HOLO-WORLD: UNIFIED CAMERA, OBJECT AND WEATHER CONTROL FOR VIDEO WORLD MODEL

Anonymous authors
Paper under double-blind review

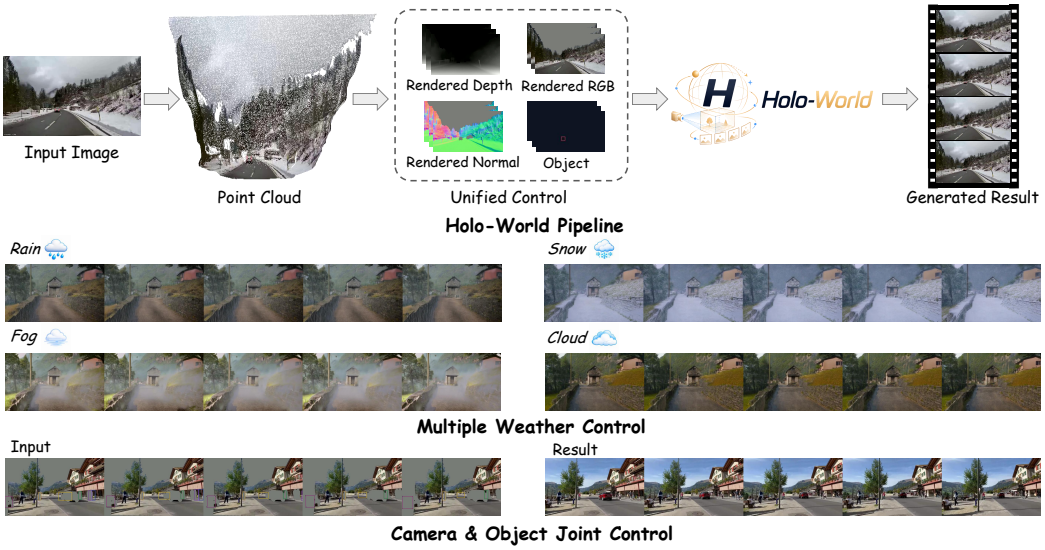


Figure 1: **Unified state control in Holo-World.** Holo-World jointly controls camera motion, object dynamics, and weather state within the same observed world.

ABSTRACT

Video world models are moving toward preserving an observed world under controllable camera and object motion while allowing its environmental state to change. Yet these controls remain isolated, and weather generation typically relies on a source video or reconstructed scene that already specifies future structure. We study a first-frame-anchored source-to-state setting, where the model starts from a single image and follows explicit camera and object controls and an optional weather instruction, then generates a video that either preserves the source world or transfers it to a target weather state. To address these challenges, we first build **HoloStateData**, a state video dataset that turns diverse videos into unified control samples for camera, object, and weather supervision. Second, we introduce **Holo-World**, a unified controllable video world model that jointly controls the scene from a single image. Its Unified Scene Adapter factorizes world preservation and weather transfer into distinct parameter subspaces, using rendered background, geometry buffers, and object controls to maintain controlled scene structure while modeling weather-dependent appearance and particle effects. Additionally, Scene-Weather Decomposed CFG guides scene and weather residuals separately, strengthening target weather effects without over-amplifying the full condition. Quantitative and qualitative experiments demonstrate that Holo-World maintains precise camera and object controls with consistent scene structure while transferring scenes into diverse target weather states, outperforming video-to-video weather editing baselines on weather-state generation. Our project page is available at https://anonymous.4open.science/w/Holo-Anonymous-Page/, and an offline page with video demonstrations is included in the supplementary material.

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

Video world models (Bruce et al., 2024; Li et al., 2025a; Zhang et al., 2025a; He et al., 2025; Huang et al., 2025a; Gao et al., 2025) are moving from generating visually plausible videos toward simulating dynamic worlds under structured conditions. Future observations in video world modeling depend not only on how the camera moves and how objects evolve, but also on the environmental state in which the same scene is rendered. Current controllable video methods usually model these factors separately, with camera-control methods (Ren et al., 2025; Yang et al., 2026; Huang et al., 2025a) specifying viewpoint changes, object-control methods (Li et al., 2025b; Chu et al., 2026; Liu et al., 2025a; Bar et al., 2025) steering dynamic entities, and weather-oriented methods (Liu et al., 2026; Lin et al., 2025) synthesizing or editing environmental appearance. Existing weather generation methods are largely formulated as video-to-video editing or simulation (Ali et al., 2025; Qian et al., 2026; Kerbl et al., 2023), where a complete source video or reconstructed scene already provides future layout, motion, and temporal continuity.

However, for a video world model these future states should be generated by the model, making joint camera, object, and weather control a different problem from video-to-video weather editing. Although recent works have begun to couple camera and object control (Zheng et al., 2026; Cao et al., 2025; Zhang et al., 2026), weather state is still rarely integrated into the same generation process. We therefore study a first-frame source-to-state setting, where a model observes a single image and synthesizes future frames under explicit camera and object controls while rendering the same scene into a target weather state (Lin et al., 2025; Mou et al., 2024; Zhu et al., 2025). This gap stems from two coupled obstacles: (1) the lack of data that jointly captures camera, object, and weather controls, and (2) the modeling conflict between world preservation and weather-state transfer within the same controllable model.

The data obstacle follows from the separation of existing controllable video methods. Existing methods (Wang et al., 2025c; 2024; Qian et al., 2026) supervise parts of this control space, but their data assumptions are not organized around a shared control interface. These advances (Wang et al., 2026; Li et al., 2026) provide important control signals, but they do not supply training samples in which camera pose (He et al., 2024; Li et al., 2025c), background geometry (Liu et al., 2025b; Wang et al., 2025a), object controls (Zhang et al., 2025b; Wu et al., 2024), and target weather-state supervision are jointly observed. Appendix Table 6 summarizes the resulting capability gap.

The modeling obstacle is to support both background preservation and weather transfer within one model. Real videos provide supervision for source appearance and geometry consistency, while paired weather videos provide supervision for weather-state transfer under matched structural controls. If these two objectives are entangled in the same adapter parameter space, the learned residuals can either suppress weather changes or disturb camera, object, and background consistency. The same entanglement appears during sampling, where vanilla classifier-free guidance (Ho & Salimans, 2022) scales a global residual that couples scene preservation with weather changes, so stronger guidance may enhance weather effects while also over-amplifying the source scene.

To provide supervision for unified control, we build **HoloStateData**, a state video dataset that organizes real videos, paired simulated weather videos, and weather-transferred videos into samples aligned with camera poses, geometry anchors, object controls, scene text, and weather-state supervision. We introduce **Holo-World**, a unified controllable video world model that jointly controls camera motion, object dynamics, and scene weather state from a single image. As shown in Figure 1, Holo-World supports both background-consistent generation and weather-state transfer under the same structured control setting. To address these challenges, we introduce Unified Scene Adapter (**UniSA**), instantiated with a World Adapter and a State Adapter that share one frozen video backbone but learn separate residual subspaces for world preservation and weather transfer. UniSA grounds camera-conditioned generation with rendered background controls, uses G-buffers to stabilize scene structure, and incorporates object controls to constrain dynamic entities, allowing weather-dependent appearance and particle effects to emerge within the same controlled scene rather than relying on a cascaded video-to-video editor. During sampling, Scene-Weather Decomposed CFG (**SW-CFG**) applies the same separation at the guidance level, using a scene residual to preserve the controlled world and an independent weather residual to strengthen weather effects without over-amplifying the full condition. Extensive quantitative and qualitative experiments on our benchmark demonstrate that Holo-World maintains precise camera and object control with consistent scene structure while transferring scenes into diverse target weather states.

Our contributions are summarized as follows:

- We formulate unified state control for video world models, where a single image-to-video model follows camera and object controls while supporting both world preservation and weather-state transfer.
- We build HoloStateData, a state video dataset that turns diverse videos into unified control samples for camera, object, and weather supervision.
- We introduce Holo-World with Unified Scene Adapter, which factorizes world preservation and weather-state transfer into distinct parameter subspaces within one frozen video backbone, together with Scene-Weather Decomposed CFG for independently guiding scene and weather residuals during sampling.
- Extensive experiments demonstrate that Holo-World maintains precise camera and object controls and consistent scene structure while generating scenes into diverse weather states, outperforming video-to-video weather editing baselines on weather generation.

2 Related Work

Video world models. Video world models learn dynamic scene evolution by predicting future world states. Recent generative world models scale this paradigm with transformer and diffusion backbones for action-conditioned generation (Bruce et al., 2024; Zhang et al., 2025a; He et al., 2025; Li et al., 2025a; Sun et al., 2025). Geometry-aware methods further incorporate explicit 3D reconstruction (Liu et al., 2025b; Wang et al., 2025a) to support world exploration and navigable scene generation (Huang et al., 2025a; Yu et al., 2025a; Sun et al., 2024; Team et al., 2026; Shen et al., 2026; Mildenhall et al., 2021). These methods advance dynamic world synthesis, but their controllable state spaces are usually limited to actions, text, camera motion, or reconstructed geometry. A unified video world model interface that jointly controls camera motion, object dynamics, and environmental state remains largely unexplored.

Video generation with camera and object motion control. Controllable video generation has moved from text-only conditioning (Kong et al., 2024) toward explicit structural controls. Camera-aware and 3D-aware methods (Wang et al., 2024; He et al., 2024; Yu et al., 2025b; Ren et al., 2025; Wu et al., 2024; Wang et al., 2025c; Shi et al., 2024; Li et al., 2025c) inject trajectories, poses, rendered geometry, or learned camera motion into video diffusion models, enabling viewpoint control and image-conditioned novel-view generation. Object-control and camera-object methods (Zheng et al., 2026; Yang et al., 2026; Cao et al., 2025; Zhang et al., 2026) further steer dynamic entities with 4D geometric controls, trajectories (Fu et al., 2025; Chu et al., 2026), regions, boxes (Li et al., 2025b; Zhang et al., 2025b), or latent motion guidance.

Weather editing and environment-state generation. Weather-oriented methods (Lin et al., 2025; Liu et al., 2026; Qian et al., 2026) synthesize or edit rain, snow, fog, wet surfaces, snow coverage, overcast skies, and related environmental effects. Some methods formulate weather control as video-to-video editing (Ali et al., 2025; Jiang et al., 2025), where a full source video already provides future layout, motion, and temporal continuity. Others rely on reconstructed 3D scenes, driving assets, multi-view inputs, maps, or multimodal video controls to constrain the target sequence. These designs are effective for weather transfer under strong evidence, but they do not directly address single-image video generation where the model must follow explicit camera and object controls while synthesizing the target weather state.

3 HoloStateData

HoloStateData is designed around this requirement by representing each clip as a source-to-state record rather than a standalone captioned video. The first frame and source-side controls define the observed world, including camera-conditioned background motion, static geometry anchors, and object boxes. When paired weather supervision is available, the target video and weather prompt specify the scene state to synthesize. This shared interface supports both no-edit background preservation on real videos and weather-state transfer on paired weather videos.

As shown in Figure 2, HoloStateData is built through three stages: data collection, video annotation, and scene construction. Data collection gathers real videos, paired simulated weather videos, and weather-transferred videos via V2V generation. Video annotation extracts factorized scene and

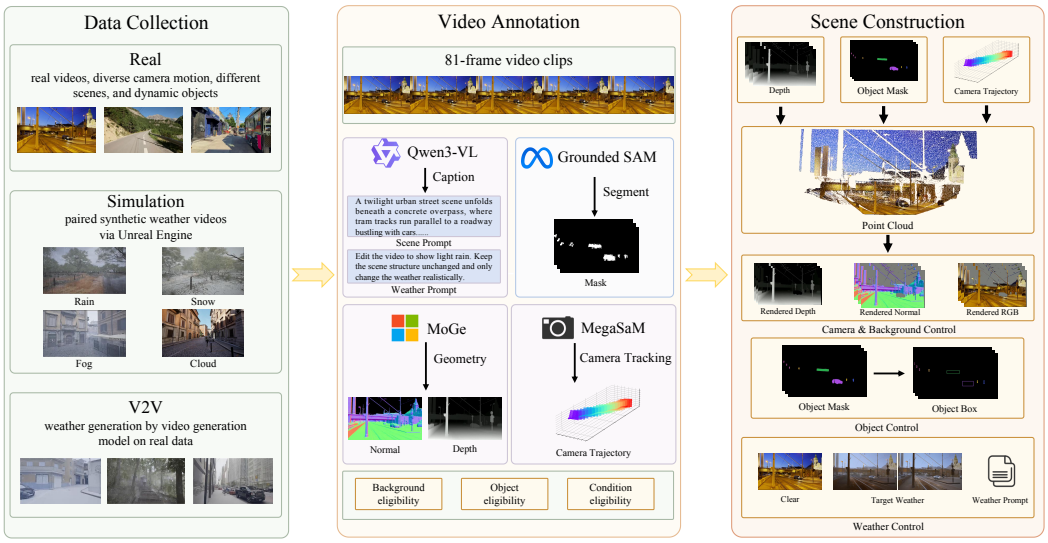


Figure 2: **HoloStateData construction pipeline.** These annotations are converted into source-side rendered controls and object controls, while paired target-weather videos provide supervision for weather-state transfer.

weather text, object masks, camera motion, and dense geometry. Scene construction renders source-side background RGB, depth, and normal controls, converts object masks into object controls, and associates paired samples with target-weather supervision. The central rule is that target-weather pixels are never used to render model-facing controls, they only provide the supervised target video.

Data sources. We first use SpatialVID-HQ (Wang et al., 2026) as the real-video pool, which provides in-the-wild scenes with diverse camera trajectories, object dynamics, and layouts. We then use open-source paired simulated weather videos rendered with Unreal Engine, where the same synthetic scenes are captured under clear and target weather states (Lin et al., 2025). These paired videos share scene structure while differing in weather appearance, providing clean supervision for weather-state transfer. Finally, we use SpatialVID-HQ clips as source videos and synthesize paired target-weather videos with a combination of our weather-transfer model and proprietary video editing models, extending weather supervision to real camera trajectories and object dynamics while complementing the more controlled synthetic pairs.

Automatic annotation and scene construction. Following in (Zheng et al., 2026; Wang et al., 2026), we generate factorized text, object, and geometry controls for each retained 81-frame clip. Qwen3-VL (Bai et al., 2025) produces a scene prompt that describes scene identity, objects, actions, temporal evolution, and camera motion while excluding weather. Simulation and V2V samples additionally receive a weather prompt that only describes the target weather state, whereas Real samples use no weather-edit prompt. Object controls are built by detecting first-frame objects, initializing and propagating masks with Grounded-SAM2 (Liu et al., 2024; Ravi et al., 2025), and converting each per-frame mask into a tight bounding box as model-facing object control. For camera/static-world control, Depth Anything (Yang et al., 2024) and UniDepth (Piccinelli et al., 2024) provide monocular depth priors, MegaSaM (Teed & Deng, 2021; Li et al., 2025d) estimates camera parameters, and MoGe (Wang et al., 2025b) supplies dense depth and normal anchors for rendering source-side background RGB, depth, and normal videos. Scene construction enforces source-target decoupling: target videos and weather prompts supervise the desired weather state, but rendered background controls and object boxes are always built from the source side. This keeps world-consistent controls separate from weather-state supervision. Appendix A.5 provides additional details on our dataset.

Data subsets. These sources are organized into three subsets with complementary supervision roles. The Real subset stabilizes the source-world interface using real camera motion, object dynamics, and scene layouts without a weather-edit target. The Simulation subset provides aligned synthetic weather pairs with shared scene structure and controlled target-weather changes. The V2V subset exposes the model to weather transfer on real videos, complementing the synthetic pairs with real trajectories and dynamic entities. After filtering, HoloStateData contains 15K training samples. We also reserve a mutually separated 150-sample benchmark dataset, organized into Real, Simulation, and V2V subsets with 50 samples each.

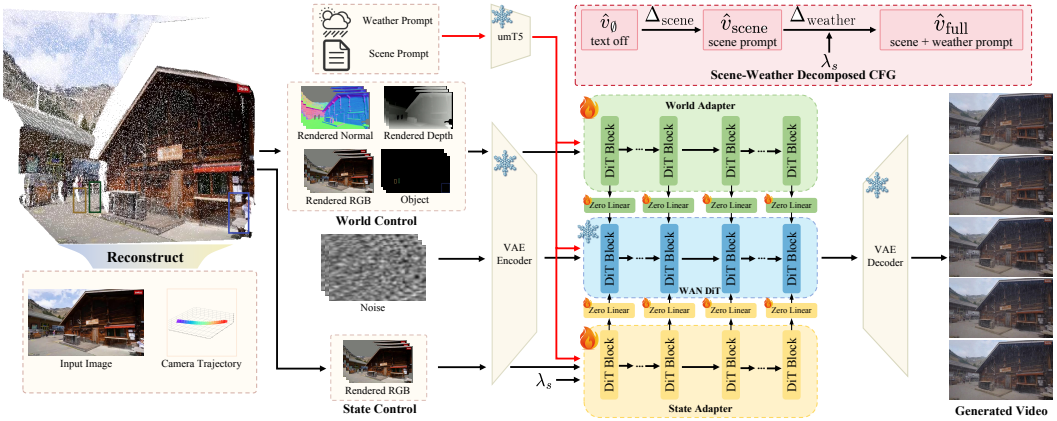


Figure 3: **Overview of Holo-World.** Given a first frame and factorized source-to-state controls, Holo-World decomposes controllable video generation into world-preservation and weather-transfer residual paths.

Metrics. On the Real subset, we report VBench-I2V (Huang et al., 2025b) for video quality, rotation error (RotErr), translation error (TransErr) (He et al., 2024) and ObjMC (Wang et al., 2024) for camera and object-control errors, and background metrics for background preservation. The background metrics compute PSNR, SSIM, and LPIPS between generated frames and rendered RGB on valid static regions, so they directly measure whether the controlled background remains stable. Weather metrics are not reported on the Real subset because there is no target weather state. On the Weather subset (Simulation/V2V), we report Weather Alignment, VLM Evaluation, and User Study. Weather Alignment uses the target weather prompt and generated video to judge whether the requested core weather condition appears in the result, and reports the aligned-sample ratio. VLM Evaluation uses the same target weather prompt and generated video, but asks a VLM evaluator to score weather background editing, particle dynamics, atmospheric coherence, and overall visual quality on a 0–100 score. User Study reports blind pairwise human preference between Holo-World and each weather-editing baseline. Appendix A.6 gives the weather evaluation prompts and aggregation rules, and Appendix A.6.8 details the background preservation metrics.

4 Holo-World

Holo-World is built around a factorized source-to-state interface that contains the observed world to preserve and the target weather state to transfer. Unified Scene Adapter (UniSA) injects structured controls into a frozen Wan base model through preservation-oriented and weather-oriented residual subspaces, while Scene-Weather Decomposed CFG (SW-CFG) applies the same separation at inference time to avoid a single global guidance direction that entangles scene preservation and weather transfer. Figure 3 summarizes the architecture.

4.1 Control Interface and Training

Holo-World uses a first-frame source-to-state interface with structured controls and factorized text. Given the input frame I_0, C_{world} represents source-side world controls, C_{state} is the source appearance anchor for state transfer, c_{scene} is the scene prompt, and c_{state} is the weather prompt. All rendered controls are constructed from the source side before generation. During training, the automatic construction pipeline uses the first frame, camera trajectory, object masks, and dense source geometry to render background controls and derive object boxes. At inference, the same interface can be filled by an explicit camera trajectory, object controls from Grounded-SAM2, or user interaction, and dense first-frame geometry from MoGe. Target-weather videos are never used to render these controls, so the model does not receive target-side appearance leakage. The world and state controls are

$$C_{\text{world}} = \{R_{\text{rgb}}, R_{\text{depth}}, R_{\text{normal}}, C_{\text{bbox}}\}, \qquad C_{\text{state}} = \{R_{\text{rgb}}\}. \tag{1}$$

where R_{rgb}, R_{depth}, and R_{normal} are RGB, depth, and normal videos rendered from the first-frame geometry and camera trajectory, and C_{bbox} is the object control video derived from propagated masks. The same rendered RGB is also used as C_{state}, serving as a source appearance anchor for weather-state transformation rather than a target-weather image or generated weather proxy.

Rendered background controls specify camera-conditioned static-world evolution, bbox videos constrain dynamic entities, and the weather prompt defines the target scene state relative to the source appearance anchor. Depth and normal maps serve as geometric anchors of structure consistency, reducing floating texture artifacts and structure drift.

Holo-World keeps the Wan flow-matching (Lipman et al., 2022) objective and only updates UniSA parameters. Let $c = \mathrm{merge}(c_{\mathrm{scene}}, c_{\mathrm{state}})$ be the global text condition received by the frozen backbone. Given the noisy latent z_t at timestep t, the model predicts the Wan flow target v_t:

$$\mathcal{L}_{\mathrm{fm}} = \mathbb{E}_{\mathcal{D},\epsilon,t}\left[\left\| f_{\theta,\phi}(z_t, t, I_0, C_{\mathrm{world}}, C_{\mathrm{state}}, c) - v_t \right\|_2^2\right]. \tag{2}$$

The VAE, text encoder, and Wan backbone parameters θ are frozen, and $\phi = \{\phi_w, \phi_s\}$ represent the trainable World and State Adapter parameters. Real samples supervise no-edit preservation with V_{origin}, while Simulation and V2V samples supervise weather-state transfer with V_{target}.

4.2 Unified Scene Adapter

The central modeling conflict is that background preservation and weather-state generation require different residual behaviors. Real samples supervise source appearance, camera-consistent background structure, and object-control adherence. Weather-paired samples keep the same structural controls but require changes in visibility, sky condition, wetness, snow coverage, fog density, and other scene-state cues. If both objectives are absorbed by the same adapter parameter space, preservation supervision can suppress weather changes, while weather supervision can disturb geometry and object adherence.

Unified Scene Adapter addresses this conflict with one adapter interface and two parameter-disjoint subspaces. The World Adapter learns preservation residuals for camera, background geometry, and object layout, and the State Adapter learns weather-transfer residuals from the source appearance anchor and target weather text:

$$h_\ell^w = A_\ell^w(C_{\mathrm{world}}, c_{\mathrm{scene}}), \tag{3}$$

$$h_\ell^s = A_\ell^s(C_{\mathrm{state}}, c_{\mathrm{state}}). \tag{4}$$

where A_ℓ^w and A_ℓ^s are the World and State Adapter blocks aligned with the ℓ-th injected Wan block. The two blocks have the same DiT structure, produce residual hints with the same hidden shape, and do not share trainable parameters. They operate on the same frozen video backbone and therefore form a single end-to-end model rather than a cascaded weather editor.

Let F_ℓ represents the ℓ-th frozen Wan DiT block and let $x_{\ell-1}$ represents the input hidden state. For injected layers $\ell \in \mathcal{L}$, UniSA augments the backbone update with the two residual hints:

$$x_\ell = F_\ell(x_{\ell-1}, c) + \lambda_w h_\ell^w + \lambda_s h_\ell^s. \tag{5}$$

Layers outside $\mathcal{L}$ follow the original pretrained Wan forward process. We use $\mathcal{L} = \{0, 5, 10, 15, 20, 25\}$, initialize the adapter blocks from their corresponding Wan blocks. The world scale is fixed as $\lambda_w = 1$, while the state scale follows the sample type:

$$\lambda_s = \begin{cases} 0, & \text{without weather prompt,} \\ 1, & \text{with weather prompt.} \end{cases} \tag{6}$$

This gate keeps Real samples from forcing the State Adapter to learn identity appearance changes, while Simulation and V2V samples train state transformation on top of the same source controls.

4.3 Scene-Weather Decomposed CFG

Even though UniSA separates world and state residuals inside the network, vanilla classifier-free guidance still extrapolates along one empty-to-full direction, jointly strengthening scene structure, weather cues, appearance shifts, and text interactions, which can enhance weather effects at the cost of source-world preservation:

$$\hat{v} = \hat{v}_\emptyset + s(\hat{v}_{\mathrm{full}} - \hat{v}_\emptyset). \tag{7}$$

where $\hat{v}_\emptyset$ is the empty inference, $\hat{v}_{\mathrm{full}}$ is the full inference, and s is the guidance scale. In Holo-World, the direction $\hat{v}_{\mathrm{full}} - \hat{v}_\emptyset$ is not a pure weather direction. It mixes scene semantics, weather semantics, color and texture shifts, World Adapter effects, State Adapter effects, and control-text interactions. Increasing scale s can make weather more visible, but the same coefficient may overdrive no-edit Real samples with color shifts, over-saturation, or texture artifacts. Reducing scale s preserves Real samples, but weakens weather generation.

Scene-Weather Decomposed CFG follows the UniSA factorization at sampling time. At each denoising step, all branches share the same latent z_t, controls, timestep, and model parameters, only the scene and weather prompt change. The sampler evaluates three predictions:

$$\hat{v}_{\emptyset} = f_\theta(z_t, t, I_0, C_{\text{world}}, C_{\text{state}}, c_{\emptyset}), \tag{8}$$

$$\hat{v}_{\text{scene}} = f_\theta(z_t, t, I_0, C_{\text{world}}, C_{\text{state}}, c_{\text{scene}}), \tag{9}$$

$$\hat{v}_{\text{full}} = f_\theta(z_t, t, I_0, C_{\text{world}}, C_{\text{state}}, c_{\text{scene}}, c_{\text{state}}). \tag{10}$$

The three predictions define two residual directions. The empty prediction serves as a control-only baseline with text removed. The scene prediction restores the scene prompt and measures the residual needed to preserve the controlled world relative to this baseline. The full prediction further includes target-weather text, making $\hat{v}_{\text{full}} - \hat{v}_{\text{scene}}$ the weather-specific residual. For Real samples, no target weather is requested, so $\lambda_s = 0$ removes state residual. For Simulation and V2V samples, $\lambda_s = 1$ keeps it active for weather-state transfer.

The guidance direction is then decomposed into a scene residual and a weather residual:

$$\Delta_{\text{scene}} = \hat{v}_{\text{scene}} - \hat{v}_{\emptyset}, \qquad \Delta_{\text{weather}} = \hat{v}_{\text{full}} - \hat{v}_{\text{scene}}. \tag{11}$$

$$\hat{v} = \hat{v}_{\emptyset} + s_{\text{scene}}\Delta_{\text{scene}} + \lambda_s s_{\text{weather}}\Delta_{\text{weather}}. \tag{12}$$

where s_{scene} is kept low to avoid over-redrawing the source scene, s_{weather} controls the weather residual strength, and λ_s is the same state scale used by UniSA. Thus Real samples suppress the weather residual, while Weather samples concentrate concrete weather changes in Δ_{weather}.

5 Experiments

5.1 Experimental Setup

Benchmark. We evaluate on a held-out benchmark split of HoloStateData, which contains 150 curated samples organized into Real, Simulation, and V2V subsets with 50 samples each. Following prior controllable video generation protocols that use curated evaluation sets to cover key motion and control factors (Zhang et al., 2026; Wang et al., 2024), our benchmark is stratified by source type and evaluation goal rather than sampled as a single mixed test pool. The Real subset contains unpaired samples whose origin and target are the same video, and evaluates whether the model follows camera and object controls for world preservation. The Weather subset combines the Simulation and V2V subsets for weather transfer, and evaluates whether the model can synthesize a target weather state from a single image, source-side camera/object controls, and a weather description. We report Real and Weather results separately because they test different sides of the source-to-state contract.

Baselines and input contract. For Real evaluation, we compare with Uni3C, GEN3C, VerseCrafter, and NeoVerse, which cover camera-aware and camera-object controllable video generation models. For Weather evaluation, we compare with Cosmos-Transfer2.5 and Wan2.7-Edit. These weather baselines are video-to-video editing models and can use a complete source video, whereas Holo-World uses only the first frame together with structured camera, object, and weather controls, we keep this input difference explicit.

Implementation details. Holo-World is built on Wan2.1-T2V-14B, we freeze the VAE, text encoder, and Wan backbone, and train only the World and State Adapter parameters in UniSA, following in (Jiang et al., 2025; Wan et al., 2025). UniSA is inserted at DiT layers $\{0, 5, 10, 15, 20, 25\}$. Training uses 81-frame 480×832 video clips and a text length of 512. The global batch size is 16, learning rate is 2×10^{-5}, weight decay is 0.03, and we train models for 3,000 steps. At inference time, all methods generate 81-frame videos. We use 50 denoising steps, disables vanilla CFG, sets $s_{\text{scene}} = 1.0$ and $s_{\text{weather}} = 2.0$ for SW-CFG.

5.2 Comparison Results

Results of World Preservation. We first evaluate the Real subset, where the model should follow camera and object controls while preserving the observed world. As shown in Table 1, Holo-World achieves the best overall VBench-I2V score compared to other methods, with strong I2V subject and background consistency. Some specialized baselines remain competitive on individual perceptual submetrics, which confirms that video quality alone is insufficient for evaluating unified state control. Table 2 further shows that Holo-World obtains the lowest camera and object-control errors. Together, these results show that training for weather-state transfer does not prevent the model from preserving the source world under structured controls.

Table 1: **Quantitative comparison of different approaches on the Real subset under VBench-I2V evaluation.** The best results are highlighted in **bold**, and the second-best results are underlined.

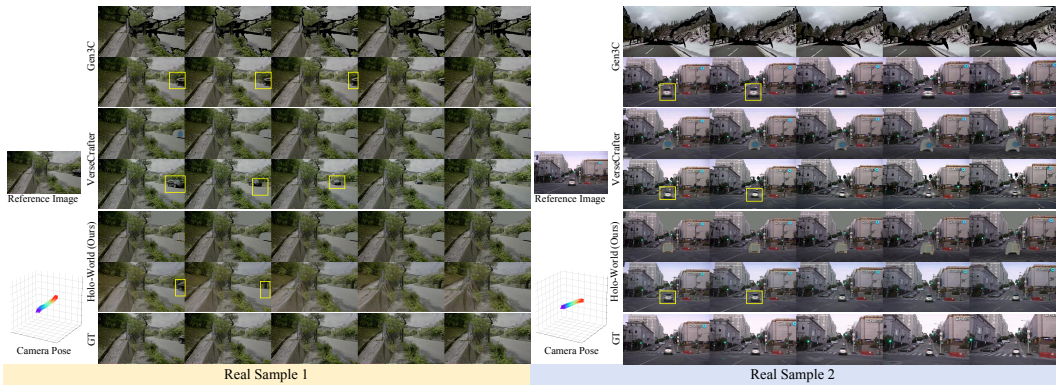
Method	Overall ↑ Score	Subject ↑ Consistency	Background ↑ Consistency	Motion ↑ Smoothness	Dynamic ↑ Degree	Aesthetic ↑ Quality	Imaging ↑ Quality	I2V ↑ Subject	I2V ↑ Background
Uni3C (Cao et al., 2025)	88.25	94.74	93.66	98.95	56.00	**50.12**	70.45	98.45	98.58
GEN3C (Ren et al., 2025)	87.24	**96.79**	**94.28**	**99.46**	18.00	49.84	68.35	99.25	99.30
VerseCrafter (Zheng et al., 2026)	88.41	94.70	93.93	98.96	46.15	48.83	**72.87**	99.28	99.24
NeoVerse (Yang et al., 2026)	88.00	93.49	93.01	99.43	**58.00**	49.39	65.16	98.96	98.93
Holo-World (Ours)	**89.05**	96.27	93.83	99.13	52.00	50.00	72.43	**99.43**	**99.40**

Table 2: **Camera and object-control results on the Real subset.** The best results are highlighted in **bold**, and the second-best are underlined.

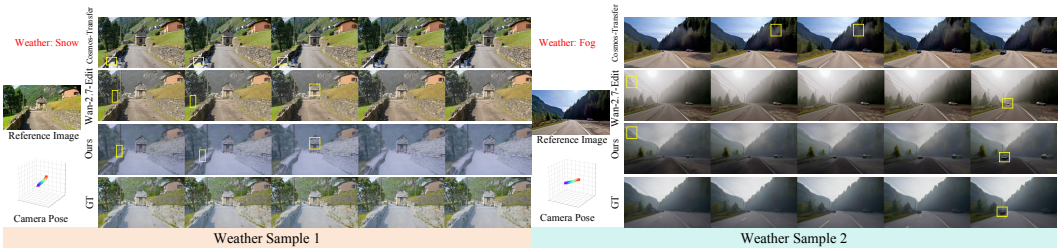
Method	RotErr ↓	TransErr ↓	ObjMC ↓
Uni3C (Cao et al., 2025)	3.036	1.228	7.771
GEN3C (Ren et al., 2025)	3.063	1.156	8.078
VerseCrafter (Zheng et al., 2026)	0.834	1.197	7.002
NeoVerse (Yang et al., 2026)	1.703	2.112	8.859
Holo-World (Ours)	**0.719**	**1.123**	**6.549**

Table 3: **Weather generation and user study results on the Weather subset.** The best results are highlighted in **bold**. User study reports blind pairwise preference rate against Cosmos-Transfer2.5 and Wan2.7-Edit.

Method	Weather ↑ Alignment	VLM ↑ Evaluation	User Study ↑ Cosmos / Wan2.7
Wan2.7-Edit	79.00	61.92	–
Cosmos (Ali et al., 2025)	30.00	47.64	–
Holo-World (Ours)	**86.00**	**68.51**	**83.00 / 62.00**



(a) World preservation.



(b) Weather Transfer.

Figure 4: **Main qualitative comparison.** Real and Weather rows show the source-to-state requirement of preserving the source-controlled world while synthesizing the requested weather state.

Results of Weather Transfer. We next evaluate the Weather subset, where Holo-World must change the weather from a single image under camera and object control. This setting is stricter than other video-to-video editing models because Holo-World cannot inherit future layout, motion, or temporal continuity from a complete source video. Table 3 shows that Holo-World (I2V) outperforms other methods (V2V) on all weather metrics, reaching 86.00% Weather Alignment and 68.51 VLM Evaluation. Human raters also prefer Holo-World in 83.00% of comparisons against Cosmos-Transfer2.5 and 62.00% against Wan2.7-Edit. The qualitative comparison in Figure 4 mirrors these trends. The Real rows preserve the source-controlled world, and the Weather rows show clear weather-state transfer while maintaining consistent camera and object evolution.

5.3 Ablation Studies

Model design. We ablate the components that separate world preservation from weather transfer, including G-buffer controls, UniSA, and SW-CFG. World preservation is evaluated using background metrics on the Real subset, and weather transfer is evaluated using Weather Alignment and VLM Evaluation on the Weather subset. As shown in Table 4, adding G-buffer control improves Background PSNR and SSIM on the Real subset, and raise Weather Alignment and VLM Evalua-

Table 4: **Ablation on the model design.** We progressively enable G-buffer controls, UniSA, and SW-CFG to assess the individual contribution of different components to background preservation and weather controllability.

G-Buffer	UniSA	SW-CFG	Back. PSNR ↑	Back. SSIM ↑	Back. LPIPS ↓	Weather Align. ↑	VLM Eval. ↑
			13.20	0.552	0.320	51.00	51.60
✓			15.34	0.577	0.329	62.00	55.20
✓	✓		16.36	0.598	0.308	58.00	58.53
✓	✓	✓	**18.12**	**0.624**	**0.245**	**86.00**	**68.51**

Table 5: **Ablation on Scene-Weather Decomposed CFG.** We compare no guidance, vanilla CFG, and different SW-CFG weather scales using the same trained UniSA model.

Method	Back. PSNR ↑	Back. SSIM ↑	Back. LPIPS ↓	Weather Align. ↑	VLM Eval. ↑
w/o CFG	18.12	0.624	0.245	51.00	55.68
Vanilla CFG	16.36	0.598	0.308	58.00	58.53
SW-CFG (ω=2)	18.12	0.624	0.245	86.00	68.51
SW-CFG (ω=4)	**18.12**	**0.624**	**0.245**	**96.00**	**78.42**

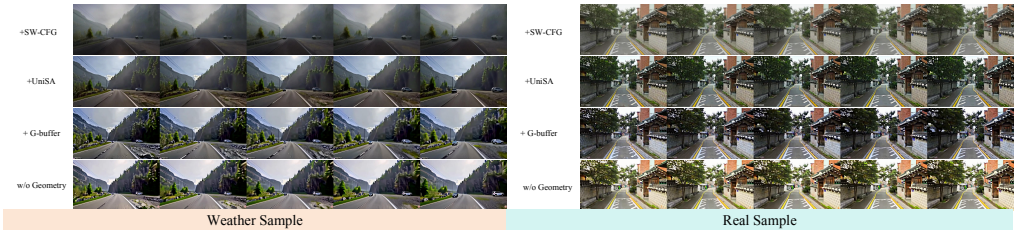


Figure 5: **Visualization of the model design ablation.** The comparison visualizes whether UniSA and SW-CFG reduce interference between Real-subset preservation and Weather-subset editing.

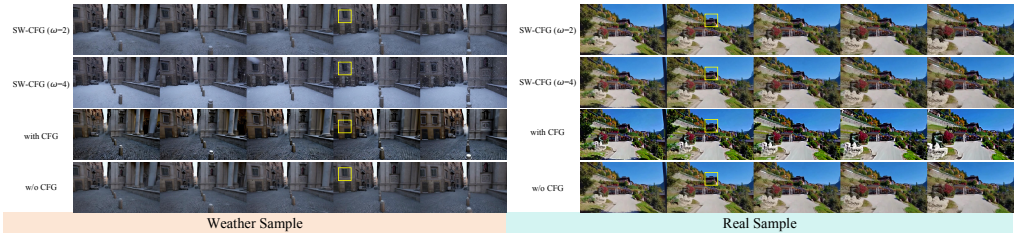


Figure 6: **Visualization of guidance ablation.** The comparison visualizes how no CFG, vanilla CFG, and Scene-Weather Decomposed CFG balance weather strength and source-world preservation.

tion on the Weather subset, indicating that explicit geometry anchors help produce weather effects and preserve the controlled scene. UniSA further improves all three background metrics and VLM Evaluation, but the Weather Alignment score decreases slightly, suggesting that architectural separation alone can make the model conservative about visible cloud, rain, snow, or fog. With SW-CFG, the full model achieves the best background preservation and the strongest weather metrics at the same time. This pattern supports the main design claim that unified state control requires both separated residual subspaces and separated sampling guidance. Figure 5 qualitatively confirms the same trend by comparing preservation and weather-transfer results under matched source controls.

Scene-Weather guidance. We further validate the sampling guidance ablations using the same trained UniSA model. Table 5 shows that vanilla CFG improves weather metrics only modestly and degrades background preservation, which is consistent with the guidance conflict discussed in Section 4.3. In contrast, Scene-Weather Decomposed CFG increases Weather Alignment and VLM Evaluation on the Weather subset while keeping the background metrics close to the no-CFG setting on the Real subset. This indicates that the decomposed weather residual strengthens cloud, rain, snow, and fog generation under the same camera, object, and geometry controls, instead of globally amplifying the full condition. Meanwhile, excessive weather guidance disrupts the scene structure (see $\omega = 4$ in Figure 6), so we balance the weather scale settings in SW-CFG. Figure 6 visualizes this trade-off, and Appendix A.3 provides a detailed visual analysis of how SW-CFG balances weather-strength enhancement and source-world preservation. Appendix Figure 7 provides multi-weather examples under fixed source-world.

6 Conclusion

We presented Holo-World, a controllable video world model for unified camera, object, and weather control from a single observed image. HoloStateData provides source-to-state supervision that separates source-world evidence from target-weather generation, while UniSA and Scene-Weather Decomposed CFG carry this separation into training and sampling. Experiments show that Holo-World preserves background and scene structure while transferring diverse weather states, outperforming V2V baselines on the Weather subset in an I2V setting.

AI USE STATEMENT

We used generative video models to construct paired weather-transfer training data, Qwen3-VL to generate scene and weather annotations, and a GPT model to evaluate weather alignment and video quality. In the period of paper writing, we only exploit the Large Language Model (LLM), i.e., GPT-6, for word polishing to improve the presentation quality. No LLMs contributes to the core scientific methods, technical contents, or any substantive research contributions.

REFERENCES

Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15791–15801, 2025.

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 *Forty-first International Conference on Machine Learning*, 2024.

Chenjie Cao, Jingkai Zhou, Shikai Li, Jingyun Liang, Chaohui Yu, Fan Wang, Xiangyang Xue, and Yanwei Fu. Uni3c: Unifying precisely 3d-enhanced camera and human motion controls for video generation. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–12, 2025.

Ruihang Chu, Yefei He, Zhekai Chen, Shiwei Zhang, Xiaogang Xu, Dingdong WANG, Hongwei Yi, Xihui Liu, Hengshuang Zhao, Yu Liu, et al. Wan-move: Motion-controllable video generation via latent trajectory guidance. In *Advances in Neural Information Processing Systems*, volume 38, pp. 404–432, 2026.

Hyung Won Chung, Xavier Garcia, Adam Roberts, Yi Tay, Orhan Firat, Sharan Narang, and Noah Constant. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining. In *The 11th International Conference on Learning Representations*, 2023.

Xiao Fu, Xian Liu, Xintao Wang, Sida Peng, Menghan Xia, Xiaoyu Shi, Ziyang Yuan, Pengfei Wan, Di Zhang, and Dahua Lin. 3dtrajmaster: Mastering 3d trajectory for multi-entity motion in video generation. In *International Conference on Learning Representations*, volume 2025, pp. 6680–6693, 2025.

Jianxiong Gao, Zhaoxi Chen, Xian Liu, Jianfeng Feng, Chenyang Si, Yanwei Fu, Yu Qiao, and Ziwei Liu. Longvie: Multimodal-guided controllable ultra-long video generation. *arXiv preprint arXiv:2508.03694*, 2025.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, et al. Matrix-game 2.0: An open-source real-time and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

Tianyu Huang, Wangguandong Zheng, Tengfei Wang, Yuhao Liu, Zhenwei Wang, Junta Wu, Jie Jiang, Hui Li, Rynson Lau, Wangmeng Zuo, et al. Voyager: Long-range and world-consistent video diffusion for explorable 3d scene generation. *ACM Transactions on Graphics*, 44(6):1–15, 2025a.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, et al. Vbench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17191–17202, 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

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.

Jiaqi Li, Junshu Tang, Zhiyong Xu, Longhuang Wu, Yuan Zhou, Shuai Shao, Tianbao Yu, Zhiguo Cao, and Qinglin Lu. Hunyuan-gamecraft: High-dynamic interactive game video generation with hybrid history condition. *arXiv preprint arXiv:2506.17201*, 2(3):6, 2025a.

Quanhao Li, Zhen Xing, Rui Wang, Hui Zhang, Qi Dai, and Zuxuan Wu. Magicmotion: Controllable video generation with dense-to-sparse trajectory guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12112–12123, 2025b.

Teng Li, Guangcong Zheng, Rui Jiang, Shuigen Zhan, Tao Wu, Yehao Lu, Yining Lin, Chuanyun Deng, Yepan Xiong, Min Chen, et al. Realcam-i2v: Real-world image-to-video generation with interactive complex camera control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28785–28796, 2025c.

Zhen Li, Chuanhao Li, Xiaofeng Mao, Shaoheng Lin, Ming Li, Shitian Zhao, Zhaopan Xu, Xinyue Li, Yukang Feng, Jianwen Sun, et al. Sekai: A video dataset towards world exploration. In *Advances in Neural Information Processing Systems*, volume 38, 2026.

Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10486–10496, 2025d.

Chih-Hao Lin, Zian Wang, Ruofan Liang, Yuxuan Zhang, Sanja Fidler, Shenlong Wang, and Zan Gojcic. Controllable weather synthesis and removal with video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13580–13591, 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.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024.

Tianyu Liu, Weitao Xiong, Kunming Luo, Manyuan Zhang, Peng Li, Yuan Liu, and Ping Tan. Autoweather4d: Autonomous driving video weather conversion via g-buffer dual-pass editing. *arXiv preprint arXiv:2603.26546*, 2026.

Xinhang Liu, Yuxi Xiao, Donny Y Chen, Jiashi Feng, Yu-Wing Tai, Chi-Keung Tang, and Bingyi Kang. Trace anything: Representing any video in 4d via trajectory fields. *arXiv preprint arXiv:2510.13802*, 2025a.

Yifan Liu, Zhiyuan Min, Zhenwei Wang, Junta Wu, Tengfei Wang, Yixuan Yuan, Yawei Luo, and Chunchao Guo. Worldmirror: Universal 3d world reconstruction with any-prior prompting. *arXiv preprint arXiv:2510.10726*, 2025b.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Chong Mou, Mingdeng Cao, Xintao Wang, Zhaoyang Zhang, Ying Shan, and Jian Zhang. Revideo: Remake a video with motion and content control. In *Advances in Neural Information Processing Systems*, volume 37, pp. 18481–18505, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182, 2023.

Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10106–10116, 2024.

Chenghao Qian, Wenjing Li, Yuhu Guo, and Gustav Markkula. Weatheredit: Controllable weather editing with 4d gaussian field. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 8511–8519, 2026.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6121–6132, 2025.

Tianchang Shen, Sherwin Bahmani, Kai He, Sangeetha Grama Srinivasan, Tianshi Cao, Jiawei Ren, Ruilong Li, Zian Wang, Nicholas Sharp, Zan Gojcic, et al. Lyra 2.0: Explorable generative 3d worlds. *arXiv preprint arXiv:2604.13036*, 2026.

Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, et al. Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations*, 2021.

Wenqiang Sun, Shuo Chen, Fangfu Liu, Zilong Chen, Yueqi Duan, Jun Zhang, and Yikai Wang. Dimensionx: Create any 3d and 4d scenes from a single image with controllable video diffusion. *arXiv preprint arXiv:2411.04928*, 2024.

Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. Worldplay: Towards long-term geometric consistency for real-time interactive world modeling. *arXiv preprint arXiv:2512.14614*, 2025.

HY Team, Chenjie Cao, Xuhui Zuo, Zhenwei Wang, Yisu Zhang, Junta Wu, Zhenyang Liu, Yuning Gong, Yang Liu, Bo Yuan, et al. Hy-world 2.0: A multi-modal world model for reconstructing, generating, and simulating 3d worlds. *arXiv e-prints*, pp. arXiv–2604, 2026.

Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *Advances in neural information processing systems*, volume 34, pp. 16558–16569, 2021.

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.

Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Chang Zeng, Yanxi Zhou, Xiao-Xiao Long, et al. Spatialvid: A large-scale video dataset with spatial annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42592–42603, 2026.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025a.

Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5261–5271, 2025b.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, 2024.

Zun Wang, Jaemin Cho, Jialu Li, Han Lin, Jaehong Yoon, Yue Zhang, and Mohit Bansal. Epic: Efficient video camera control learning with precise anchor-video guidance. *arXiv preprint arXiv:2505.21876*, 2025c.

Jianzong Wu, Xiangtai Li, Yanhong Zeng, Jiangning Zhang, Qianyu Zhou, Yining Li, Kai Chen, and Yunhai Tong. Motionbooth: Motion-aware customized text-to-video generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 34322–34348, 2024.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.

Yuxue Yang, Lue Fan, Ziqi Shi, Junran Peng, Feng Wang, and Zhaoxiang Zhang. Neoverse: Enhancing 4d world model with in-the-wild monocular videos. *arXiv preprint arXiv:2601.00393*, 2026.

Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T Freeman, and Jiajun Wu. Wonderworld: Interactive 3d scene generation from a single image. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5916–5926, 2025a.

Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–18, 2025b. doi: 10.1109/TPAMI.2025.3613256.

Guiyu Zhang, Yabo Chen, Xunzhi Xiang, Junchao Huang, Zhongyu Wang, and Li Jiang. Symphomotion: Joint control of camera motion and object dynamics for coherent video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11127–11137, 2026.

Yifan Zhang, Chunli Peng, Boyang Wang, Puyi Wang, Qingcheng Zhu, Fei Kang, Biao Jiang, Zedong Gao, Eric Li, Yang Liu, et al. Matrix-game: Interactive world foundation model. *arXiv preprint arXiv:2506.18701*, 2025a.

Zhongwei Zhang, Fuchen Long, Zhaofan Qiu, Yingwei Pan, Wu Liu, Ting Yao, and Tao Mei. Motionpro: A precise motion controller for image-to-video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 27957–27967, 2025b.

Sixiao Zheng, Minghao Yin, Wenbo Hu, Xiaoyu Li, Ying Shan, and Yanwei Fu. Versecrafter: Dynamic realistic video world model with 4d geometric control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40277–40290, 2026.

Zehao Zhu, Yuliang Zou, Chiyu Max Jiang, Bo Sun, Vincent Casser, Xiukun Huang, Jiahao Wang, Zhenpei Yang, Ruiqi Gao, Leonidas Guibas, et al. Scenecrafter: Controllable multi-view driving scene editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6812–6822, 2025.

A APPENDIX

A.1 FLOW MATCHING OBJECTIVE

Flow Matching learns a time-dependent velocity field that transports samples between the data distribution and a simple prior distribution (Lipman et al., 2022; Ho et al., 2020; Song et al., 2021). Given z_0 as a clean video latent and $z_1 \sim \mathcal{N}(0, I)$ as a Gaussian latent. For $\tau \in [0, 1]$, the linear probability path is

$$z_\tau = (1 - \tau)z_0 + \tau z_1. \tag{13}$$

The corresponding velocity target is constant along this path:

$$u_\tau(z_0, z_1) = \frac{dz_\tau}{d\tau} = z_1 - z_0. \tag{14}$$

The model therefore predicts the path velocity rather than directly predicting the clean latent. Holo-World uses Wan2.1-T2V-14B as the pretrained latent video backbone (Wan et al., 2025). Wan2.1 compresses videos into a latent space via a 3D VAE, encodes text conditions with a text encoder (Chung et al., 2023), and uses a DiT denoiser (Peebles & Xie, 2023) to predict the flow target over spatio-temporal latent tokens. Holo-World keeps this Wan flow objective unchanged. Given conditioning $\mathcal{C}$, training minimizes

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{z_0, z_1, \tau}\left[\|v_\theta(z_\tau, \tau, \mathcal{C}) - (z_1 - z_0)\|_2^2\right], \tag{15}$$

where θ represents the frozen Wan VAE, text encoder, and DiT backbone. In Holo-World, $\mathcal{C}$ contains the first frame, source-side world controls, source appearance anchors, and factorized text conditions. UniSA only changes how these conditions enter selected Wan DiT blocks through trainable residual adapters, while the latent flow path and velocity objective remain those of the Wan2.1-T2V backbone.

A.2 ADDITIONAL CAPABILITY COMPARISON

Table 6 shows the capability comparison along the control axes exposed by each SOTA method. The goal is not to rank methods by generation quality, but to clarify their evidence contract. Existing image-to-video (I2V) methods expose camera control or joint camera-object control via 3D rendered geometry, while weather-oriented baselines are typically formulated as video-to-video (V2V) editing models that need input a complete source video. Holo-World targets the harder I2V task in which the model takes as inputs only a single image, explicit camera and object controls with geometry, and a target weather state. The model must synthesize future layout, motion, and weather-dependent appearance together, rather than relying on an already available future video as the editing substrate. This makes the weather subset results in Table 3 and the qualitative comparison in Figure 4 especially demanding, since Holo-World outperforms the evaluated V2V weather-editing baselines while operating with weaker source evidence and a broader control interface.

Table 6: **Capability comparison of controllable video generation paradigms.** ✓ denotes an explicit control axis in the method's primary inference interface, and blank cells indicate that the corresponding state factor is not independently exposed.

Method	Paradigm	Camera Control	Object Control	Weather Control
GEN3C (Ren et al., 2025)	I2V	✓		
VerseCrafter (Zheng et al., 2026)	I2V	✓	✓	
Uni3C (Cao et al., 2025)	I2V	✓	✓	
NeoVerse (Yang et al., 2026)	I2V	✓		
Wan2.7-Edit	V2V			✓
Cosmos-Transfer2.5 (Ali et al., 2025)	V2V			✓
Holo-World (Ours)	I2V	✓	✓	✓

A.3 ADDITIONAL RESULTS

This section provides additional results for the world preservation and weather transfer capabilities of our Holo-World. In all samples, the first frame, camera trajectory, rendered geometry controls, and object controls are defined by the source world, while the target-weather prompt specifies the environmental state to synthesize.

Figure 6 further illustrates the guidance conflict that motivates SW-CFG. On the Weather sample, removing CFG preserves the source structure but produces weak weather changes, while vanilla CFG makes the target weather more visible at the cost of over-amplified appearance shifts and less

stable scene structure. On the Real sample, stronger global guidance also changes the source appearance even though no weather transfer is requested, showing that a single CFG direction couples preservation and transfer. SW-CFG separates these two effects: the scene residual keeps the camera-conditioned world and object region stable, while the weather residual can be scaled independently to strengthen rain, snow, fog, or cloud effects. Increasing the weather scale therefore improves weather visibility on Weather samples without globally amplifying the full scene condition, which explains why SW-CFG better balances source-world preservation and weather-state transfer.

Figure 7 evaluates multi-weather results under fixed source-world controls. Across urban streets, residential scenes, mountain roads, and city plazas, changing the weather prompt produces clearly different rain, fog, cloud, and snow states while retaining the same viewpoint progression and scene layout. Rain variants darken the scene and introduce wet or rainy appearance, fog variants reduce long-range visibility and soften distant structures, cloud variants shift the illumination toward overcast conditions, and snow variants alter ground and surface appearance. The repeated temporal columns remain aligned within each sample, showing that weather transfer does not override the underlying camera-conditioned world evolution.

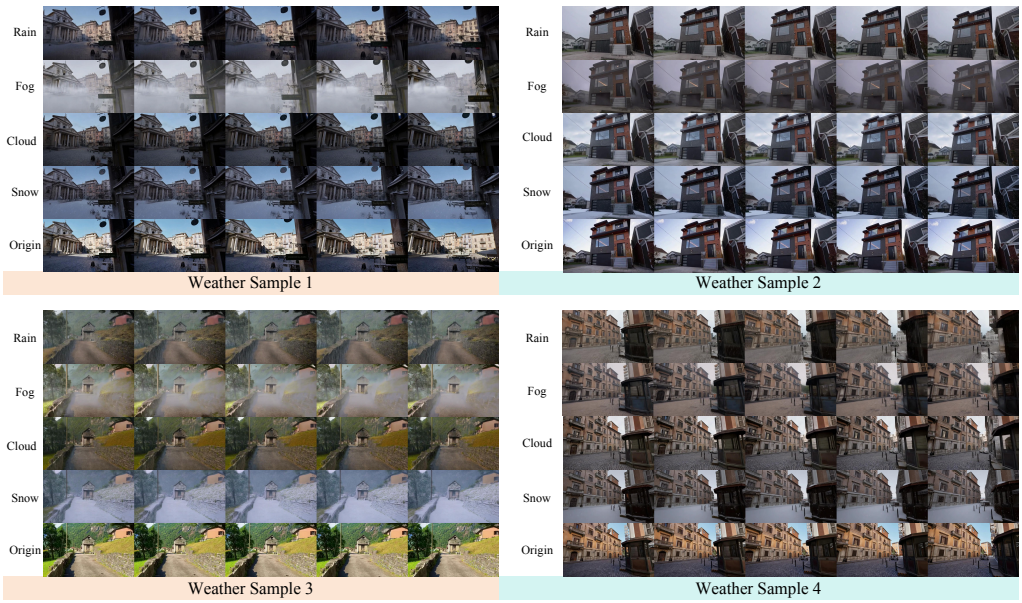
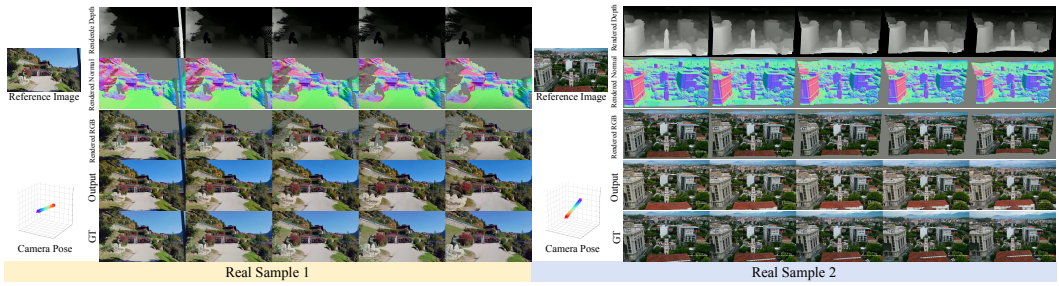


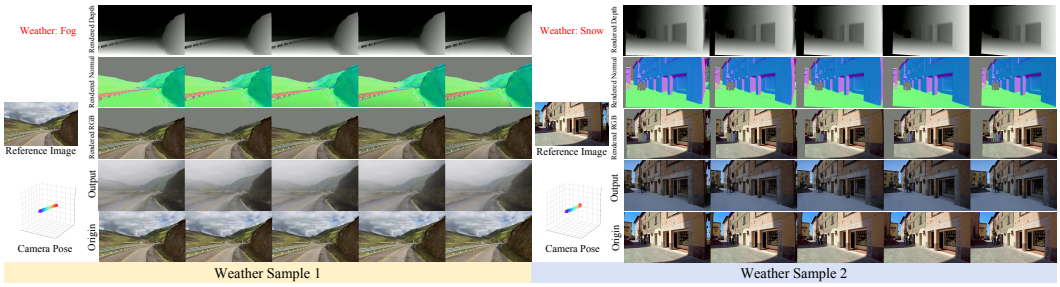
Figure 7: **Multi-weather state control.** With the same source world and camera/object controls, changing only the target weather prompt produces different weather videos.

Figure 8 exposes the intermediate representations used by our Holo-World and compares to the generated videos. In the Real examples, rendered RGB supplies the camera-conditioned background, while depth and normal buffers provide geometry-aligned conditions for scene structure. The outputs follow the rendered controls across time, preserving road boundaries, building facades, vegetation layout, and object placement without introducing unnecessary weather or global restyling. This demonstrates that our Holo-World can preserve the world as expected by the source-side controls.

The Weather examples use the same type of source-side controls but with a target weather state. In the fog case, Holo-World lowers distant contrast and visibility along the road while preserving the road curvature, guardrail structure, mountainside boundary, and camera progression. In the snow case, the model changes the street and facade appearance toward a snowy state while maintaining the original building geometry and viewpoint trajectory. The weather effects remain spatially grounded in the rendered scene structure rather than appearing as drifting texture overlays. Figures 7 and 8 show that Holo-World uses source controls to determine which world should be followed and uses the weather condition to determine how that world should be rendered. Table 7 further reports the same camera and object-control metrics on the Weather subset. This diagnostic complements the Real-subset control results in Table 2 and verifies that Holo-World still preserves camera and object control fidelity while changing the scene into a target weather state.



(a) World preservation.



(b) Weather transfer.

Figure 8: **Additional visualization of Holo-World.** Real examples visualize background-consistent generation under rendered controls, while Weather examples visualize target weather transfer under the same source-side control interface.

Table 7: **Camera and object-control results on the Weather subset.** The metrics follow the same RotErr, TransErr, and ObjMC definitions as Table 2, but are computed on Simulation and V2V weather-transfer samples.

Method	RotErr ↓	TransErr ↓	ObjMC ↓
Holo-World	1.681	1.745	6.744

A.4 Application to nuScenes Driving Scenes

To explore Holo-World in autonomous-driving scenes, we extend HoloStateData with additionally annotated video clips from the nuScenes dataset and include them in Holo-World training. These clips follow the same source-to-state annotation format, with camera trajectories, object controls, and rendered RGB, depth, and normal buffers. Figure 9 shows qualitative results for world preservation and weather transfer on the nuScenes dataset. In the preservation example, the generated sequence follows the forward camera motion while retaining the overall road layout, surrounding buildings, and leading vehicles, despite increasingly incomplete source renderings. In the transfer example, the same source scene is generated under rain, fog, cloud, and snow conditions: rain darkens the scene and introduces visible rain streaks, fog reduces distant visibility, cloud changes the sky appearance, and snow covers the road surface. Across these variants, the street layout and leading truck remain broadly recognizable, although fine vehicle and facade details can still vary. These examples illustrate the applicability of Holo-World's unified control interface to driving scenes after incorporating nuScenes training data.

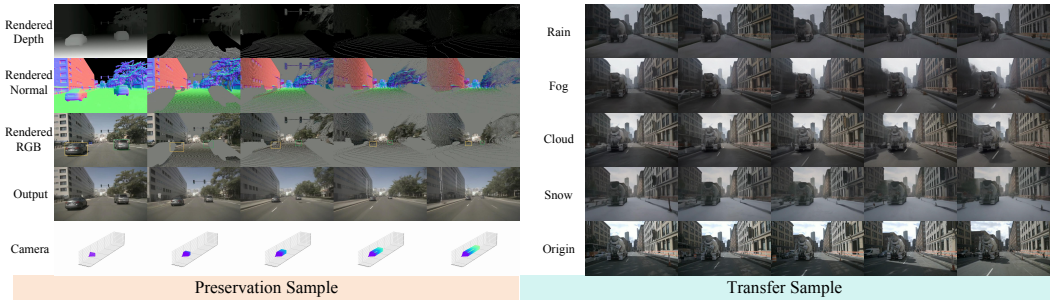


Figure 9: **Application of Holo-World to nuScenes driving scenes.** Holo-World supports world preservation (left) and weather transfer (right) under source-side camera and object controls. The original video is shown for reference in the transfer example.

A.5 HoloStateData Details

Figure 11 illustrates how HoloStateData converts a raw video clip into a source-to-state training sample. Each sample starts from a source frame and source-side controls, including rendered background RGB, geometry buffers, camera motion, and object controls. These fields specify the world that should be followed during generation. When paired weather supervision is available, the sample additionally provides a target-weather video and a weather prompt condition, which specify how the same controlled world should be rendered under a new environmental state.

The key design is to keep source controls and target supervision separated. Rendered background controls and object controls are always constructed from the source world, so target-weather pixels cannot leak into the model-facing controls. Real samples use the original video as no-edit supervision for world preservation, while paired weather samples use the target-weather video only as the supervised output for weather. This shared schema lets HoloStateData train both background-consistent generation and weather-state transfer without changing the inference interface.

Dataset split statistics. Table 8 summarizes the split design of HoloStateData. The training set contains about 15K samples organized into Real, Simulation, and V2V subsets, while the held-out benchmark contains 150 samples with 50 samples from each subset so that no-edit world preservation and target-weather generation can be evaluated separately. The benchmark is designed as a stratified diagnostic split rather than a randomly sampled mixed test pool, so each subset corresponds to a distinct source-to-state evaluation contract. The Weather training subset is formed by the Simulation and V2V subsets and covers four target weather families. Figure 10 visualizes the family-level distribution, making the balance of cloud, rain, snow, and fog supervision explicit for training.

Table 8: **HoloStateData split statistics.** The benchmark is balanced across Real, Simulation, and V2V subsets, while training-set subset counts are reported for dataset transparency.

Subset	Training Samples	Benchmark Samples
Real	7,571	50
Simulation	3,541	50
V2V	3,954	50
Total	15,066	150

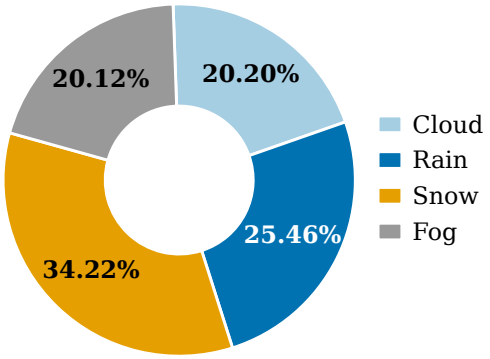


Figure 10: **Weather-family distribution in HoloStateData.** The pie chart shows the target weather-family distribution of the Weather training subset used for state-transfer supervision.

VLM text annotation. HoloStateData uses Qwen3-VL to produce factorized text conditions with low-randomness decoding (Bai et al., 2025). The annotation pipeline uses two VLM prompts rather than one generic caption. The scene prompt is generated from the source video and describes scene identity, spatial layout, visible objects, object relations, temporal evolution, and camera motion. It explicitly excludes weather, global illumination, imagined content, and hidden causes, so the source-world description does not bind the scene to a particular environmental state. For paired weather samples, a separate weather prompt is generated from the target-weather video and restricted to a compact weather taxonomy. Real samples keep the weather field empty. This factorized annotation keeps scene identity, camera/object evolution, and weather-state transfer in separate text channels.

```
You are a video caption generation specialist whose goal is to create high-quality,
↪  detailed English captions for the user's input video by closely observing its
↪  content. The captions are intended for video world model training data: prioritize
↪  scene layout, how scene state evolves over time, and how people and objects interact
↪  with the environment and each other, while still covering people and camera work when
↪  visible. Your task is to describe the video comprehensively and expressively while
↪  maintaining authenticity and naturalness. Integrate visual and temporal elements
↪  strictly adhering to the formatting of the examples provided.

Task Requirements:
1. Describe the main theme and setting (such as location and spatial layout of the
↪  scene). Do not describe weather or lighting here; omit those entirely (they are
↪  covered by a separate annotation prompt).
2. Describe the environment and scene-anchored cues (terrain, architecture, furniture,
↪  obstacles, open space, notable static props), then key subjects (appearance,
↪  clothing, expression, quantity, ethnicity, posture, etc.), spatial relationships, and
↪  camera movements;
3. Describe relationships and interactions between people, objects, and the environment
↪  (positioning, contact, support, occlusion, entry and exit, use of tools or furniture,
↪  etc.);
4. Describe any character present and their emotions or activities (such as expressions,
↪  postures, etc.), in relation to nearby objects and the scene;
5. The caption should reflect the style of the video (e.g., cinematic, anime,
↪  documentary, vlog, vintage film, etc.);
6. Describe the content in chronological order, including scene-level changes (rearranged
↪  or moved objects, reconfiguration of space) and how characters or objects move;
↪  describe how the camera changes perspective. Use accurate verbs to describe movement;
7. Describe background and environmental details (such as architecture, natural scenery,
↪  materials and clutter, etc.) without weather or lighting;
8. Describe camera motion control (e.g., zoom in, zoom out, push in, pull out, pan right,
↪  pan left, truck right, truck left, tilt up, tilt down, pedestal up, pedestal down,
↪  arc shot,  tracking shot, static shot, and handheld shot, etc.);
9. Do not describe imagined content. Only describe what can be determined from the video.
↪  Avoid listing things. Do not use abstract concepts (love, hate, justice, infinity,
↪  joy) as subjects. Use concrete nouns (human, cup, dog, planet, headphones) for more
↪  accurate results. Use verbs to describe movement and state changes. Write in plain,
↪  conversational language. Start with the scene or central situation when it is clear
↪  (often a place or situational phrase); otherwise start with the clearest main
↪  subject. Without "\n", subheading and title.
10. Control the caption length to around 80-150 words;
```

(1) Scene annotation prompt. The prompt extracts source-scene content and motion while excluding weather from the scene text.

```
You are a weather-only caption specialist for video weather editing data. From the input
↪  video (or the specified target), produce one short English line that states only the
↪  weather condition using the fixed phrase inventory below. Do not describe scene layout,
↪  objects, people, lighting nuances, camera paths, or style--except where the chosen
↪  output template already mentions preserving scene or camera.

Weather taxonomy (pick exactly one family per output; never combine families):

1. Cloud (cloud-only) -- use one phrase only:
    - `a few clouds` / `cloudy` / `overcast`

2. Rain (precipitation and/or wet ground, independent) -- choose what is visibly present;
↪  do not force a pair. Use only phrases from the lists below:
    - Rain (precipitation): `light rain` / `rain` / `heavy rain`
    - Puddle / rain ground: `slightly wet rain ground` / `rain ground` / `flooded rain
    ↪  ground`
    - Compose one line: one matching phrase or, if both rain and wet ground are clearly
    ↪  present, two phrases in order `{rain phrase} and {puddle phrase}`. If only rain or
    ↪  only wet ground is visible, output that single phrase (no invented second tier).

3. Snow (falling snow and/or snow on ground, independent) -- same independence as rain: do
↪  not require falling snow and ground snow together. Use only:
    - Falling snow: `light snow` / `snow` / `heavy snow`
    - Snow-covered ground: `light snow-covered ground` / `snow ground` / `deep snow ground`
    - One phrase, or both as `{falling-snow phrase} and {ground-snow phrase}` when both are
    ↪  clearly visible; otherwise a single phrase matching what you see.

4. Fog (fog-only) -- use one phrase only:
    - `light fog` / `fog` / `thick fog`

Disambiguation rules:
- Do not add cloud or fog when the family is Rain or Snow, and do not add rain or snow when
↪  the family is Cloud or Fog. This keeps labels disjoint for training (cloudy / rainy /
↪  snowy / foggy are separate regimes).
- Do not mix rain and snow in one line; pick one dominant family. If both are strongly
↪  visible, prefer Rain for this dataset unless snow clearly dominates the frame.
- Use only the strings above for weather content--no synonyms (no "drizzle", "mist",
↪  "partly cloudy", etc.).

Output templates (choose exactly one per sample; do not output both):

- Template A: `Edit the video to show {phrases}. Keep the scene structure unchanged and
↪  only change the weather realistically.`

- Template B: `{phrases}. Realistic weather, preserve scene layout and camera motion.`

Replace `{phrases}` with the composed weather content for the chosen family: a single cloud
↪  or fog phrase; or for rain/snow, one or two inventory phrases joined by `and` only when
↪  both tiers are justified (rain before puddle; falling snow before ground snow).

Formatting:
- One single line of English; no line breaks, no subheadings, no quotes around the full
↪  line.
- Keep it short; only the template sentence(s) above.

Directly output the single English line and nothing else.
```

(2) Weather annotation prompt. The prompt maps target-weather evidence to a compact weather-state phrase without adding scene, object, or camera descriptions.

Automatic construction. The construction pipeline turns the same sample into model-facing controls. For the V2V subset, SpatialVID-HQ videos are used as source clips. Our weather-transfer model is fine-tuned from Wan2.1-Fun-Control-14B on open-source paired simulated weather videos (Lin et al., 2025) rendered with Unreal Engine, and is used together with proprietary video editing models to generate candidate target-weather clips for SpatialVID-HQ sources. We retain generated pairs only when the target weather is clearly visible, the source scene layout is preserved, and no severe temporal artifacts or object-level distortions are observed. These checks are applied before the retained V2V samples enter the same source-to-state construction pipeline as the Real and Simulation subsets. Depth Anything and UniDepth provide monocular depth priors, MegaSaM estimates camera parameters, and MoGe supplies dense depth and normal anchors for rendering source-side background RGB and geometry-buffer controls along the camera trajectory (Yang et al., 2024; Piccinelli et al., 2024; Li et al., 2025d; Teed & Deng, 2021; Wang et al., 2025b). For ob-

ject control, Grounding DINO detects first-frame object candidates and SAM-2 propagates masks through the clip, after which each mask is converted into a tight per-frame bounding box (Liu et al., 2024; Ravi et al., 2025). The first rendered frame is pinned to the observed source frame and its geometry estimates to reduce conditioning-frame drift. For paired weather samples, the target-weather video is used only as the supervised output and never as a source for rendered controls. This construction rule matches the model design: world controls define the scene scaffold to preserve, while weather text and paired targets supervise the state in which that scaffold should be rendered.

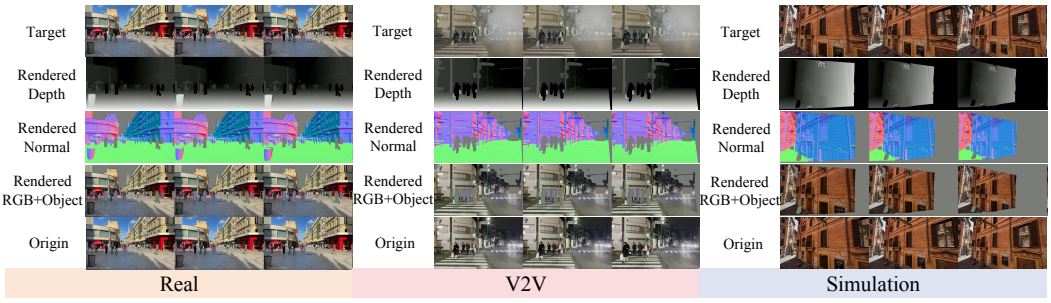


Figure 11: HoloStateData example. The visualization shows how one video record is converted into the source first frame, rendered world controls, object controls, and weather-state annotations used by the unified source-to-state training interface.

A.6 Evaluation of Holo-World

A.6.1 Metric Overview

This appendix gives the formulas, prompts, evaluator caveats, and user-study protocol for VBench-I2V, camera/object-control errors, Weather Alignment, VLM Evaluation, and User Study. A valid Weather-subset run fixes the split, generated-video manifest, evaluator prompts, frame sampling, and human-study design before scoring any method. The Real subset uses standard video, camera, object, and rendered-background metrics; the Weather subset uses weather-specific metrics for target-weather semantics, generation quality, and human preference.

Weather Alignment ignores scene identity, object identity, camera motion, and general image quality except when they make the weather evidence unreadable. VLM Evaluation is broader: it asks whether the weather background, weather dynamics, and video quality are plausible, but it still does not score camera-control or object-control accuracy. User Study is the complementary human signal and is reported as two separate pairwise win rates rather than a single mixed score. For the weather metrics via VLM, we adopted GPT model to evaluate the Weather Alignment and VLM Evaluation.

Each Weather-subset item contains I_0, a target weather prompt c_{state}, generated video $\hat{V}$, and optional source/control context for human raters. Automatic Weather Alignment and VLM Evaluation use only c_{state} and $\hat{V}$; the input context is reserved for provenance and human preference. Only Simulation and V2V subset samples participate because Real samples have no target weather edit. The three Weather-subset metrics are therefore never averaged with Real-subset scores or collapsed into a mixed benchmark number.

A.6.2 VBench-I2V Metrics

We evaluate image-to-video generation quality on the Real subset using the official VBench-I2V metrics (Huang et al., 2025b). For each generated clip, the evaluator receives the conditioning image and the generated video, then reports eight dimensions used in the main table: imaging quality, aesthetic quality, dynamic degree, motion smoothness, background consistency, subject consistency, I2V subject consistency, and I2V background consistency, with camera motion excluded. Following VBench-I2V, the Overall Score is a weighted aggregation of these eight normalized dimension scores. Let $\{s_k\}_{k=1}^{8}$ denote the eight raw VBench-I2V scores over the Real subset. We compute

$$S_{\text{Overall}} = \frac{\sum_{k=1}^{8} w_k \, \text{Norm}_k(s_k)}{\sum_{k=1}^{8} w_k}. \tag{16}$$

Here, w_k denotes the effective per-dimension weight that preserves the within-group weighting and the equal weighting of the video-quality and I2V consistency groups after excluding camera motion. The main table reports the averaged VBench-I2V scores over the Real subset, with all values shown on a percentage scale.

A.6.3 Camera and Object-Control Metrics

We evaluate camera and object-control fidelity on the Real subset by comparing trajectories recovered from generated videos with the target trajectories. For each generated video, the evaluation methods estimate a camera trajectory and dynamic object trajectories using the same geometry annotation.

For camera control, let $\{R^t_{\text{target}}, T^t_{\text{target}}\}_{t=1}^T$ represent the target camera rotation and translation parameters and $\{R^t_{\text{gen}}, T^t_{\text{gen}}\}_{t=1}^T$ represent the camera trajectory estimated from the generated video. We calculate rotation error (RotErr) by the average geodesic distance on $\mathrm{SO}(3)$:

$$\text{RotErr} = \frac{1}{T}\sum_{t=1}^{T} \arccos\left(\frac{\text{tr}(R^t_{\text{gen}}(R^t_{\text{target}})^\top) - 1}{2}\right). \tag{17}$$

Translation error (TransErr) is the average Euclidean distance between the target and generated camera translations:

$$\text{TransErr} = \frac{1}{T}\sum_{t=1}^{T} \left\|T^t_{\text{gen}} - T^t_{\text{target}}\right\|_2. \tag{18}$$

For object motion control, let N be the number of target controlled objects. For target object o and predicted object k, their trajectory distance is

$$d(o,k) = \frac{1}{T}\sum_{t=1}^{T} \left\|\hat{\mu}^t_k - \mu^t_o\right\|_2, \tag{19}$$

where μ^t_o and $\hat{\mu}^t_k$ are the target and generated 3D object centers at frame t. Since object identities are not known in generated videos, we match target and predicted trajectories with the Hungarian algorithm. Unmatched target objects are assigned a fixed penalty λ. The object motion control score is

$$\text{ObjMC} = \frac{1}{N}\sum_{o=1}^{N} d_o, \qquad d_o = \begin{cases} d(o,k), & \text{if target object } o \text{ is matched to prediction } k, \\ \lambda, & \text{if } o \text{ is unmatched.} \end{cases} \tag{20}$$

It is noted that we only calculate the special object class set "person . human . rider . bicycle . motorcycle . scooter . car . truck . bus . animal" via Grounded-SAM2, which is the same as the dataset annotation setting.

A.6.4 Weather Alignment

Weather Alignment is a binary weather-semantics score. Its input is a generated video $\hat{V}$, the target weather prompt c_{state}, and a frozen frame/clip sampler. Its output is one aligned/not-aligned decision per sample. For sample i,

$$a_i = E_{\text{wa}}(\hat{V}^{(i)}, c^{(i)}_{\text{state}}), \qquad a_i \in \{0,1\}, \tag{21}$$

where $a_i = 1$ denotes alignment. We report the pass rate

$$S_{\text{WA}} = \frac{100}{N}\sum_{i=1}^{N} a_i. \tag{22}$$

The evaluator prompt is intentionally short and exclusionary: it asks whether the target core weather family is clearly visible and forbids using scene identity, object identity, camera motion, composition, or general image quality as scoring criteria.

```
You are evaluating only whether a generated video aligns with the target weather prompt.

Target weather prompt:
{c_state}

Only consider the core weather meaning in the prompt. The allowed core weather
types are: cloud, rain, snow, fog.

Generated video:
{video}

Question:
Does the generated video align with the target weather prompt?

Judge only whether the core weather requested by the prompt is visible:
- cloud: cloudy or overcast sky;
- rain: visible falling rain;
- snow: visible falling snow;
- fog: visible fog, haze, or reduced visibility caused by fog.

Do not judge scene identity, object identity, camera motion, composition,
or general video quality unless they make the weather condition impossible to judge.
Do not consider puddles, wet ground, snow accumulation, snow-covered ground,
weather intensity words, or any other secondary weather effects.

Return a JSON object:
{
  "aligned": true or false,
  "reason": "one short sentence explaining the weather evidence"
}

Decision rule:
aligned = true only if the core weather requested by the prompt is clearly visible.
aligned = false if the target weather is absent, ambiguous, too weak,
or belongs to another category.
```

(3) Weather alignment evaluator prompt. The prompt measures only whether the target core weather family is visible and excludes scene, object, camera, and general quality factors.

The Weather-subset table reports the aligned-sample percentage over the 100 Simulation/V2V samples. Intensity words such as light rain, heavy snow, or thick fog do not change the criterion: WA only requires the core weather family to be visible.

Evaluator validation and bias. If Qwen3-VL is used for both annotation and evaluation, WA may inherit vocabulary and calibration bias. We therefore validate it on GPT model, distinguishing labeled VLMs.

A.6.5 VLM EVALUATION

VLM Evaluation scores weather generation quality on a 0–100 scale. The evaluator receives the target weather prompt and generated video, but not the target-weather video. It focuses on three subdimensions: weather background editing, weather dynamics, and video quality. This score is distinct from Weather Alignment: a video can pass Weather Alignment by showing visible rain or snow, yet receive a low VLM Evaluation score if the weather background is unchanged, particle motion is unnatural, or the video flickers. For sample i,

$$q_i = E_{\mathrm{vlm}}(\hat{V}^{(i)}, c_{\mathrm{state}}^{(i)}), \qquad q_i \in [0, 100], \tag{23}$$

and the reported score is

$$S_{\mathrm{VLM}} = \frac{1}{N} \sum_{i=1}^{N} q_i. \tag{24}$$

```
You are evaluating the quality of a generated weather video for a research benchmark.

Target weather prompt:
{c_state}

Generated video:
{video}

Please evaluate only the generated weather result. Give one overall score from 0 to 100.
Focus on:
1. Weather Background: whether the sky, ground, road, buildings, visibility,
   surface materials, reflections, snow coverage, wetness, fog atmosphere,
   or other background cues are plausibly edited for the target weather.
2. Weather Dynamics: whether dynamic weather elements such as rain, snow,
   fog, mist, or moving atmospheric effects appear natural, temporally coherent,
   and consistent with the target weather.

Do not reward a method merely for keeping the background unchanged.
A good result should edit the weather background when the target weather requires it.
Do not evaluate camera-control accuracy or object-control accuracy.

Use the following rough scoring scale:
- 0-20: unusable result, no meaningful target weather generation, or weather is almost
↪  entirely absent.
- 21-40: very weak weather generation; only minor or partial cues such as slight wetness,
↪  faint haze, or limited surface change.
- 41-60: recognizable target weather, but background editing or weather dynamics are
↪  incomplete, weak, unstable, or unnatural.
- 61-80: good weather generation; background editing and weather dynamics are mostly
↪  plausible, with some visible artifacts or limitations.
- 81-100: excellent weather generation with natural background editing, convincing
↪  weather dynamics, coherent atmosphere, and high video quality.

Return a JSON object:
{
  "score": an integer from 0 to 100,
  "reason": "one concise explanation of the main strengths and weaknesses"
}

Scoring rule:
0 = unusable or no meaningful weather generation
100 = excellent weather generation with natural background editing,
      convincing weather dynamics, and high video quality
```

(4) VLM evaluation prompt. The prompt scores weather background editing, weather dynamics, and video quality without rewarding unchanged backgrounds or measuring camera and object control.

The main score S_{VLM} is the mean of Eq. 24. The evaluation manifest may store textual reasons for failure analysis, but paper tables report only the scalar mean. The prompt discourages rewarding unchanged backgrounds because the Weather subset evaluates weather-state generation, not background copying.

A.6.6 Human-Metric Alignment

To validate whether automatic weather metrics reflect human judgments, we compute the Pearson correlation coefficient (r) between human ratings and automatic scores on shared Weather-subset samples. Five annotators score each of the 100 Weather-subset samples on a five scales along weather correctness, weather dynamics, and geometry preservation under the generated weather, and we average the annotator scores before computing correlations. Table 9 reports the alignment for Weather Alignment and VLM Evaluation, where a higher correlation indicates better agreement with human assessments of weather visibility and overall weather-generation quality.

Table 9: **Human-metric alignment.** Pearson correlation coefficient (r) between automatic weather metrics and human ratings on the Weather subset.

Metric	Weather Alignment	VLM Evaluation
Pearson (r)	0.78	0.66

A.6.7 User Study

Human preference complements the automatic metrics. Evaluators see the target weather prompt, optional input/source context, and two randomly ordered generated videos with hidden method names, then choose the video with more natural weather generation and weather-

background editing. They are not asked to judge camera or object accuracy. For baseline $b \in \{\text{Cosmos-Transfer2.5}, \text{Wan2.7-Edit}\}$, let $u_i^b \in \{0, 1\}$ indicate that the majority vote prefers Holo-World over b on sample i. We report

$$S_{\text{User}}^b = \frac{100}{N} \sum_{i=1}^{N} u_i^b. \tag{25}$$

The main table keeps Holo-World vs. Cosmos-Transfer2.5 and Holo-World vs. Wan2.7-Edit separate.

A.6.8 Background Preservation Metrics

The Background columns in Tables 4 and 5 use rendered-background preservation metrics on the Real subset. These metrics evaluate source-supported background regions from rendered RGB controls; they are not Weather-subset scores.

For frame t, let $B_t \in [0, 1]^{H \times W \times 3}$ denote the rendered reference background RGB frame and let $\hat{I}_t \in [0, 1]^{H \times W \times 3}$ denote the generated frame. The renderer may contain invalid placeholder regions. We define an empty-color set $\mathcal{E}$ containing pixels close to the black placeholder and the gray placeholder used by the rendered controls. The valid background mask is the intersection of non-empty regions in the reference and generated frames:

$$M_t = \mathbf{1}[B_t \notin \mathcal{E}] \wedge \mathbf{1}\left[\hat{I}_t \notin \mathcal{E}\right]. \tag{26}$$

The same mask is applied to both images, so the metric reflects disagreement on comparable, source-supported background pixels rather than differences in unrendered regions.

LPIPS is sensitive to convolutional context around mask boundaries. We therefore composite both frames onto the same neutral color c before computing image-level metrics:

$$B_t^M = M_t \odot B_t + (1 - M_t) \odot c, \tag{27}$$

$$\hat{I}_t^M = M_t \odot \hat{I}_t + (1 - M_t) \odot c. \tag{28}$$

For the evaluated frame set $\mathcal{T}$, the reported background metrics are

$$\text{Background-PSNR} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \text{PSNR}(B_t^M, \hat{I}_t^M), \tag{29}$$

$$\text{Background-SSIM} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \text{SSIM}(B_t^M, \hat{I}_t^M), \tag{30}$$

$$\text{Background-LPIPS} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \text{LPIPS}(B_t^M, \hat{I}_t^M). \tag{31}$$

Higher Background PSNR/SSIM and lower Background LPIPS indicate better source-supported background preservation. The mask excludes unrendered regions and placeholder-color artifacts, making the metric better aligned with the Real-subset no-edit contract than whole-frame reconstruction.