CTRLWAM: CONTROLLABLE WORLD ACTION MODELS WITH ALIGNED INTENT AND FORESIGHT

Anonymous authors
Paper under double-blind review

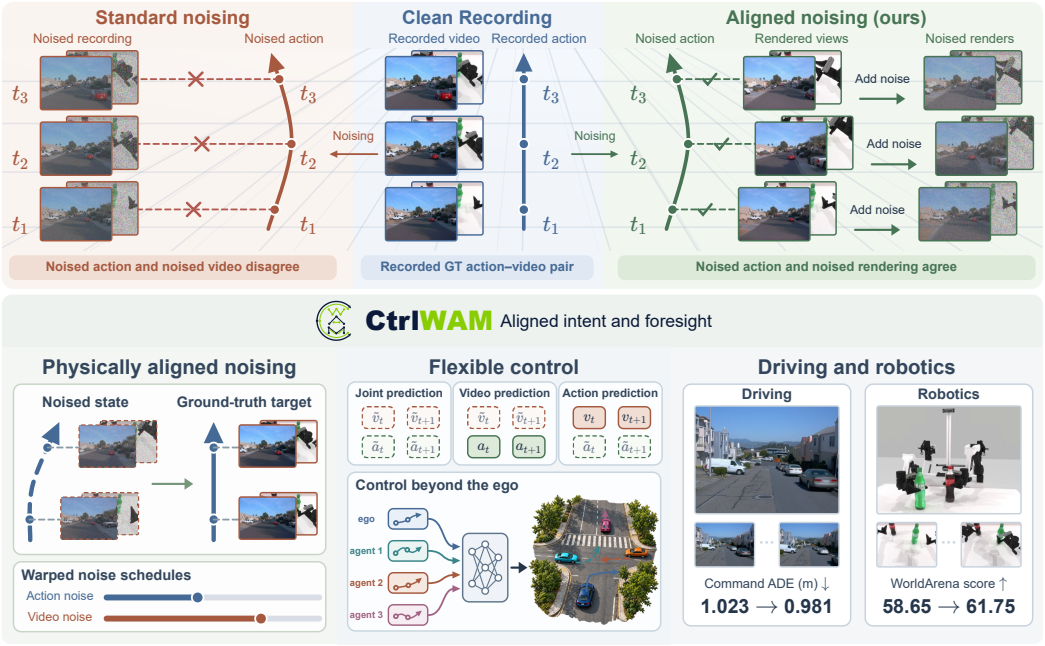


Figure 1: Overview of CTRLWAM. Standard noising pairs perturbed actions with video tied to the recorded future. Aligned noising instead renders the consequences of the perturbed trajectories before adding visual noise, while retaining the recorded action–video pair as the training target. Combined with warped video–action noise schedules, the model supports joint prediction, action-conditioned video generation, video-conditioned action prediction, and control of surrounding agents, with applications to driving and robotic manipulation.

ABSTRACT

World action models (WAMs) jointly predict actions (intent) and visual future (foresight). Standard training adds noise to recorded actions and video simultaneously, but such training paradigms introduce a mismatch: perturbed actions imply counterfactual future visual, while the noised video remains tied to the GT recording. In low-noise regime, the scene geometry and even the dynamic behavior remain clearly visible from the noisy future frames despite the added noise. We present CTRLWAM, which executes perturbed actions in a simulator and pairs them with their noised visual consequences for joint WAM learning. To accommodate the different denoising requirements of video and actions, we introduce warped video–action noise schedules that aim to keep visual layout responsive as action predictions evolve. We further extend the action interface from ego-only control to a variable number of agent streams, allowing a unified model to represent predicted or commanded futures for multiple agents. Driving experiments show more accurate action forecasts, closer agreement between generated video and actions, and better following of supplied commands; robotics experiments show stronger motion fidelity and controllability. Matched controls support the benefit of off-path renders for command following and manipulation fidelity. Together, these findings contribute to a more controllable world action model.

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.

1 INTRODUCTION

World action models (WAMs) jointly predict an embodied agent's future actions (*intent*) and their visual consequences (*foresight*) (Zhu et al., 2025; Ye et al., 2026). The imagined video should therefore remain consistent with the consequences of predicted actions. In unified models such as Cosmos 3 (Agarwal et al., 2026), which perform both joint prediction and forward dynamics (action-conditioned video generation), this intent–foresight alignment is essential in both modes.

However, current training procedures do not explicitly preserve this alignment during denoising. Methods such as DreamZero (Ye et al., 2026) independently corrupt the recorded actions and video with Gaussian noise (Ho et al., 2020). Perturbing the actions in this way does not change the motion depicted in the video: the corrupted video remains tied to the recorded future, even when the perturbed actions would produce a different trajectory. The model is thus trained to recover the paired clean targets, but it receives no explicit supervision on the physical correspondence between its intermediate action and video states.

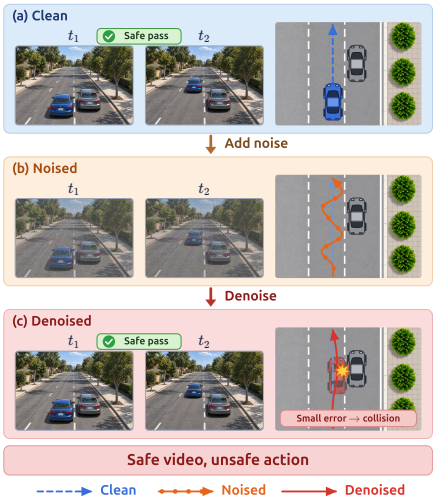


Figure 2: **Safe video, unsafe action.** This schematic illustrates intent–foresight misalignment: after noising and denoising, the video depicts a safe pass, while a small deviation in the predicted trajectory leads to a collision.

This gap matters because useful visual structure can emerge early in denoising. DIDO (Lyu et al., 2026) observes that static scene structure forms before gripper–object dynamics are fully resolved. In its four-step video-model ablation, conditioning the action expert on first-step representations already achieves 97.7% success on LIBERO, compared with 98.4% after all four steps. These findings suggest that intermediate visual predictions already provide substantial information for action generation, which motivates closer attention to their physical consistency with the evolving action estimates. Without explicit alignment, an intermediate action estimate could veer toward an obstacle while the imagined video still depicts a safe pass (Figure 2).

We introduce CTRLWAM for physically aligned denoising in world action models. Its central idea, *aligned noising*, pairs each action perturbation with its visual consequence to construct physically corresponding training inputs (Figure 1). Specifically, a simulator executes the perturbed actions and renders the resulting observations, and we use RoboTwin (Chen et al., 2025) for manipulation and AlpaSim (NVIDIA et al., 2025) for driving. We then add noise to the rendered video and train the model to recover the recorded actions and video at the corresponding timestamps. These paired inputs provide a physically grounded training signal for flow matching (Lipman et al., 2022).

Learning from these pairs also requires choosing how much visual evidence to expose at each stage of denoising. Video and actions can be assigned separate noise levels (Chen et al., 2024) and can therefore follow different noise schedules (Guo et al., 2026). Our preliminary inspection suggests that the video layout can stabilize and enter detail refinement while the action estimates are still changing (Figure 3). This observation motivates *warped video–action noise schedules*. By assigning video a higher noise level than actions, we aim to keep the video layout responsive to evolving action predictions, and then let it settle into detail refinement as the actions approach convergence.

In interactive scenes, the future also depends on the behavior of other agents. A driving scenario, for example, may leverage a neighboring vehicle's maneuver in addition to the history ego trajectory for the action prediction. We therefore introduce *structured multi-agent conditioning*, which extends the ego action stream to a variable number of additional agents. Conditioning masks specify which futures are predicted and which are supplied as commands. This interface supports both joint multi-agent prediction and action-conditioned generation, and it reduces exactly to the ego-only model when no other agents are present.

We evaluate CTRLWAM on autonomous driving (NVIDIA et al., 2025) and bimanual manipulation (Chen et al., 2025). In driving, we assess action forecasting, video–action consistency, and adherence to supplied commands. In manipulation, we assess motion fidelity and video quality.

In summary, our main contributions are as follows:

- **Physically aligned noising**, which pairs perturbed actions with their simulator-rendered visual consequences and trains the model to recover the recorded actions and video, providing physically corresponding inputs for video–action denoising.
- **Warped video–action noise schedules**, which assign video a higher noise level than actions, aiming to keep the video layout adaptable while actions evolve and to shift toward detail refinement as they stabilize.
- **Structured multi-agent conditioning**, which extends the ego action stream with streams for additional agents, so that each agent's future can be either predicted to improve the ego-prediction or supplied as a command to enhance the controllability.

2 RELATED WORK

Unified World Action Models Joint video–action models learn action prediction and visual dynamics within a shared generative model. The Unified World Models (UWM) framework couples video and action diffusion within one transformer (Zhu et al., 2025). The Unified Video Action (UVA) model learns a joint latent representation with separate video and action decoders (Li et al., 2025). Both support policy learning and forward or inverse dynamics through flexible conditioning. VideoVLA and DreamZero adapt pretrained video generators for joint prediction of actions and visual futures in robotic control (Shen et al., 2026; Ye et al., 2026; Yuan et al., 2026).

Controllable World Models Controllable world models condition visual futures on commands, actions, or scene structure. In robotics, action-conditioned video prediction supports model-predictive planning (Finn & Levine, 2017), while Genie learns latent actions for interactive generation from unlabeled videos (Bruce et al., 2024). GAIA-1 uses video, text, and action inputs to steer driving scenarios (Hu et al., 2023), while DriveDreamer learns structured traffic constraints for controllable generation (Wang et al., 2024). Vista supports controls ranging from high-level commands and goal points to trajectories, steering angles, and speeds (Gao et al., 2024).

Modality-specific noising Conditional flow matching learns velocity fields along prescribed probability paths (Lipman et al., 2022). Diffusion-model design studies show how noise sampling and loss weighting allocate learning effort across denoising levels (Karras et al., 2022). Diffusion Forcing uses independent per-token noise levels for sequence modeling (Chen et al., 2024), while UWM uses separate video and action timesteps to support flexible conditioning (Zhu et al., 2025).

3 CTRLWAM

3.1 PHYSICALLY ALIGNED NOISING

A recorded action–video pair specifies how the two modalities agree at the clean endpoint. It does not specify how they should deviate together from that endpoint as noise is added. For example, noising an action trajectory produces a different ego camera path, implying a different visual future, yet adding Gaussian noise to the recorded video does not reflect this change (Figure 2). Aligned noising instead constructs an action perturbation and its visual outcome as a paired training input (Figure 1). This gives the model physically related evidence in both modalities as it learns to recover their shared recorded future.

For each recording, we sample $t \in [0, 1)$, which sets the action noise level $\sigma_a = 1 - t$. We then construct the perturbed command $\mathbf{a}_t^{\text{cmd}} = t\mathbf{a}^\star + (1-t)\epsilon_a$ with $\epsilon_a \sim \mathcal{N}(0, I)$. A simulator executes this command and renders its visual consequences. We use RoboTwin (Chen et al., 2025) for manipulation and AlpaSim (NVIDIA et al., 2025) for driving. We store the executed action trajectory $\mathbf{a}_t$ and the latents $\mathbf{r}$ of the rendered video as one training pair. The recorded history

provides the conditioning prefix, and the recorded future over the same timestamps as the rollout supplies the clean supervision targets.

The executed trajectory $\mathbf{a}_t$ serves as the perturbed action input to the denoiser. The video, in contrast, is corrupted with freshly sampled noise: $\mathbf{v}_{\sigma_v} = \sigma_v \epsilon_v + (1 - \sigma_v)\mathbf{r}$ with $\epsilon_v \sim \mathcal{N}(0, I)$. The video noise level σ_v is assigned separately, as described in Section 3.2. The resulting input therefore pairs a perturbed action with a corrupted view of *its* consequences, rather than with a noised view of the original, unperturbed future. We refer to this pairing as *physical correspondence*.

Keeping the recording as the target turns the simulated deviation into an input from which the model learns to recover a reference future. Using the perturbed rollout as the target would instead teach the model to reconstruct that alternative future, rather than the common future. Because the input no longer lies on the standard interpolation path, the denoising target accounts for the full difference between the actual training input and the recording. This difference includes both the change introduced by the simulated motion and the added video noise. The model is thus supervised to correct both the behavioral and the visual discrepancy in each paired example.

3.2 Warped video–action noise schedules

During joint generation, the video can commit to a layout in early denoising steps while the action estimates are still changing. Our preliminary inspection (Figure 3) suggests that later video denoising steps then mainly refine appearance (left). It can be observed that the implied action from the video stabilizes quickly although the video is still on a high-level noise. Meanwhile, the explicitly predicted actions are still changing. As a result, the motion implied by the early video may differ from that of the evolving action. A shared noise schedule therefore allows the video to commit to a trajectory before the action prediction settles, and there is a mismatch between the video-implied action and the predicted action.

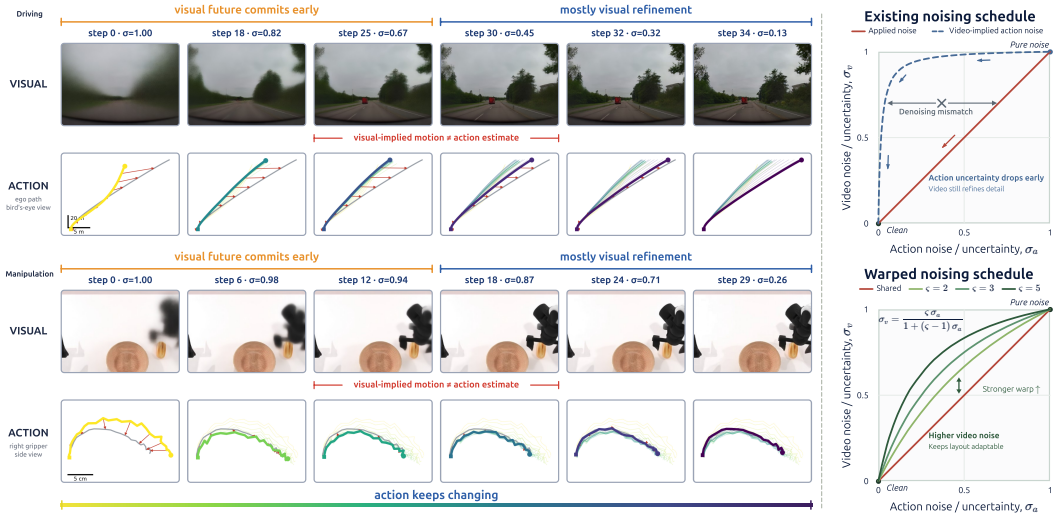


Figure 3: Visual and action predictions evolve at different rates during denoising. Left: driving and manipulation examples show that visual structure becomes established earlier, while the estimated ego and gripper trajectories continue to change. Columns indicate denoising steps and noise levels. Right: schematic schedules illustrate the mismatch under shared noising. Our proposed warp instead assigns higher video noise at a given action noise level.

Warped video–action noise schedules aim to pace this process. They keep the layout open to revision while the actions evolve, and then let it settle into detail refinement as the actions approach convergence. We encourage this progression by assigning video a higher noise level than actions during training. At a given action noise level, the model therefore sees a more strongly corrupted version of the corresponding render, which limits its reliance on an already clear visual layout. Building on separate noise levels for different modalities (Zhu et al., 2025), we adapt the timestep-shift formula-

tion of Esser et al. (2024) and set

$$\sigma_v = \mathrm{warp}(\sigma_a; \varsigma) = \frac{\varsigma \sigma_a}{1 + (\varsigma - 1)\sigma_a}, \tag{1}$$

where ς controls the strength of the warp.

Setting $\varsigma = 1$ recovers the shared schedule $\sigma_v = \sigma_a$. For $\varsigma > 1$, the video input contains more Gaussian noise and less rendered content at a fixed action noise level, and the two levels remain deterministically coupled. Each predicted modality receives its own timestep embedding, and actions supplied as conditioning are kept fixed.

Along this training path, the video remains more corrupted while the action is uncertain, and both modalities approach their clean endpoints as the action noise level decreases. The intended effect is to leave room for changes in scene layout before generation concentrates on fine detail.

3.3 Structured multi-agent control

Interactive driving would benefit from both predicting the behavior of surrounding agents and supporting explicit control over their actions. Even when no commands are provided, the likely maneuvers of these agents shape the ego vehicle's decisions, so predicting them matters for modeling the ego trajectory. The ego-only action interface leaves these futures implicit and provides no dedicated channel for commanding individual agents. We therefore extend the shared transformer with additional agent streams that represent surrounding agents explicitly, supporting both joint behavior prediction and selective control.

The extended action input contains one ego block and K agent blocks, each with history and future tokens (Figure 4). When no future actions are supplied, the model jointly predicts the ego trajectory, the trajectories of the other agents, and the video. The predicted agent futures serve as estimates of each agent's intent, so that ego-action prediction can account for anticipated motions and interactions. When commands are available, the corresponding agent tokens are supplied clean, and the model predicts the video tokens. A conditioning mask implements these roles. Noised future tokens are denoised as prediction targets, whereas clean future tokens act as commands. In both prediction and controlled generation, the surrounding agents thus remain explicitly represented in the model.

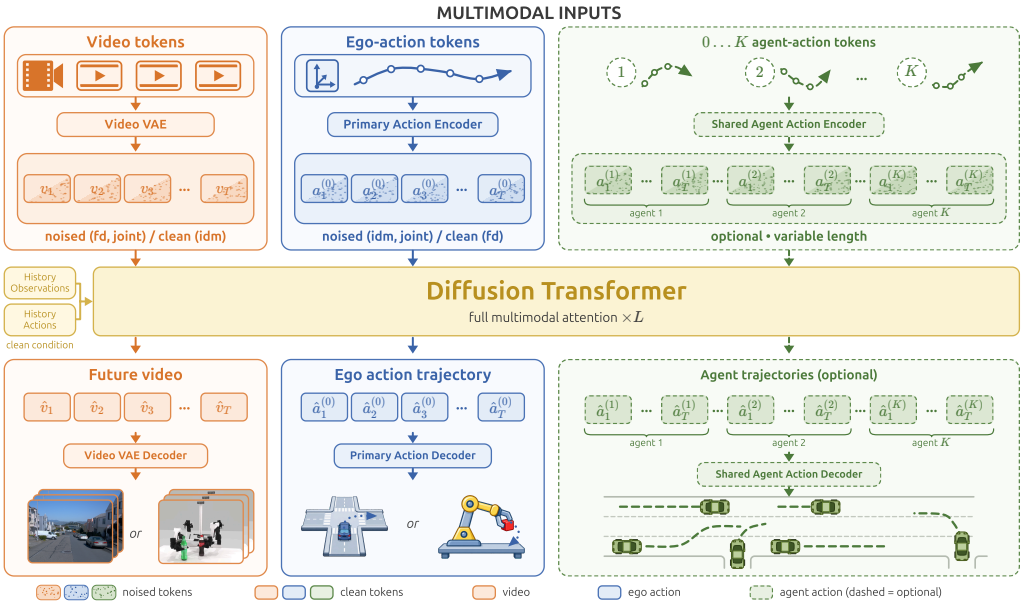


Figure 4: The architecture of CtrlWAM. A shared diffusion transformer processes video, ego-action, and optional agent-action tokens, conditioned on observed history. Future tokens are denoised as prediction targets or supplied clean as conditions, enabling joint prediction, action-conditioned video generation, and video-conditioned action prediction. The additional agent streams support prediction and control of surrounding agents.

4 EXPERIMENTS

4.1 DATASETS AND EVALUATION METRICS

Driving datasets. We use recorded sequences from NVIDIA's PhysicalAI Autonomous Vehicles dataset (NVIDIA, 2025a) and 3DGS scene reconstructions from NuRec (NVIDIA, 2025b) to generate perturbed renderings. Each recorded clip is around 20 seconds. We slice each clip into multiple short simulation windows over approximately 8,900 reconstructed scenes, yielding 33,988 usable clip windows after kinematic, encoding, and timestamp-pairing checks. The training split provides approximately 270,000 aligned items with up to eight perturbation levels per window. Appendix B provides the data inventory details.

Driving evaluation. The held-out evaluation set contains 1,156 clips and 1,434 event-centered segments. We train a separate inverse dynamics model (IDM) to estimate trajectories from videos, implying the intent embedded in the video. In *Joint mode*, the model predicts future video and actions together. In *forward dynamics*, future actions are supplied as commands and the model predicts only video. The main benchmark uses recorded future actions as commands; the counterfactual tests replace them with modified trajectories.

Driving metrics. Using the standard ADE/FDE definitions (Gupta et al., 2018), in Joint mode, *Self ADE* compares the IDM read-out of the generated video with the model's own predicted trajectory. It measures video–action consistency. *Rollout ADE* compares the directly generated action trajectory with the recording and measures action forecasting accuracy without an IDM read-out. In forward dynamics, we report *Cmd ADE* by comparing the IDM read-out from the generated video with the trajectory specified by the supplied action command. We also report future-frame peak signal-to-noise ratio (PSNR) (Hore & Ziou, 2010) for visual reconstruction quality.

Robotics datasets. We use RoboTwin 2.0 (Chen et al., 2025) for bimanual manipulation. We curated 59,696 simulator-derived samples consisting of the perturbed action-rendering pairs. Generated videos are evaluated on 1,000 held-out episodes using WorldArena (Shang et al., 2026).

Robotics metrics. Our WorldArena Track-1 evaluation reports 15 metrics grouped into six dimensions: visual quality, motion quality, content consistency, physics adherence, 3D accuracy, and controllability (Shang et al., 2026). EWMScore is 100 times the mean of the 15 individual metrics. Appendix D gives the metric definitions and full breakdown.

4.2 IMPLEMENTATION DETAILS

Driving model and optimization. All experiments are conducted on NVIDIA H100 GPUs. We use Cosmos 3-Nano (16B) (Agarwal et al., 2026) with its language pathway frozen while training the video and action transformer. Images are stored at 832×480 resolution. We represent driving actions using controls rather than camera poses, as detailed in Appendix B.

Driving ablations start from the same checkpoint and use a fresh optimizer for 10,000 training steps. The learning-rate schedule includes 1,000 warmup steps and cosine decay over the final 2,000 steps. For the ablations, we use two nodes with 16 GPUs, global batch size 16, and peak learning rate 10^{-5}. For the full training, we use 8 nodes with a global batch size of 64. Evaluation uses nine history frames, the same text context, and 35 UniPC sampling steps (Zhao et al., 2023).

Robotics model and generation. We use Cosmos 3-Nano (Agarwal et al., 2026) as the backbone and fine-tune it with the bimanual action space from RoboTwin (Chen et al., 2025). To generate complete WorldArena episodes, the model produces video autoregressively in 32-frame chunks, using the final generated frame to condition the next chunk. Generation continues to the recorded episode length. Appendix D provides the action representations and training settings.

4.3 DRIVING RESULTS

4.3.1 MAIN DRIVING COMPARISON

Counterfactual ego control. Figure 5 shows responses to modified ego commands with the initial scene held fixed: stopping reduces forward viewpoint progression, acceleration advances its position relative to road boundaries and parked vehicles. We report the quantitative results of different



Figure 5: Counterfactual ego control with CTRLWAM. Rows show the generation under the factual ego, and stop, accelerate commands. Columns show corresponding times from 1.8 to 6.4 s; the small plots at right show the ego trajectories associated with each row.

Table 1: Driving evaluation under joint prediction and action-conditioned video generation. In Joint mode, Self ADE compares the IDM read-out of predicted video with the model's own action trajectory; Rollout ADE compares the generated action trajectory with the recording. In forward dynamics, Cmd ADE/FDE compare the IDM read-out with the supplied action command. CTRLWAM-S and CTRLWAM-M denote the ego-only and multi-agent model, respectively.

	Joint mode			Forward dynamics		
Model	PSNR (dB) ↑	Self ADE (m) ↓	Rollout ADE (m) ↓	PSNR (dB) ↑	Cmd ADE (m) ↓	Cmd FDE (m) ↓
Cosmos 3-Nano	19.58	6.212	8.267	20.05	1.600	4.040
SFT Baseline	20.38	0.538	2.920	21.23	1.023	2.803
CTRLWAM-S	20.34	0.521	2.848	21.23	1.001	2.722
CTRLWAM-M	**20.42**	**0.513**	**2.804**	**21.28**	**0.981**	**2.671**

warp strength in Figure 6. As the perturbation level increases, the Cmd ADE of both schedules increase. However, the warp schedule degrade less compared to the shared schedule with GT pairs, and consistently outperforming the one without aligned noising or warp schedule.

Command following and motion forecasting. CTRLWAM improves following of supplied ego commands relative to the SFT baseline (Table 1). In forward dynamics, both Cmd ADE and FDE decrease, indicating that video-inferred motion more closely matches the command. In Joint mode, Self ADE decreases from 0.538 to 0.513 m, showing closer agreement between generated video and actions. Rollout ADE decreases from 2.920 to 2.804 m, showing more accurate action forecasts against the recording.

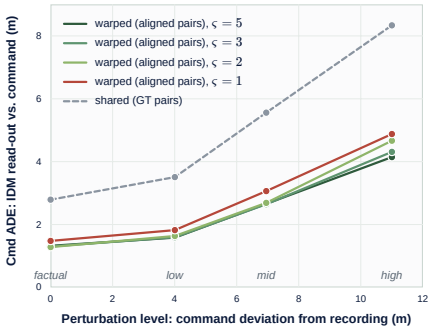


Figure 6: Ego-command counterfactuals on 100 clips. Each level is averaged over four commands (brake to stop, accelerate, nudge left, nudge right). Warped schedules (ς = 2, 3, 5) follow commands more closely than the shared schedule (ς = 1) at every level, with the gap widening as perturbations grow.

4.3.2 MULTI-AGENT CONTROLLABILITY

Our proposed multi-agent interface supports both predicting unspecified agent futures and conditioning video on supplied agent trajectories. To evaluate the controllability, we hold scene history, the ego command, and sampling noise fixed while changing a selected agent's future. In Table 2, we input the generated videos and commanded trajectories to the Gemini 3.8 Flash model, asking it to rate the compliance on a scale of 0 to 1. This measures whether the intended agent follows the requested behavior in the generated video. Compared with the ego-only model, the multi-agent interface improves the mean compliance score from 10.07% to 18.94%, reporting the best performance over other methods, because these models like Cosmos 3 Nano and the SFT baseline cannot receive control signal for other agents.

Figure 7 complements the scores with videos generated under matched factual and counterfactual commands. We stop the vehicle in front of the ego agent, and from the generated video, we can observe that the commanded agent clearly follows the input action.

Table 2: Counterfactual agent-command following. Scene history, the ego command, and sampling noise are held fixed while a selected agent's future trajectory changes. Gemini 3.8 Flash evaluates agreement between the generated video and the supplied agent command; scores are reported on a percentage scale.

Model	VLM compliance (%) ↑
Cosmos 3-Nano	9.13
SFT Baseline	10.56
CTRLWAM-S	10.07
CTRLWAM-M	18.94



Figure 7: Counterfactual control of a surrounding vehicle. The top row uses the agent's factual trajectory; the bottom row replaces it with a stop command. Columns show matching times, and the top-down view at right show the unchanged ego path, factual and modified agent command.

Table 3: WorldArena Track-1 summary (Shang et al., 2026) on RoboTwin manipulation (1,000 episodes). Rows report six category averages and EWMScore, which is 100× the mean of the 15 individual metrics. Oracle evaluates ground-truth videos and is excluded from rankings.

Category	Cosmos 3 Zero-shot	Cosmos 3 SFT	GT-Src	CtrlWAM	Oracle
Visual Quality	0.447	0.560	<u>0.563</u>	**0.576**	0.598
Motion Quality	0.247	<u>0.316</u>	0.306	**0.327**	0.348
Content Consistency	0.413	<u>0.650</u>	0.620	**0.665**	0.662
Physics Adherence	0.258	0.456	<u>0.480</u>	**0.540**	0.782
3D Accuracy	0.812	0.884	<u>0.899</u>	**0.921**	0.956
Controllability	0.635	0.774	<u>0.788</u>	**0.819**	0.869
EWMScore	44.88	<u>58.70</u>	58.65	**61.75**	66.92

Best <u>Second best</u> By metric; ties share formatting.

4.4 ROBOTICS RESULTS

We evaluate whether aligned noising improves manipulation video generation under matched training budgets on the WorldArena benchmark (Shang et al., 2026).

Training controls. The main comparison starts from a common checkpoint trained for 2,000 iterations. Each arm receives 2,000 additional iterations using aligned renders (CTRLWAM), clean recorded frames in place of renders (GT-Src), or standard SFT of Cosmos 3-Nano.

Within the comparison in Tab. 3, CTRLWAM achieves the highest score in all six categories, with EWMScore 61.75 versus 58.70 for SFT and 58.65 for GT-Src. Figure 8 compares a recorded bowl-manipulation sequence with generated videos at the same frame indices. In this example, CTRLWAM more closely follows the recorded object motion and final configuration.

4.5 ABLATIONS

Render source and video–action consistency Table 4 groups the eight-level Joint results by visual source, pairing shared and warped schedules within each group. GT-Src pairs perturbed actions with recorded future frames. Rendering uses simulator-generated frames directly, while Cleaned processes these frames with Difix3D+ (Wu et al., 2025) using the timestamp-paired recorded frames as a reference to remove the artifacts from the Gaussian renderings (NVIDIA et al., 2025).

The schedule comparison depends on the visual source. GT-Src performs better with the shared schedule, whereas Rendering and Cleaned favor the warped configuration for action errors. We believe that the warp mechanism delays the video denoising, not aligned with the original training schedule using GT pairs. For Rendering inputs, the warped configuration lowers these errors from 0.916/3.153 to 0.798/3.019 m. Cleaned inputs show the largest reduction, from 0.915/3.128 to 0.786/2.881 m, giving the lowest Self ADE and Rollout ADE among all six configurations.



Figure 8: Qualitative RoboTwin manipulation comparison. Rows show the recorded video and generations from Cosmos-SFT, the GT-Src control, and CtrlWAM at matching frame indices. The trajectory-accuracy annotations apply to this episode, for which CtrlWAM more closely reproduces the recorded bowl motion and final arrangement.

Table 4: Joint-mode render-source comparison. Colors rank all six configurations within each metric.

Source	Schedule	PSNR(dB) ↑	Self ADE(m) ↓	Rollout ADE(m) ↓
GT-Src	shared	20.67	0.845	2.968
GT-Src	warped	20.58	0.864	3.045
Rendering	shared	21.21	0.916	3.153
Rendering	warped	20.87	0.798	3.019
Cleaned	shared	21.23	0.915	3.128
Cleaned	warped	21.19	0.786	2.881

Best Second best Third best By metric; ties share a shade.

Table 5: Joint-mode warp-strength sweep. When $\varsigma = 1$, the model reduces to the shared schedule.

Warp ς	Self ADE (m) ↓	Rollout ADE (m) ↓
1	0.916	3.153
2	0.935	3.119
3	0.862	3.081
5	**0.798**	**3.019**

Best Second best By metric; ties share formatting.

Visual reconstruction instead favors shared scheduling for all three sources; the Cleaned configuration achieves the highest PSNR (21.23 dB). This is the cost from the warped scheduling, because the videos are held on higher-level noises. However, the minor degradation in PSNR is almost negligible compared to the gains in Self ADE and Rollout ADE, as shown in Tab. 4.

Warped schedules improve video–action consistency Table 5 varies warp strength on Rendering inputs. Increasing ς from 1 to 5 lowers Self ADE from 0.916 to 0.798 m, although the intermediate $\varsigma = 2$ setting slightly worsens consistency. Warping therefore improves video–action consistency at selected strengths.

5 Conclusion

CtrlWAM addresses intent–foresight misalignment by training on perturbed actions paired with their simulated visual consequences. Warped video–action noise schedules aim to keep visual layout responsive while actions evolve, and multi-agent streams allow surrounding futures to be predicted or supplied as commands. Driving experiments show gains in command following, video–action consistency, and action forecasting; robotics controls support improved motion fidelity. Together, these findings support physically aligned denoising as a practical route toward world action models whose imagined outcomes better reflect their actions with improved controllability.

Limitations. The current model now inherits the bidirectional architecture of Cosmos 3, therefore, the generation latency is a bottleneck at this moment. We plan to distill the slow bidirectional model into a fast auto-regressive model and report the close-loop performance in future work.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks that require disclosure. *Implementing methods:* coding agents (Anthropic Claude Code, with OpenAI Codex as an independent read-only reviewer) implemented parts of the training and evaluation code, including the multi-agent action-stream extension and its revision, the counterfactual evaluation scripts, the agent-guidance sampler, and the job launch and monitoring scripts, under author specification. *Interpreting results:* the tools aggregated training logs and evaluation summaries, drafted comparisons across checkpoints, and proposed readings of the counterfactual outcomes. *Supporting qualitative analysis:* Gemini 3.8 Flash serves as the automatic judge in the counterfactual controllability evaluation of Table 2.

Additionally, we used generative AI tools for tasks with recommended disclosure: revising parts of the manuscript, including appendix sections that describe implementation details and results; editing text for readability; editing LaTeX and table formatting; creating scripts that plot figures; summarizing experiment logs. We have reviewed all AI-assisted work. We checked the accuracy of the AI-polished text and the reasonableness of the suggested training parameters. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide implementation and evaluation details in the main paper and appendix. We will publicly release our source code, including training and evaluation scripts, experiment configurations, and data preparation instructions, to facilitate reproducibility.

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.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C.Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando De Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 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. *arXiv preprint arXiv:2506.18088*, 2025.

Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. Worldscore: A unified evaluation benchmark for world generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 27713–27724, 2025.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 2786–2793. IEEE, 2017.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. *Advances in Neural Information Processing Systems*, 37:91560–91596, 2024.

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.

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264. IEEE, 2018.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pp. 2366–2369. IEEE, 2010.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

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.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model, 2025. *arXiv preprint arXiv:2503.00200*, 2025.

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

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. Evalcrafter: Benchmarking and evaluating large video generation models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22139–22149. IEEE, 2024.

Ge Ya Luo, Gian Mario Favero, Zhi Hao Luo, Alexia Jolicoeur-Martineau, and Christopher Pal. Beyond fvd: Enhanced evaluation metrics for video generation quality. *arXiv preprint arXiv:2410.05203*, 2024.

Jing Lyu, Shuanghao Bai, Runze Xiao, Zhenyu Liao, Wenxing Tan, Zihan Tang, Ruochuan Shi, Cheng Peng, Yuheng Ji, Yihao Wang, et al. Dido: Distilling interaction-centric dynamics into one-step denoising for world action models. *arXiv preprint arXiv:2609.15570*, 2026.

NVIDIA. PhysicalAI Autonomous Vehicles. Hugging Face dataset, 2025a. URL https://huggingface.co/datasets/nvidia/PhysicalAI-Autonomous-Vehicles.

NVIDIA. PhysicalAI Autonomous Vehicles NuRec. Hugging Face dataset, 2025b. URL https://huggingface.co/datasets/nvidia/PhysicalAI-Autonomous-Vehicles-NuRec.

NVIDIA, Yulong Cao, Riccardo de Lutio, Sanja Fidler, Guillermo Garcia Cobo, Zan Gojcic, Maximilian Igl, Boris Ivanovic, Peter Karkus, Janick Martinez Esturo, Marco Pavone, Aaron Smith, Ellie Tanimura, Michal Tyszkiewicz, Michael Watson, Qi Wu, and Le Zhang. Alpasim: A modular, lightweight, and data-driven research simulator for autonomous driving, October 2025. URL https://github.com/NVlabs/alpasim.

Yu Shang, Zhuohang Li, Yiding Ma, Weikang Su, Xin Jin, Ziyou Wang, Lei Jin, Xin Zhang, Yinzhou Tang, Haisheng Su, et al. Worldarena: A unified benchmark for evaluating perception and functional utility of embodied world models. *AI Open*, 7:208–226, 2026.

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

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.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *European conference on computer vision*, pp. 55–72. Springer, 2024.

Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. Difix3d+: Improving 3d reconstructions with single-step diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26024–26035. IEEE, 2025.

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.

Hu Yue, Siyuan Huang, Yue Liao, Shengcong Chen, Pengfei Zhou, Liliang Chen, Maoqing Yao, and Guanghui Ren. Ewmbench: Evaluating scene, motion, and semantic quality in embodied world models. *arXiv preprint arXiv:2505.09694*, 2025.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:49842–49869, 2023.

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. *arXiv preprint arXiv:2504.02792*, 2025.

APPENDIX CONTENTS

A Aligned denoising and optimization **14**

A.1 Paired states and corrective targets 14

A.2 Loss weighting and recorded-data anchors 14

A.3 Sampling 14

B Driving data and experimental protocol **15**

B.1 Data and aligned exports 15

B.2 Model inputs and initialization 15

B.3 Evaluation and main ablations 16

C Counterfactual ego-control protocol and examples **16**

D Robotics protocol and metric breakdown **17**

D.1 Training and generation 17

D.2 WorldArena evaluation 18

E Selected-agent interface and evaluation **18**

E.1 Representation and selection 18

E.2 Matched training and evaluation 19

F Discussion and limitations **19**

A ALIGNED DENOISING AND OPTIMIZATION

A.1 PAIRED STATES AND CORRECTIVE TARGETS

With $\sigma = 0$ denoting clean data, standard flow matching (Liu et al., 2022; Lipman et al., 2022) constructs $\mathbf{x}_\sigma = \sigma\epsilon + (1-\sigma)\mathbf{x}^{\mathrm{GT}}$ and regresses

$$\boldsymbol{f}^{\mathrm{std}} = \epsilon - \mathbf{x}^{\mathrm{GT}}. \tag{2}$$

Here $\mathbf{x}$ denotes either video or actions. The two modalities receive independent Gaussian draws; history tokens remain clean and are excluded from the loss.

For each recording, we independently sample Gaussian action noise and construct a perturbed command

$$\mathbf{a}_t^{\mathrm{cmd}} = t\mathbf{a}^\star + (1-t)\epsilon_a, \qquad \epsilon_a \sim \mathcal{N}(0, I), \qquad \sigma_a = 1 - t. \tag{3}$$

The levels are predefined: $t \in \{0, 0.25, 0.5, 0.75\}$ for the four-level grid, with $\{0.125, 0.375, 0.625, 0.875\}$ added for eight levels. AlpaSim executes the noised commands in the reconstructed scene (NVIDIA et al., 2025). We store these executed noised commands as $\mathbf{a}_t$, together with their rendered observations; the timestamp-paired recorded actions and video supply clean targets. Real history remains the conditioning prefix.

At training time, $\mathbf{a}_{\sigma_a} = \mathbf{a}_t$ is used without additional action noising. Given rendered video latents $\mathbf{r}$, fresh video noise produces $\mathbf{v}_{\sigma_v} = \sigma_v\epsilon_v + (1-\sigma_v)\mathbf{r}$. Shared scheduling uses $\sigma_v = \sigma_a$; warped scheduling uses Equation 1. For either modality at its own $\sigma > 0$, the target and clean-data estimate are

$$\boldsymbol{f}^* = \frac{\mathbf{x}_\sigma - \mathbf{x}^{\mathrm{GT}}}{\sigma}, \qquad \hat{\mathbf{x}}_0 = \mathbf{x}_\sigma - \sigma\boldsymbol{f}_\theta(\mathbf{x}_\sigma, \sigma, \mathrm{cond}). \tag{4}$$

A.2 LOSS WEIGHTING AND RECORDED-DATA ANCHORS

Multiplying velocity MSE by the modality's squared noise level gives endpoint regression:

$$\mathcal{L}_{\mathbf{x}} = \sigma^2\|\boldsymbol{f}_\theta - \boldsymbol{f}^*\|^2 = \|\sigma\boldsymbol{f}_\theta - (\mathbf{x}_\sigma - \mathbf{x}^{\mathrm{GT}})\|^2 = \|\hat{\mathbf{x}}_0 - \mathbf{x}^{\mathrm{GT}}\|^2. \tag{5}$$

Predictions and targets are scaled in fp32. This removes explicit low-noise amplification of endpoint error, without implying equal empirical gradients across levels.

Each training window contributes one recorded video–action anchor trained with the original unweighted flow-matching objective and fresh Gaussian draws. Size-proportional sampling gives approximately 20% anchors on four levels and 11% on eight levels. Anchors retain exposure to real video and the continuous pretraining noise distribution.

A.3 SAMPLING

Generation uses UniPC (Zhao et al., 2023) in its data-prediction form. At each step the network velocity is converted to the clean estimate $\hat{\mathbf{x}}_0 = \mathbf{x}_\sigma - \sigma\boldsymbol{f}_\theta$ of Equation 4, and the multistep predictor–corrector update is formed from the history of these estimates.

Paired grids for warped checkpoints. The inference protocol that mirrors warped training runs the two modalities on distinct grids over the same raw times: the video block on the shifted grid, $\sigma_{v,k} = \mathrm{warp}(t_k; \varsigma)$, and the action block on the unshifted grid, $\sigma_{a,k} = t_k$, so that $\sigma_{v,k} = \mathrm{warp}(\sigma_{a,k}; \varsigma)$ at every step, exactly as during training, and both reach $\sigma = 0$ on the same step. Each step makes one joint network forward in which video tokens carry the embedding of $\sigma_{v,k}$ and action tokens the embedding of $\sigma_{a,k}$. The predicted velocity is split by block, and two independent UniPC solvers, each with its own multistep history, update their block from their own clean estimate $\hat{\mathbf{x}}_0 = \mathbf{x}_\sigma - \sigma\boldsymbol{f}_\theta$ at their own level.

Commands held fixed. In forward dynamics and agent control, the supplied future action tokens are marked by the conditioning mask and enter the network at $\sigma = 0$ with their clean values. They are excluded from the noised state, their velocity is zero, and they are fixed points of the solver, so they are never updated. Only the video tokens are integrated, conditioned on the fixed command at every step. This is the mechanism that holds history tokens clean during training, and it is how the counterfactual command experiments of Section 4.3 and Appendix C are run.

B DRIVING DATA AND EXPERIMENTAL PROTOCOL

B.1 DATA AND ALIGNED EXPORTS

PhysicalAI Autonomous Vehicles supplies the recordings (NVIDIA, 2025a); its NuRec release supplies the reconstructed scenes (NVIDIA, 2025b). The export sweeps cover 8,924 scenes and 33,988 windows. Validation and test exports contain clean rollouts and their paired recordings. Table 6 gives the retained item counts; rejected renders mean that not every training window has every perturbation level.

Table 6: Training data inventory (training split unless noted).

	Windows	**Items**
Aligned, 4-level grid ($\leq$4 levels/window)	33,988	134,198
Aligned, 8-level grid ($\leq$8 levels/window)	33,988	269,151
Ground-truth anchored	33,988	33,988
Validation (real, pre-training set)	—	94,232 segments
Validation (simulation scenes, alternative)	2,034	2,034
Golden benchmark (real, standard driving evaluation)	1,156 clips	1,434 segments

Export checks. Exports require complete camera streams, a uniform clock, 1.6 s of real history, timestamp pairing, finite poses and actions, a 64×100 ms action grid, and successful simulator decoding. Rollout-versus-dump disagreement must be below 5 cm; action round-trip error must be below 0.5 m and 0.05 rad.

Rendering uses simulator frames directly. Cleaned applies reference-conditioned Difix3D+ (Wu et al., 2025). Both variants share timestamps and the 832×480 resize operator. Validation uses real samples with the standard objective.

B.2 MODEL INPUTS AND INITIALIZATION

B.2.1 PACKED INPUTS.

The Cosmos 3-Nano backbone couples future video and action tokens through full attention while retaining causal attention on context tokens. Each input contains nine history and 32 future frames at 5 Hz, with a 6.4 s future horizon. The causal VAE compresses time by $4\times$ and space by $16\times$, giving 11 latent frames with 48 channels at 30×52 resolution; three latent frames encode history. The action stream contains 14 history rows derived from 16 recorded poses and 64 future rows at 0.1 s intervals, padded to 64 channels. History is conditioned throughout; forward dynamics additionally conditions future actions.

B.2.2 UNICYCLE ACTIONS.

Each action row stores acceleration a_t and curvature κ_t. Encoding fits smoothed heading to recorded yaw and speed to displacement projected along that heading, with initial speed estimated from history. Acceleration is the regularized finite difference of speed; curvature is $\Delta\theta_t/s_t$, where $s_t = v_t\Delta t + \frac{1}{2}a_t\Delta t^2$. Acceleration is standardized with mean 0.029 and standard deviation $0.681\,\mathrm{m/s^2}$; curvature uses mean 2.7×10^{-4} and standard deviation $0.0261\,\mathrm{m^{-1}}$. Feasibility bounds are $|a| \leq 9.8\,\mathrm{m/s^2}$ and $|\kappa| \leq 0.33\,\mathrm{m^{-1}}$. Only these two channels enter the noise, loss, and sampler; padding is masked. Decoding integrates from the recorded t_0 state. Counterfactual paths use the same encoding and decoding operators.

B.2.3 SUPERVISED INITIALIZATION.

The video and action experts are initialized from Cosmos 3-Nano (Agarwal et al., 2026) and trained, with the Wan 2.2 (Wan et al., 2025) VAE fixed. Four PhysicalAI-AV NVIDIA (2025a) sources are sampled equally: chain-of-thought, chain-of-thought with meta-actions, 500k meta-action-only windows, and 500k trajectory-only windows. Joint, forward, and inverse tasks have probabilities 0.4/0.4/0.2. The pretraining uses 256 H100 GPUs, batch 256, and 20,000 steps; AdamW with

$\beta = (0.9, 0.999)$, $\epsilon = 10^{-6}$, weight decay 10^{-5}, gradient clipping 1.0, bf16 and fully sharded data parallelism; peak learning rate 1.5×10^{-4} after 2,000 warmup steps, cosine decay to 10^{-7}, and a $5\times$ action-input-projection multiplier. Noise levels are uniform and the objective is standard flow matching. The step-20,000 checkpoint initializes the driving post-training runs.

B.3 Evaluation and main ablations

B.3.1 Evaluation.

The standard benchmark contains 1,156 clips and 1,434 segments with paired sampling noise. Self ADE compares video-inferred motion with the jointly predicted trajectory; Rollout ADE directly compares the predicted trajectory with the recorded GT trajectory. In forward dynamics, Cmd ADE/FDE compare the IDM read-out with the supplied command. IDM estimates the ego trajectory from the given video; Cmd FDE is computed from that same selected trajectory. The real-video IDM reference is 0.523/1.304 m ADE/FDE. The 100-clip counterfactual evaluation in Appendix C is randomly selected from the held-out evaluation set.

B.3.2 Optimization.

The main driving ablations start from the same pretrained checkpoint and use a fresh optimizer for 10,000 steps, 16 H100 GPUs, global batch 16, and peak learning rate 10^{-5}. The schedule uses 1,000 warmup steps and cosine decay over the final 2,000 steps. Full ego-only and multi-agent training uses the matched eight-node, batch-64 recipe in Appendix E.

B.3.3 Schedule-comparison settings.

Table 4 compares render sources under different schedules. For warped rows we use $\varsigma = 5$. Table 5 is conducted on the Rendering sources (not cleaned by Difix3D (Wu et al., 2025) model) and varies $\varsigma \in \{1, 2, 3, 5\}$, with the same initialization, step count, batch size, and learning rate.

C Counterfactual ego-control protocol and examples

Figure 6 evaluates 100 randomly sampled held-out clips. History frames, text, and sampling noise are held fixed across command variants. Commands use the training-time unicycle encoding, and Cmd ADE compares the IDM read-out with the supplied command.

Figure 9 adds highway and intersection examples to Figure 5; these illustrate command-dependent changes rather than an additional aggregate result.

Perturbation protocol. Each counterfactual replaces the recorded ego trajectory with one of four edited commands: *brake to stop* decelerates at a constant rate until the ego halts; *accelerate* increases speed at a constant rate; and *nudge left/right* offsets the recorded path laterally by a distance. Longitudinal strengths are in m/s^2 and lateral strengths in metres, so severity is defined by pairing one strength per type. The low/mid/high settings use braking decelerations of $1/2/3$ m/s^2, accelerations of $0.75/1.5/3$ m/s^2, and left/right offsets of $2/3.5/5$ m, respectively.

At each severity level, we average Cmd ADE over the four command types. The common horizontal axis is the mean ADE between the edited command and the recording, identical for every checkpoint: 4.0 m (low), 7.0 m (mid), and 11.0 m (high). The factual recorded command is the leftmost point at 0 m deviation.

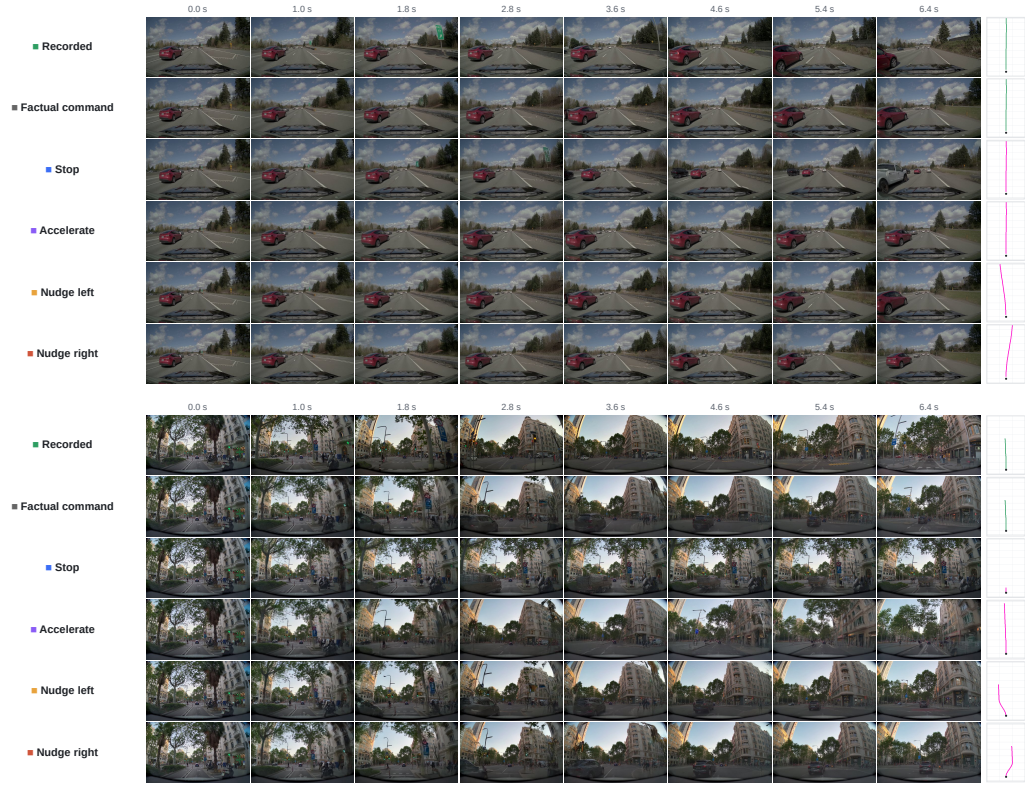


Figure 9: Additional counterfactual ego control on a highway (top) and at an urban intersection (bottom). Each panel compares the recording with generations under factual, stop, accelerate, nudge-left, and nudge-right commands at matching times over 0.0–6.4 s. Plots at right show the corresponding ego trajectories.

D Robotics protocol and metric breakdown

D.1 Training and generation

The main comparison uses 14-dimensional absolute joint positions (12 arm joints and two grippers) in 32-step chunks. A common joint-space SFT checkpoint trained for 2,000 iterations initializes three further 2,000-iteration stages: continued SFT, GT-Src, and aligned post-training. Each stage restarts the optimizer and schedule. GT-Src replaces aligned rendered frames with real video while retaining the perturbed actions and aligned recipe. Each aligned stage uses 59,696 simulator-derived samples and a 6:2 aligned-to-anchor mixture.

An aligned sample pairs its conditioning frame and 32 simulator frames with the executed noised command $\mathbf{x}_\sigma$, at $\sigma = 1 - t_{\text{clean}}$ and $t_{\text{clean}} \in \{0, 0.25, 0.5, 0.75\}$. A Gaussian draw sampled independently per window ; the clean rollout is rendered as a validity gate. Action states enter verbatim, while video latents receive fresh Gaussian noise. Both modalities use fp32 endpoint supervision with conditioning rows masked; anchors use the standard objective. The SFT loss scale and action-loss weight remain 10.

Training uses eight H100 GPUs, FusedAdam, nominal learning rate 10^{-4}, weight decay 0.05, and a $5\times$ action-projection multiplier. A linear multiplier decays from 0.4 to zero over each 2,000-iteration stage without warmup. Token packing uses 45,056 tokens per rank, approximately 11 samples per rank and global batch 88. Generation uses forward dynamics with 30 UniPC steps, shift 10, guidance 1, and seed 0. The last generated frame conditions the next chunk until the recorded episode length is reached. Robotics output is 736×544 at 30 Hz; the fixed 6.4 s horizon and 832×480 resolution apply to driving.

D.2 WORLDARENA EVALUATION

Evaluation uses 1,000 held-out RoboTwin episodes and the WorldArena harness (Shang et al., 2026). Table 7 consolidates the individual metrics for the models in Table 3. Visual and consistency metrics build on VBench (Huang et al., 2024); Flow Score follows EvalCrafter (Liu et al., 2024); motion smoothness and photometric consistency follow WorldScore (Duan et al., 2025). JEPA similarity follows JEDi (Luo et al., 2024), and trajectory accuracy uses the normalized dynamic-time-warping comparison adopted by EWMBench (Yue et al., 2025). Other metric definitions and normalization follow WorldArena. Flow, trajectory, photometric, and motion scores use the harness's published percentile-based min–max bounds. EWMScore is 100 times the mean of the 15 normalized metrics.

Table 7: Individual WorldArena metrics for Table 3. SFT, GT-Src, and CTRLWAM share a joint-space SFT 2k initialization and receive 2k additional iterations. Zero-shot uses text conditioning; Oracle scores recorded videos. EWMScore is 100 times the mean of 15 normalized metrics. Oracle JEPA and depth (*) are identity-comparison checks; photometric consistency (†) can favor smoother-than-real motion.

Metric	Zero-shot	SFT	GT-Src	CTRLWAM	Oracle
Image Quality	0.398	0.396	0.394	0.394	0.401
Aesthetic Quality	0.387	0.377	0.375	0.371	0.396
JEPA Similarity	0.555	0.905	0.921	0.963	0.998*
Dynamic Degree	0.142	0.199	0.190	0.212	0.229
Flow Score	0.073	0.099	0.095	0.109	0.125
Motion Smoothness	0.525	0.651	0.634	0.660	0.690
Subject Consistency	0.440	0.711	0.667	0.737	0.753
Background Consistency	0.475	0.762	0.714	0.785	0.811
Photometric Consistency†	0.325	0.476	0.478	0.474	0.423
Interaction Quality (VLM)	0.417	0.611	0.633	0.668	0.719
Trajectory Accuracy	0.099	0.302	0.327	0.412	0.845
Depth Accuracy	0.772	0.918	0.930	0.955	1.000*
Perspectivity (VLM)	0.853	0.850	0.867	0.887	0.911
Instruction Following (VLM)	0.405	0.651	0.678	0.738	0.830
Semantic Alignment	0.865	0.897	0.897	0.899	0.907
EWMScore (mean of 15 $\times$ 100)	44.88	58.70	58.65	61.75	66.92

Individual metrics do not uniformly favor aligned training: continued SFT has higher image and aesthetic quality. Aggregate improvements therefore reflect gains across several measures rather than dominance on every metric. Ground-truth oracle scores are calibration references rather than strict upper bounds, because the GT videos provided by WorldArena (Shang et al., 2026) is of lower resolution than the one used in our experiments.

E SELECTED-AGENT INTERFACE AND EVALUATION

E.1 REPRESENTATION AND SELECTION

For each scene, the interface contains an ego block and K selected surrounding-agent blocks, with $0 \le K \le 8$ varying across scenes:

$$\mathbf{a} = [\mathbf{a}^{(0)} \| \mathbf{a}^{(1)} \| \cdots \| \mathbf{a}^{(K)}], \qquad \mathbf{a}^{(k)} = [\mathbf{a}^{(k)}_{\text{hist}} \| \mathbf{a}^{(k)}_{\text{fut}}]. \tag{6}$$

Each block contains 14 history and 64 future rows at 0.1 s intervals. Only selected agents receive explicit action blocks and trajectory supervision. Histories are clean context; all selected futures are prediction targets in Joint mode, or clean commands when supplied for controlled generation. With $K = 0$, the input reduces to the ego-only layout.

Agent histories contain ego-frame (x, y) positions at t_0, scaled by 20 m, while future unicycle controls use each actor's local t_0 frame. Predicted trajectories are transformed to the recorded trajectories' frame for evaluation. Tracks require finite states throughout the horizon, feasible controls, and an encode–decode error below 0.5 m. Missing leading history is back-filled from the earliest finite state.

Agent selection and encoding . Static actors with net displacement below 2 m are excluded before packing. Remaining actors must project into the front image at positive depth, lie within 60 m, span at least 12 px in height at 832×480, and pass the projected-occlusion test (a nearer retained box contains the candidate centre with a 1 m depth margin). The nearest eligible actors up to $K_{\max} = 8$ are retained. Actor tokens use a separate embodiment projection initialized from the ego projection, and spatial rotary coordinates given by the actor's projected t_0 position on the 15×26 latent grid. Temporal rotary positions match the ego block. For $K > 0$, ego future rows are weighted by $1 + K$ and each selected-agent block by $(1 + K)/K$; for $K = 0$, no agent loss is computed.

E.2 Matched training and evaluation

We first pretrain the model as described in Sec. B.2.3. CtrlWAM-S and CtrlWAM-M use the same initialization, training data, aligned-noising recipe, training duration, optimizer, learning-rate schedule, batch size, and evaluation protocol. They differ only in the additional agent interface. Both start from the pretrained checkpoint and use eight nodes, batch 64, learning rate 3×10^{-5} for 20k steps. For a fair comparison, we continue to finetune the pretrained checkpoints using the standard noising with the same setting, reporting the SFT baseline results. The Agent ADE/FDE of 3.215/7.224 m compare generated trajectories directly with recordings: average within each evaluation batch over selected actors, then average batches with batch-size weights. Under selection, 4,084 actors in 1,197 of the 1,434 benchmark segments enter these metrics. Agent prediction errors are not evaluated in forward dynamics, where future actions are supplied, because we do not have an inverse dynamic models (IDM) to estimate the agents' trajectories from the given video.

Counterfactual actor control. For Table 2, scene history, the ego command, and sampling noise are held fixed while the selected actor's future changes. CtrlWAM-M receives the edited agent command through its agent interface. Cosmos 3 zero-shot, the SFT baseline, and CtrlWAM-S receive no non-ego agent command; their scores measure agreement with the requested agent behavior without conditioning on that command.

Gemini 3.8 Flash receives the generated video and commanded trajectory and assigns a compliance score $s_i \in [0, 1]$. The reported percentage is $100\, N^{-1} \sum_{i=1}^{N} s_i$, where N is the number of scored examples. We use 100 clips for evaluation, each with four type of counterfactual commands. This is a mean judge score, with no thresholding into success or failure.

F Discussion and limitations

The experiments assess generated futures and offline open-loop metrics. Self ADE measures video–action consistency, Rollout ADE measures direct forecasting accuracy, and Cmd ADE/FDE measure agreement with supplied commands through an IDM. Most robotics metrics also rely on external models. Both the evaluation is heavily dependent on the used models, leading to a risk of the model bias.

Limitations. The aligned noising pipeline requires a simulator to execute perturbed actions and render their consequences. The multi-agent extension is limited to driving because we lack suitable dynamic robotics datasets with actor-level action–observation pairs. We expect future real–sim–real workflows to enable reconstruction of multi-agent interactions, generation of controlled variations, and transfer of learned behavior back to real environments, making multi-agent prediction and control in robotics feasible.