+}1$ and whose reprojection $\pi_{i+1}(\mathbf{x}_i(u))$ lies inside frame $i{+}1$ with positive depth and passes the occlusion test of Sec. B.2, let Ω_{d}^{*} denote dynamic pixels whose tracks are valid in every guidance frame (at most 500 are used), and let $d_{\mathrm{ref}}(u)$ be a fixed reference depth of the conditioning image I_0. With ρ_δ the Huber penalty of width δ, the terms of Eq. (9) are

$$\begin{aligned} \mathcal{L}_{\mathrm{stat}} &= \operatorname*{mean}_{i<K-1} \operatorname*{mean}_{u\in\Omega_{\mathrm{s}}^{i}} \Big|\log d_{i\to i+1}(u) - \log d_{i+1}\big(\pi_{i+1}(\mathbf{x}_i(u))\big)\Big|, \\ \mathcal{L}_{\mathrm{dyn}} &= \operatorname*{mean}_{t,\,u\in\Omega_{\mathrm{d}}^{*}} \|\Delta^2 \mathbf{y}_t(u)\|/s, \qquad \mathcal{L}_{\mathrm{anchor}} = \operatorname*{mean}_{u\in\Omega_{\mathrm{s}}} \rho_\delta\big(\log d_0(u) - \log d_{\mathrm{ref}}(u) - b\big), \end{aligned} \tag{14}$$

where K is the number of guidance frames and $b = \mathrm{med}_{u\in\Omega_{\mathrm{s}}}\big(\log d_0(u) - \log d_{\mathrm{ref}}(u)\big)$ is detached and absorbs the global scale ambiguity between the two reconstructions. $\mathcal{L}_{\mathrm{stat}}$ is the mean-aggregated counterpart of Eq. (1) over all consecutive guidance-frame pairs, with $d_{i+1}(\cdot)$ bilinearly sampled at the reprojected location so that gradients flow through both the projected point and the sampled depth map. Because $\mathcal{L}_{\mathrm{stat}}$ is purely relative, the reconstruction of all frames can drift jointly without penalty; $\mathcal{L}_{\mathrm{anchor}}$ supplies an absolute reference by tying the first-frame reconstruction to the geometry of the conditioning image, which the image-conditioned generator is required to preserve, and the Huber penalty limits the influence of local errors in d_{ref}. All terms use means rather than medians for dense gradients; a term with too few valid pixels is set to zero.

Step Size, Gradient Path, and Schedule. The guidance strength in Eq. (11) is $\eta_k = \eta(1 - \sigma_k^2)$, and the norms are global ℓ_2 norms over the full latent. $\hat{z}_{0,k}$ is detached from the denoiser and treated as the optimization variable; $\mathbf{g}_k$ flows through the verifier and the VAE decoder, which is evaluated with gradient checkpointing over temporal chunks of three latent frames with the causal convolution cache detached at chunk boundaries. Valid masks (including the reprojection and occlusion masks of Ω_{s}^{i}), the static/dynamic partition, the scene scale, d_{ref}, and b are computed without gradient and held fixed. Guidance is applied at steps with $\sigma_{\min} < \sigma_k < \sigma_{\max}$ and $k \bmod f = 0$; in the combined setting it starts at the first pruning checkpoint, immediately after candidates are scored and pruned.

Table 4: **Principal settings of GeoScale.** Values are shared across datasets unless stated otherwise.

Group	Hyperparameter	Value
Verifier	Input long side / patch size	518 / 14
	Frames K (first checkpoint / later / guidance)	12 / 20 / 8
	Confidence validity quantile	0.20
	Dynamic threshold (relative to s)	0.01
	Neutral fallback score	0.5
Reward	$(\lambda_{\mathrm{s}}, \lambda_{\mathrm{d}}, \lambda_{\mathrm{g}})$	$(0.5, 0.3, 0.2)$
	τ_{s} (static), occlusion margin, frame strides	0.05, 1.05, $\{1, 3, 5\}$
	τ_{d} (dynamic)	0.02
	τ_{c} (camera), τ_{r} (rotation)	0.02, 0.05
	First-frame gate center / width	0.8 / 0.05
Search	Candidates N, denoising steps, schedule shift	8, 40, 5.0
	Checkpoints σ	0.83, 0.63
	Elimination ratio r, min. survivors $n_{\min}$	0.5, 2
	Tie threshold ϵ	0.02
Guidance	Step size η; norm regularizer ε	0.001; 10^{-8}
	Loss weights $(\omega_{\mathrm{s}}, \omega_{\mathrm{d}}, \omega_{\mathrm{a}})$	$(1, 0.5, 0.3)$
	Static pairs in $\mathcal{L}_{\mathrm{stat}}$; Huber width δ	all $(i, i{+}1)$; 0.1
	Max. dynamic tracks in $\mathcal{L}_{\mathrm{dyn}}$	500
	Frequency f; window $(\sigma_{\min}, \sigma_{\max})$ (alone / combined)	5; $(0.08, 0.90)$ / 3; $(0.08, 0.83)$

C Computational Resources and Inference Cost

Hardware and Runtime. GeoScale requires no training or fine-tuning; all additional computation is incurred at inference time. Table 5 reports wall-clock generation time on the hardware specified in Sec. 4.1. Runtime includes candidate denoising, intermediate decoding, reward evaluation, pruning, guidance, and final selection; it excludes model loading, dataset I/O, evaluation metrics, and optional video-to-action conversion.

Table 5: **Inference-time computational cost at different candidate budgets.** Wall-clock time is measured per generated video on $4\times$ NVIDIA A800 80GB GPUs.

Candidate budget	Runtime (s)	Runtime (min)
N=2	∼1264	∼21.1
N=4	∼1432	∼23.9
N=8	∼2094	∼34.9
N=16	∼3698	∼61.6

Denoising Cost. Under the default setting, progressive search interrupts discarded trajectories at the σ checkpoints, so its DiT cost is lower than independently denoising all N candidates to completion. Search-time decoding and reward evaluations follow the active population at each checkpoint and at the final selection, and can increase when a tie keeps an extra survivor. Guidance adds separate sparse decoding and reconstruction passes. We report measured wall-clock time rather than a fixed step count (Tables 5 and 3).

D Evaluation Details

D.1 Datasets and Generation Protocol

Evaluation Datasets. We evaluate on two complementary domains. (1) *Real-world*: 50 manipulation sequences from the Droid dataset (Khazatsky et al., 2024), covering three task categories: ILIAD, which involves moving an object to a new position and orientation in kitchen- or sink-like countertop scenes; IRIS, which involves moving a lid on or off a container in a cleaner tabletop setting; and RAD, which covers additional real-world manipulation cases. Each sequence provides a first-frame observation and a language action instruction. (2) *Synthetic*: 50 sequences from RLBench (James et al., 2020) with access to exact ground-truth depth, point clouds, and camera parameters, enabling pixel-accurate geometric evaluation. All videos are generated at 480×832 resolution with 81 frames.

Generation Protocol. For each test case (first frame + action instruction), every method produces 20 independently seeded outputs. Each GeoScale output is obtained from an independent group of N=8 candidates using the progressive selection and guidance procedure. We average per-video metrics over the 20 outputs for each case and then over test cases. Because FVD is a distribution-level metric, we instead compute it once from the full sets of generated and reference clip features.

D.2 Evaluation Metrics

Depth Accuracy Metrics. We extract per-frame depth with DA3 (Lin et al., 2025). Synthetic sequences use exact ground-truth depth, whereas real-world sequences use DA3 estimates from the reference videos as pseudo ground truth. Depth is scale-aligned per frame using median matching before computing AbsRel, δ_1, and δ_2.

3D Reconstruction Metric. *Chamfer Distance*: We back-project each frame's depth map into 3D using known (synthetic) or estimated (real) camera intrinsics, producing per-frame point clouds. The L1 Chamfer Distance (Fan et al., 2017) is computed between predicted and ground-truth point clouds after ICP alignment (Besl & McKay, 1992), then averaged over all frames.

Correspondence Metric. δ^{vis}_{avg}: Following TAP-Vid (Doersch et al., 2022), we uniformly sample 256 visible query points from the first frame, which is shared by the generated and paired ground-

truth videos. We initialize CoTracker (Karaev et al., 2024) with the same queries in both videos and independently track them through the temporally aligned 81-frame sequences. At each frame in which a reference point is visible, we compare the generated-video track with its corresponding ground-truth-video track; a missing or invalid generated track is counted as incorrect. For each threshold $\tau \in \{1, 2, 4, 8, 16\}$ pixels, we compute the fraction of tracks whose Euclidean localization error is below τ. The final metric averages over thresholds, frames, and sequences and is reported as a percentage. Semantic or motion deviations from the paired reference are therefore counted as tracking errors rather than discarded. Higher values indicate better temporal correspondence with the reference rollout.

D.3 Embodied Evaluation Protocols

Droid Human Evaluation and Simulator Task Success. For embodied evaluation, we follow the Adaptive Inverse Dynamic System (AIDS) strategy of GEM-4D (Zhou et al., 2026) to convert generated video rollouts into executable 6-DoF actions. Given a generated rollout, we first estimate per-frame depth and camera intrinsics with the same geometry backbone used for evaluation, and obtain task-relevant object and end-effector masks from the first frame and the language instruction. We then apply FoundationPose (Wen et al., 2024) to estimate the 6D pose of the manipulated object and the end effector in the generated frames. The resulting pose sequence is filtered by temporal consistency: low-confidence or discontinuous pose estimates are replaced by depth-centroid translation estimates and rotation interpolation from neighboring valid frames. This produces a recovered 6-DoF end-effector trajectory $\{(\mathbf{R}_{\text{ee}}^t, \mathbf{T}_{\text{ee}}^t)\}_{t=0}^{T-1}$ aligned with the T-frame predicted video rollout.

The recovered pose trajectory is then converted into executable robot actions. We select task waypoints from the end-effector trajectory, insert grasp or release waypoints when required by the task, smooth the trajectory with interpolation, and solve inverse kinematics to obtain joint-space commands. On RLBench, we replay the extracted action sequence and measure task completion using the environment's built-in success criteria.

For the real-world Droid evaluation, we recruit 20 participants and conduct a blinded, randomized study. For each of the 50 test cases and five evaluated methods, we uniformly sample one rollout from the 20 generations using a fixed study seed, producing 250 anonymized clips. Each participant evaluates 50 clips, with one method output per test case presented in an independently randomized order, so no participant directly compares multiple methods on the same case. The assignment is balanced across participants such that every combination of method and case receives four independent ratings ($20 \times 50 = 250 \times 4 = 1000$ total ratings). For each clip, participants answer three binary questions: (1) *task success*, whether the interaction completes the requested manipulation; (2) *object and robot-arm deformation*, whether the object and manipulator preserve their structure without visible distortion; and (3) *instruction following*, whether the predicted motion agrees with the language command. Table 2 reports the first judgment, averaged within the ILIAD, IRIS, and RAD task categories across assigned clips and raters; the other two judgments serve as complementary quality checks. The human-evaluated Droid results are reported separately from RLBench execution success.

Qualitative Real-Robot Experiment. We conduct five manipulation tasks on an AgileX robotic arm in real-world tabletop scenes. As shown in Fig. 9, a wrist-view camera observes local interactions while a fixed front-view camera captures the global workspace. The tasks use varied household objects, including blocks, cups, bowls, plates, a towel, and an articulated drawer, with backgrounds and object arrangements unseen by the video generator.

E Additional Qualitative Results

Figures 8, 10, 11, and 12 present additional generated rollouts and their corresponding dynamic point-cloud reconstructions across diverse manipulation settings. Each task block shows six temporally ordered RGB frames in the upper row and the reconstructed dynamic geometry in the lower row, with the language instruction shown beneath the block.

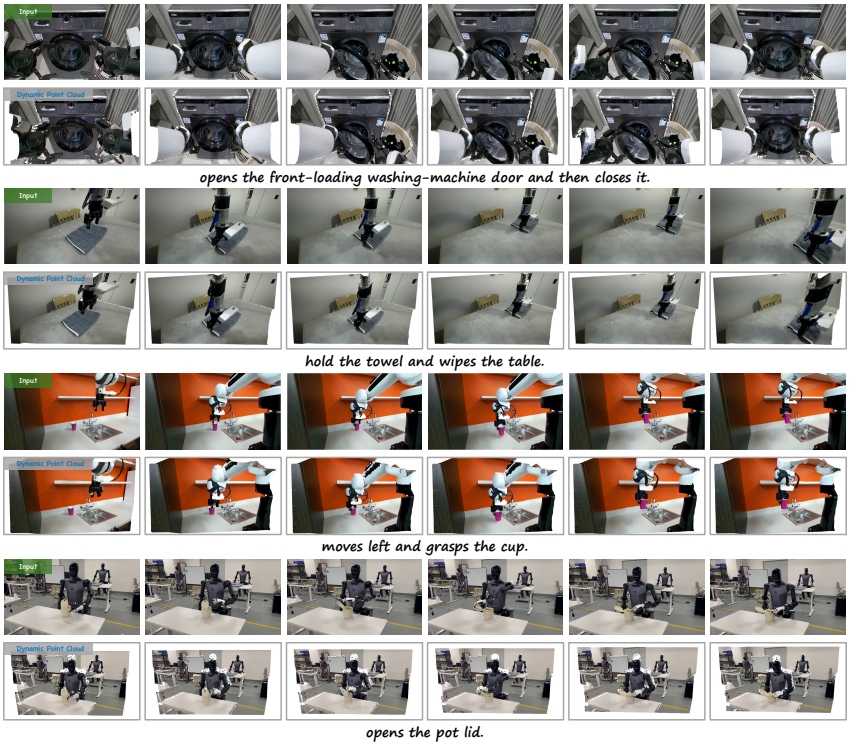

Figure 8: **Additional qualitative results (I).** Examples include opening and closing a washing-machine door, wiping a table with a towel, grasping a cup, and opening a pot lid.

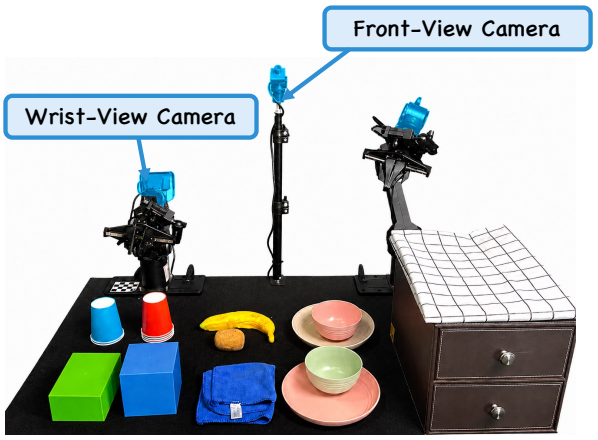

Figure 9: **Real-robot evaluation environment.** The tabletop setup includes wrist-view and fixed front-view cameras together with diverse household objects used in the manipulation tasks.

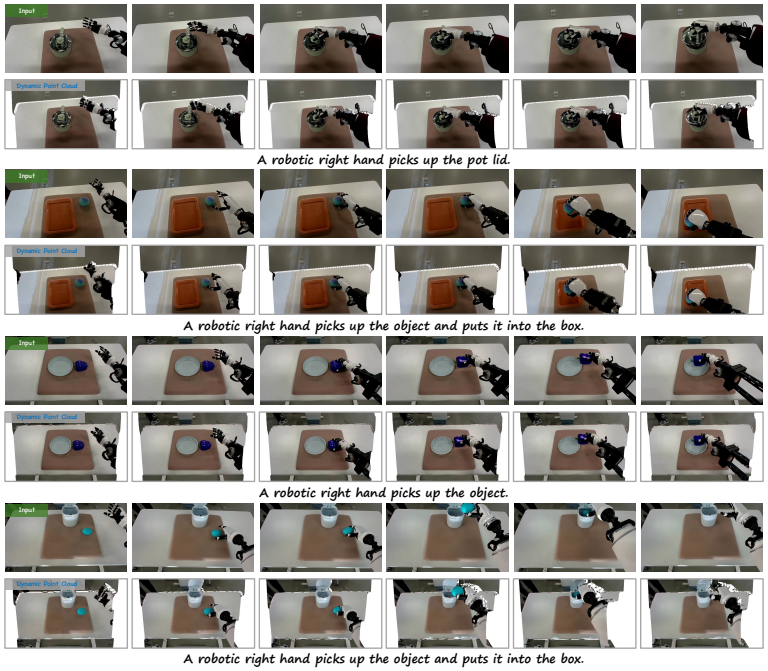

Figure 10: **Additional qualitative results (II).** Tabletop manipulation examples involving object pickup, lid removal, and object placement into containers.



Figure 11: **Additional qualitative results (III).** Examples of cup grasping, object insertion, spoon pickup, and pot-lid manipulation in varied real-world scenes.

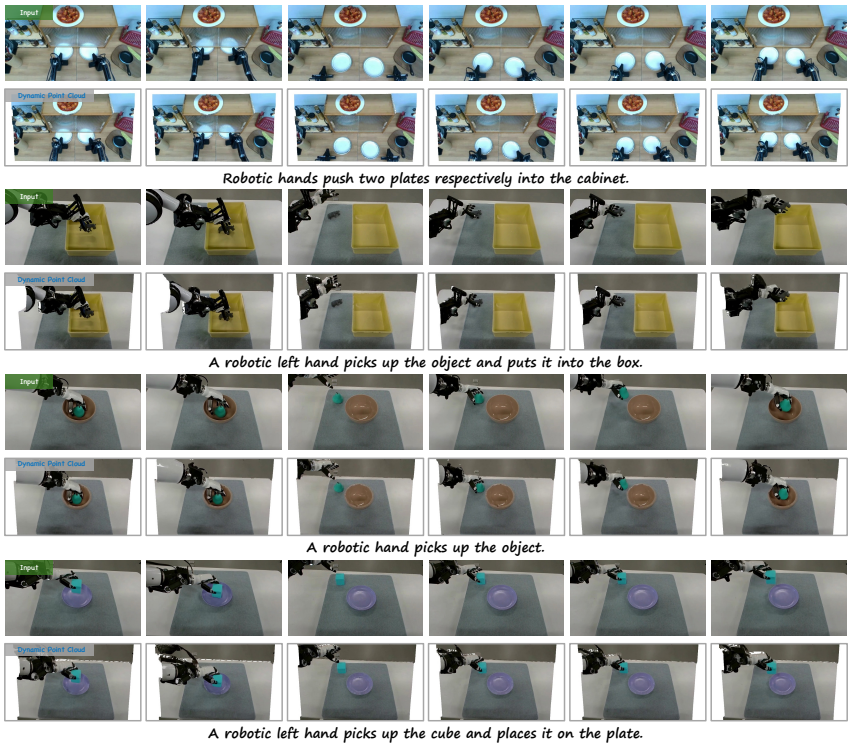

Figure 12: **Additional qualitative results (IV).** Examples of bimanual plate placement and single-arm object transfer to containers and plates.

F Additional Ablation Settings

Effectiveness of various Video Backbones. Figure 13 evaluates GeoScale with HY-Video (Kong et al., 2024), Cosmos (Agarwal et al., 2026), and Wan2.2-I2V-14B (Wan et al., 2025). The consistent improvements across all three backbones demonstrate the effectiveness and generality of our inference-time geometric guidance.

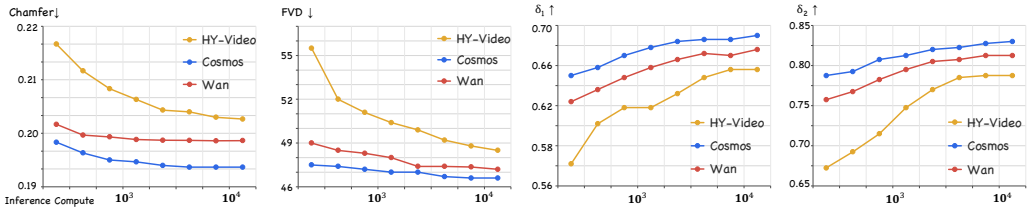

Figure 13: **Ablation study of various video backbones.** We report Chamfer distance and FVD (lower is better), together with δ_1 and δ_2 (higher is better).

G Limitations

Dependence on Geometric Feedback. GeoScale relies on a pretrained 4D reconstruction model to score and guide generated rollouts. Reconstruction errors under severe occlusion, fast motion, weak texture, or reflective and transparent objects may therefore produce inaccurate rewards or gradients. Although the combined consistency terms improve robustness, GeoScale cannot guarantee physically correct geometry when the underlying 4D estimate is unreliable.

Inference Cost. GeoScale trades additional test-time computation for improved geometric consistency. Progressive Best-of-N substantially reduces redundant denoising, but repeated candidate

generation, VAE decoding, and 4D reconstruction remain expensive and currently require multiple high-memory GPUs. Future work could mitigate this cost through more efficient attention implementations such as FlashAttention and faster distilled causal video-generation backbones.

Semantic Mismatch between Images and Instructions. GeoScale improves the geometry of candidate rollouts but cannot reliably resolve an infeasible semantic condition between the initial observation and the language instruction. As shown in Figure 14, when the scene contains only a cup but the instruction asks the robot to pick up an apple, the video model may introduce the absent target object in the generated rollout. Handling such cases requires explicit instruction grounding and feasibility detection beyond the current 4D geometric feedback.



Figure 14: **Limitation.** The observation contains a cup but no apple, while the instruction refers to an absent target object. The video model consequently introduces an apple in the generated rollout.

H Potential Social Impact

GeoScale may improve the usefulness of video world models for robot planning, simulation, and synthetic data generation. More geometrically coherent rollouts could reduce the number of costly physical trials required during system development and support safer evaluation in simulation before real-world deployment.

At the same time, realistic generated videos may be used to fabricate robot behavior or overstate a system's capabilities. Generated rollouts and recovered actions can also contain subtle geometric errors that are difficult to detect visually and could lead to unsafe behavior if executed without verification. Biases or failure modes inherited from the video generator, geometry backbone, and evaluation data may further cause uneven performance across objects, environments, and tasks. We therefore recommend clearly labeling synthetic content, independently validating trajectories before execution, and retaining human oversight and standard robot-safety mechanisms in real-world applications. Any public release should document known limitations, intended uses, dataset and model licenses, and prohibited misuse.