Coarse-to-Real: Generative Rendering for Populated Dynamic Scenes

Gonzalo Gómez-Nogales
Universidad Rey Juan Carlos
Móstoles, Spain

Yicong Hong
Adobe Research
San Jose, USA

Chongjian Ge
Adobe Research
San Jose, USA

Peiye Zhuang
Roblox
San Mateo, USA

Marc Comino-Trinidad
Universidad Rey Juan Carlos
Móstoles, Spain

Dan Casas[†]
Universidad Rey Juan Carlos
Móstoles, Spain

Yi Zhou
Adobe Research
San Jose, USA



Figure 1: **C2R (Coarse-to-Real):** transforms a coarse 3D control video from different types (left col.) like coarse block-like low-poly characters (top), humanoid coarse models (middle) and game videos (bottom), into diverse scenes via text prompts, varying texture detail, lighting, weather, location, dynamics and clothing styles while maintaining consistent camera motion and human trajectories.

Abstract

Traditional rendering pipelines rely on complex assets, accurate materials and lighting, and substantial computational resources to produce realistic imagery, yet they still face challenges in scalability and realism for populated dynamic scenes. We present C2R (Coarse-to-Real), a generative rendering framework that synthesizes real-style urban crowd videos from coarse 3D simulations. Our approach uses coarse 3D renderings to explicitly control scene layout, camera motion, and human trajectories, while a learned neural renderer generates realistic appearance, lighting, and fine-scale dynamics guided by text prompts. To overcome the lack of paired training data between coarse simulations and real videos, we adopt a two-stage synthetic-real domain-hedging strategy that first learns a strong generative prior from large-scale real footage, then introduces controllability by using a small amount of paired synthetic coarse–fine data to anchor shared implicit spatio-temporal features across domains. The resulting system supports coarse-to-fine control, generalizes across diverse CG and game inputs, and produces temporally consistent, controllable, and realistic urban scene videos from minimal 3D input.

[†]Work done prior to joining Amazon.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Rendering populated realistic urban scenes with dynamic camera trajectories and complex group motion remains expensive and difficult in traditional computer graphics (CG) pipelines. Achieving realism typically requires high-quality assets, detailed materials, accurate lighting, and carefully tuned simulations, leading to substantial production cost and memory overhead, yet often still exhibiting a gap from real-world appearance. Recent advances in generative text- and image-to-video models offer an alternative Ho et al. [2022], Chen et al. [2023a], Yang et al. [2025], Ruan et al. [2023], Singer et al. [2023], but existing methods struggle with scenes involving multiple interacting humans, long-term temporal consistency, and coherent camera motion. Purely text-based control is insufficient for precisely specifying scene structure and dynamics, and image-conditioned approaches Geng et al. [2025], like Sora, remain limited to simple and similar camera positions for scenes involving complex crowd dynamics, as shown in the Appendix Section A.2. Video-to-video models Wang et al. [2023], NVIDIA et al. [2025], Cheng et al. [2024], Li et al. [2024] combined with ControlNet Zhang et al. [2023] offer promising and more controllable alternatives, but they usually use strong geometric control signals that over-constrain the generation and the expressivity of the output video.

A key observation motivating this work is that many challenges related to dynamic video generation are trivial in 3D, while many of the most expensive aspects of 3D rendering are naturally handled by data-driven models. Camera motion, spatial structure, and temporal consistency are explicit and stable in coarse 3D scenarios, whereas detailed geometry, textures, materials, lighting, and visual diversity are costly to author and render. This complementary relationship motivates a hybrid paradigm, where coarse 3D simulation provides structural and dynamic control, and a learned generative model functions as a renderer that synthesizes realistic appearance.

We introduce **C2R (Coarse-to-Real)**, a generative rendering framework that produces real-style populated urban videos from coarse 3D inputs. C2R takes temporally consistent coarse renderings encoding scene layout, camera trajectories, and human motion, and enriches them with realistic textures, lighting, and fine-scale dynamics learned from real-world video data. The framework offers flexible control, including inpainting clothing, hair, buildings, and environmental details beyond the input structure, adjusting the strength of generative rendering, and supporting coarse-to-fine inputs. Importantly, C2R is agnostic to specific human templates and asset formats, allowing it to generalize across a wide range of CG, game, and animation scenes.

A central technical challenge is the absence of paired training data between coarse CG simulations and real-world videos. To address this challenge, we propose a synthetic-real domain-hedging strategy based on a two-stage learning framework. In the first stage, the model is trained on large-scale unpaired real-world videos to learn a strong photorealistic generative prior. In the second stage, controllability is introduced by grounding this prior using *implicit spatio-temporal features* extracted from both real and synthetic inputs, enabling the model to interpret structural cues in a shared feature space across domains despite the absence of explicit pairing, and to adaptively hallucinate content based on the level of detail present in the input. A small proportion of synthetic paired coarse–fine data anchors the correspondence between coarse structure and realistic appearance, while the dominance of real data prevents contamination by CG-specific artifacts. The transferable contribution is this training strategy rather than any particular conditioning block. In Sec. 4.3 we show that the same recipe also moves other controllable backbones out of degenerate operating points on coarse inputs. To support diverse generation conditions, we curate and annotate footage from five continents, covering a wide range of cities, weather conditions, lighting, and clothing styles. We evaluate different latent feature insertion strategies, compare quantitatively against 16 baseline configurations, including baselines fine-tuned on our data, and demonstrate that C2R produces temporally consistent, controllable, and realistic urban scene renderings from minimal 3D simulation.

2 Related Works

We review research relevant to the high-quality synthesis of populated urban videos, from the perspective of traditional computer graphics pipelines (Sec.2.1), video and world models (Sec.2.2) and controllable video generation techniques (Sec.2.3). Our discussion emphasizes the unique challenges of maintaining structural control and temporal consistency in dynamic scenes with complex crowd interactions.

2.1 Traditional CG for Dynamic Populated Scenes

Rendering populated urban environments with dynamic camera trajectories and complex group motion is among the most resource-intensive tasks in traditional computer graphics pipelines. Achieving

high-fidelity realism necessitates extensive libraries of quality assets, including diverse 3D human meshes, detailed architecture, and physically based materials. Generating these components typically requires expert manual modeling or costly 3D scanning Loper et al. [2015], Anguelov et al. [2005]. As scene complexity scales, the memory overhead for high-resolution textures and geometry strains computational resources and compromises rendering efficiency. Beyond static assets, animating large crowds requires simulating intricate group behaviors and social interactions Helbing and Molnar [1995]. While traditional techniques like skeletal animation and motion graphs Kovar et al. [2002] provide stability, they often struggle to produce naturalistic motion and vivid dynamics. Consequently, significant visual discrepancies compared to real-world appearances persist—particularly for large-scale crowds with diverse clothing and hair patterns—rendering the scalable authoring of realistic urban scenes practically challenging. Unlike these resource-intensive pipelines, our C2R framework leverages coarse 3D simulations as lightweight structural proxies, delegating the synthesis of photorealistic details and vivid dynamics to a data-driven generative renderer.

2.2 Video and World Models

Recent advances in text-to-video Ho et al. [2022], Chen et al. [2023a], Yang et al. [2025], Ruan et al. [2023], Wan et al. [2025], Singer et al. [2023] and image-to-video Esser et al. [2023], Chen et al. [2023b] generation offer a compelling alternative by synthesizing realistic imagery directly from data. However, existing generative video models struggle to reliably produce scenes with multiple interacting humans, long-term temporal consistency, or coherent camera motion. In Figure 10 in the Appendix Section A.2, SORA and WAN 2.1 Wan et al. [2025] show limited controllability over human motion and camera trajectories, and tend to generate similar viewing angles across populated urban scenes. Moreover, purely text-based control is insufficient for precisely specifying scene structure, camera trajectories, and crowd dynamics, while image-conditioned approaches inherit the limitations of the input frame and offer limited control over motion and layout. As a result, current video generation systems fall short as practical tools for authoring structured, dynamic urban scenes.

To provide the structural grounding required for such urban scenes, a parallel line of research has emerged around world models. Unlike purely pixel-based generators, these models aim to leverage explicit 3D priors Kerbl et al. [2023], Wang et al. [2024] to build consistent traversable scenes. For instance, WorldGen Wang et al. [2025] utilizes procedural generation to ensure global layout stability and precise camera control. However, while these models excel at maintaining static structure, they often struggle to capture the vivid, stochastic dynamics of populated urban crowds.

Unlike these approaches, C2R does not attempt to reconstruct a 3D world or rely on purely procedural logic. Instead, we use coarse 3D simulations as structural proxies and delegate the computationally expensive rendering of photorealistic textures, lighting, and fine-grained crowd dynamics to a learned generative model. This hybrid formulation allows C2R to bypass the "sim-to-real" gap that persists in traditional CG and the lack of structural control in monolithic video models.

2.3 Controllable Video Generation

To bridge the gap between generative realism and structural precision, several lines of work have explored explicit conditioning for video synthesis: (1) One of the strategies focuses on sparse motion and camera guidance. Methods such as CameraCtrl He et al. [2025], GEN3C Ren et al. [2025], and Wan-Move Chu et al. [2025] incorporate explicit camera parameters or trajectories to guide the denoising process. Similarly, motion-driven approaches like Motion Prompting Geng et al. [2025] and MCDiff Chen et al. [2023b] utilize user-defined strokes or points to specify movement. While effective for single-subject scenes, these methods often struggle with the combinatorial complexity of crowded environments, where multiple characters interact with independent and varying velocities. Furthermore, sparse signals often fail to ground the global scene layout as robustly as a 3D proxy. (2) Another strategy involves dense structural conditioning, often referred to video-to-video (vid2vid) synthesis. Existing frameworks Esser et al. [2023], Wang et al. [2018], NVIDIA et al. [2025] use per-frame geometric cues, e.g., depth maps, Canny edges, or semantic layouts, to transform input sequences. However, these strong geometric signals often over-constrain the model expressivity, forcing the output to strictly follow the input geometry and preventing the generative renderer from adding "vivid dynamics".

Our C2R takes a different path by leveraging an implicit spatial-temporal feature representation. Unlike previous methods that rely on either sparse trajectories or rigid dense maps, C2R utilizes coarse 3D representations as lightweight structural proxies. This allows our framework to maintain the global stability and intent of the 3D scene while giving the generative model the freedom to

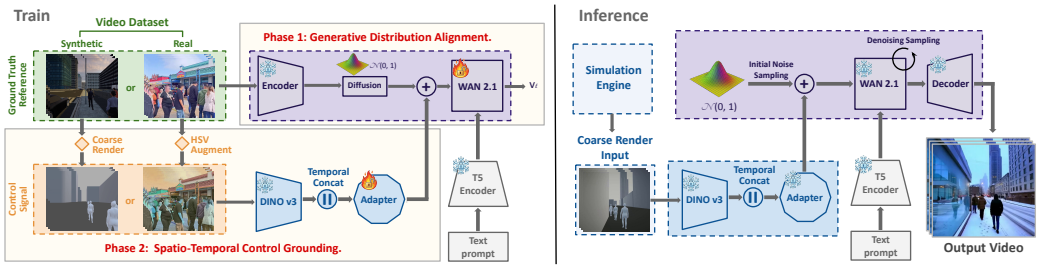

Figure 2: **Overview of C2R** A two-stage framework decouples photorealistic prior learning from structural control: the diffusion backbone is first adapted to real videos, then grounded through a synthetic-real domain-hedging stage that uses implicit spatio-temporal features shared across real and synthetic data. At inference, a coarse render and text prompt guide denoising to synthesize realistic videos following input motion and layout.

generate high-fidelity textures, physically plausible lighting, and fine-grained crowd dynamics learned from real-world footage.

3 Method

3.1 Overview and Design Choices

Our goal is to generate vivid and real-styled videos that follow the camera motion and scene dynamics of coarse 3D or game-engine renders. This setting introduces three challenges: (1) coarse renders provide reliable structure but lack realistic appearance; (2) paired supervision between coarse 3D inputs and photorealistic targets is only available for CG data, while real videos are abundant but lack aligned control signals, requiring a bridge between real and synthetic inputs; and (3) explicit structural controls (e.g., depth, edges, poses) over-constrain generation by entangling appearance with structure, often leading to *feature leakage*, where the model bypasses synthesis and reconstructs the control input.

To address these challenges, we decouple photorealistic prior learning from structural controllability using a two-stage training strategy. We first learn a high-fidelity generative prior from large-scale real videos without structural conditioning. We then introduce controllability through a domain-hedging stage, where *implicit spatio-temporal features* extracted from both real and synthetic inputs are mapped into a shared feature space via a lightweight adapter. This design yields adaptive behavior: coarse inputs allow flexible synthesis, while richer inputs naturally enforce stronger adherence to motion and layout.

3.2 Architecture Preliminaries

We build on WAN 2.1 Wan et al. [2025] and denote its VAE encoder and decoder by E and D, and its diffusion backbone (Diffusion Transformer) by DiT with parameters θ. Text conditioning is provided by a pretrained T5-XXL encoder $\mathrm{T5}$ Raffel et al. [2020] that maps a prompt $\mathbf{c}$ to an embedding $\mathbf{e}_{\text{text}} = \mathrm{T5}(\mathbf{c})$. For control signals, we extract patchwise features using a pretrained DINOv3 ViT Siméoni et al. [2025], and train an adapter A that projects these features into the diffusion latent space.

3.3 Stage I: Generative Distribution Alignment

Motivation. Coarse control signals alone cannot teach photorealistic appearance. We therefore first adapt the diffusion backbone to the target "realistic urban video" distribution so that it learns lighting, materials, and natural motion before being asked to follow coarse structure.

Training. In this stage, we fine-tune only the diffusion backbone θ on real-world videos, while freezing the VAE E, D and the text encoder. Given a real video $\mathbf{x}$ and its corresponding prompt embedding $\mathbf{e}_{\text{text}}$, we first encode the video to latent representation:

$$\mathbf{z}_0 = E(\mathbf{x}). \tag{1}$$

Then, following the Flow Matching formulation applied in Wan 2.1, we sample a timestep $t \in [0, T]$ and apply the standard diffusion process by adding Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ to obtain the noisy

latents $\mathbf{z}_t = (1-t)\mathbf{z}_0 + t\epsilon$. The model is trained to predict the velocity field:

$$\mathbf{v}_t = \mathrm{DiT}_\theta(\mathbf{z}_t, t, \mathbf{e}_{\text{text}}), \tag{2}$$

where the objective is defined as:

$$\boldsymbol{L}_{\text{FM}} = \mathbb{E}_{\mathbf{z}_0, t, \mathbf{e}_{\text{text}}}\left[\left\|\mathrm{DiT}_\theta(\mathbf{z}_t, t, \mathbf{e}_{\text{text}}) - (\epsilon - \mathbf{z}_0)\right\|_2^2\right]. \tag{3}$$

At inference, we can predict the denoised latents $\hat{\mathbf{z}}_0$ through an iterative sampling process, and the final clean video results can be obtained by the vae decoder via $\hat{\mathbf{x}} = D(\hat{\mathbf{z}}_0)$.

3.4 Stage II: Spatio-Temporal Control Grounding

Motivation. After Stage I, the model generates realistic videos but lacks controllability. Stage II teaches the model to interpret coarse structure *without* forcing an explicit signal (e.g., depth/edges) as in ControlNet Zhang et al. [2023] that would rigidly constrain silhouettes. Instead, we use *implicit* features from a self-supervised encoder that preserve layout and motion while being less tied to pixel appearance.

Training. We freeze the tuned diffusion backbone and train only the control pathway, i.e., the adapter A (and any lightweight temporal aggregation described below). Let $\mathbf{x}_{\text{ctrl}}$ be the control-branch input. We extract features frame-by-frame:

$$\mathbf{f}_i = \mathrm{DINO}(\mathbf{x}_{\text{ctrl}}^{(i)}), \quad i = 1, \ldots, N, \tag{4}$$

and construct a spatio-temporal representation by temporal concatenation:

$$\mathbf{f}_{1:N} = \mathrm{Concat}_i(\mathbf{f}_1, \ldots, \mathbf{f}_N). \tag{5}$$

The adapter projects these features to a latent guidance tensor:

$$\hat{\mathbf{z}}_{\text{ctrl}} = A(\mathbf{f}_{1:N}). \tag{6}$$

We inject the control guidance into the diffusion latent by element-wise addition:

$$\hat{\mathbf{z}}_{\text{t}} = \mathbf{z}_{\text{t}} + \hat{\mathbf{z}}_{\text{ctrl}}. \tag{7}$$

Our experiments show that simple feature addition matches or outperforms multi-head cross-attention for integrating control features. Similar to Stage I, the diffusion backbone then denoises $\hat{\mathbf{z}}_{\text{t}}$ conditioned on text, and we optimize A with the same diffusion objective (Eq. 3) on the corresponding target video.

3.5 Synthetic-Real Domain Hedging

A controllable video generation model would ideally be trained on paired data, where a coarse 3D input video is matched with a photorealistic target. In practice, such paired supervision can only be obtained from computer-generated (CG) data, which is expensive to produce, limited in diversity, and constrained by rendering fidelity. In contrast, real-world videos are abundant and available at a vastly larger scale, exhibiting rich variations in appearance, motion, and camera trajectories that are difficult to reproduce synthetically. To exploit this diversity and improve photorealism, we adopt a synthetic-real domain-hedging strategy, training C2R with *orders of magnitude more real videos than synthetic ones*, using real data to shape a strong photorealistic generative prior and synthetic data to provide sparse but essential paired supervision.

To bridge the domain gap between unpaired real and paired synthetic data, we incorporate a step within our training pipeline that maps both real videos and synthetic coarse renders into a *common spatio-temporal feature space*. In this space, control signals extracted from real videos and from coarse 3D simulations are encouraged to be compatible, enabling joint training despite their very different origins. Paired synthetic data anchors this space by linking coarse geometry to high-quality target appearance, while real videos densely populate it and enrich the learned generative distribution. Please refer to the Appendix Section A.3 for details on the collection and annotation of the synthetic and real datasets.

3.6 Adaptive Spatio-Temporal Control from Implicit Features

A central challenge in controllable video generation is extracting spatio-temporal control signals that generalize across real and synthetic domains. Explicit structural representations such as depth, edges,

optical flow, or poses Wang et al. [2023], NVIDIA et al. [2025], Cheng et al. [2024], Li et al. [2024] often entangle geometry with appearance, especially in real videos rich in texture and fine detail. As a result, they tend to over-constrain generation and encourage direct reconstruction rather than synthesis.

To address this, we adopt an *implicit* spatio-temporal representation that preserves scene layout and motion while abstracting away low-level appearance. We extract dense patch-level features using a pretrained self-supervised vision encoder Siméoni et al. [2025] and aggregate them temporally to form a spatio-temporal feature grid. These features (visualized in Figure 8) are appearance-robust and semantically structured, making them suitable as a shared control representation for both real videos and synthetic coarse renders.

To further reduce appearance leakage when using real videos, we apply a video-consistent HSV transformation to the control branch during training. This suppresses color and texture correlations while preserving geometric structure and temporal coherence, encouraging the control signal to focus on layout and dynamics rather than pixel fidelity. Details can be found in the Appendix Section A.1.3.

This implicit formulation naturally yields *adaptive control*. When the input is very coarse, the extracted features provide high-level structural cues, allowing the model to freely hallucinate fine-scale details and secondary motion. When richer structural information is present, the same mechanism enforces stronger adherence to the input motion and spatial configuration. As a result, C2R supports a wide range of inputs—from low-poly game-engine videos to more detailed simulations—without requiring different control strategies or hand-crafted rules.

3.7 Inference

As shown in Fig. 2, at inference time, given a coarse control video $\mathbf{x}_{\text{coarse}}$ and a text prompt $\mathbf{c}$, we compute a control latent $\hat{\mathbf{z}}_{\text{ctrl}} = A(\text{DINO}(\mathbf{x}_{\text{coarse}}))$ and a text embedding $\mathbf{e}_{\text{text}} = \text{T5}(\mathbf{c})$. We then perform standard diffusion sampling using a Flow Matching (Rectified Flow) sampler, which is solved with a first-order Euler integrator, starting from Gaussian noise $\mathbf{z}_T \sim \mathcal{N}(0, I)$. At each denoising step, control is injected via additive fusion (Eq. 7), while textual conditioning is incorporated through cross-attention. After the final step, the decoded output is obtained as $\hat{\mathbf{x}} = D(\hat{\mathbf{z}}_0)$.

3.7.1 Guidance control

To balance structural faithfulness and textual alignment, we use Adaptive Prompt Guidance (APG) Castillo et al. [2025] during sampling. APG dynamically scales conditioning contributions, allowing the model to deviate from coarse input appearance (e.g., colors/textures) while maintaining motion and layout.

4 Experiments

In this section, we present a comprehensive evaluation of C2R to assess its performance in synthesizing high-fidelity, controllable urban videos. We utilize a training dataset consisting of 240k clips of real-world footage and 1.3k clips of synthetic data, with specific details regarding collection and annotation provided in the Appendix Section A.3. Our evaluation begins by introducing the quantitative metrics used to measure both semantic alignment and structural fidelity. To further validate our architectural and data-driven design choices, we conduct ablations on the real-to-synthetic sampling ratio in our domain-hedging strategy, our additive feature injection strategy, and the necessity of HSV decorrelation. We then provide qualitative evidence of the model's adaptive capacity to handle varied levels of input geometry, ranging from low-poly simulations to more detailed renders. Finally, we compare against recent controllable video generation baselines, both off-the-shelf and fine-tuned on our data under the same regime as C2R, along three axes (control–prompt trade-off, visual quality and motion adherence), and report failure rates per input type. The full protocol and tables are given in Appendix A.4.

4.1 Metrics

We evaluate our method along two complementary axes: *text–video alignment* and *structural consistency*. To measure how well the generated video follows the input motion and structure, we use VE-Bench Sun et al. [2025], which is designed for video editing and control-based evaluation. While VE-Bench is applicable to our setting, it may penalize large appearance changes and favor conservative edits that closely preserve the source. To account for this limitation, we additionally report VQAScore Lin et al. [2025], a text–video alignment metric that directly evaluates prompt adherence. Using both metrics together provides a more balanced quantitative assessment of controllability and

(a) Quantitative results

Model	VE-Bench↑	VQAScore↑
100% synth.	0.4675	**0.5242**
50% real + 50% synth.	0.4370	0.5120
70% real + 30% synth.	0.4560	0.4538
90% real + 10% synth.	0.5020	0.4438
95% real + 5% synth.	0.4809	0.4262
99% real + 1% synth.	0.5317	0.3420
100% real	**0.8090**	0.1743
0 head	0.5317	0.3420
1 head	0.5837	**0.3536**
10 heads	**0.7893**	0.0046

(b) Ablation trends

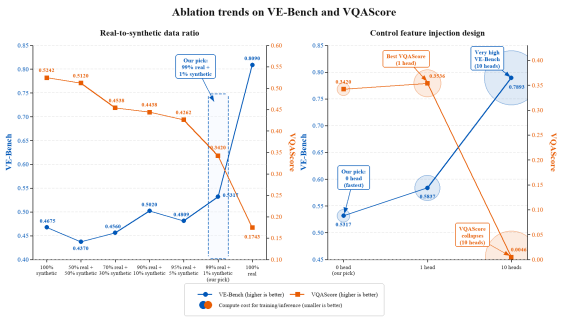

Figure 3: **Quantitative ablation results.** We report VE-Bench and VQAScore for different real-to-synthetic sampling ratios and control feature injection designs. (a) Numerical results for both ablation studies. (b) Corresponding trend plot. Bold and underlined values indicate the best and second-best results, respectively.

semantic alignment. Since neither metric isolates adherence to the control input from appearance quality, we additionally report *visual quality* as VBench Huang et al. [2024] Balanced Quality (the mean of six VBench dimensions) and *motion adherence* as RAFT Teed and Deng [2020] optical-flow agreement with the control video, calibrated against our matched coarse-to-fine synthetic pairs, whose motion is identical by construction (Appendix A.4).

4.2 Ablations

4.2.1 Real-to-Synthetic Sampling Ratio

We study how the real-to-synthetic sampling ratio in our domain-hedging strategy influences the trade-off between structural controllability (VE-Bench) and prompt adherence (VQAScore). For example, a ratio of 50% real + 50% synthetic indicates that, for each training batch, samples are drawn from the real and synthetic data pools with equal probability. Figure 3 shows that increasing the proportion of real data generally improves controllability, while progressively reducing prompt adherence. In particular, strongly real-dominant mixtures such as 90%/10% and 99%/1% achieve substantially higher VE-Bench scores than synthetic-heavy settings, at the cost of lower VQAScores.

When training exclusively on real data (100%), the model achieves the highest VE-Bench score but suffers a sharp drop in VQAScore, reflecting a loss of generative flexibility and a tendency to strictly reproduce the input rather than hallucinate novel content.

This behavior highlights the central trade-off in our domain-hedging strategy: real data strengthens the photorealistic generative prior and improves structural fidelity, while synthetic data preserves controllability by anchoring the mapping between coarse inputs and target appearance. These trends are also evident in the qualitative results shown in Figure 4. Based on both quantitative and qualitative evaluations, we adopt a strongly real-dominant ratio of 99% real and 1% synthetic data.

This ratio was selected on the held-out validation split (Appendix A.1.1) using both metrics jointly with qualitative inspection, since neither metric is meaningful alone, and the 1% synthetic component is what prevents the degenerate behaviour at 100% real. The structural drift at higher synthetic ratios follows from how the pairs are built. Synthetic pairs are aligned in motion but deliberately not in structure (unclothed body meshes and cuboid buildings against clothed, haired humans and detailed façades) and thus teach a large coarse-to-fine delta, whereas for real data the control branch and the target are the same video and the delta is zero, so the ratio sets how much deviation the adapter learns to treat as expected (Appendix A.5.3). The background pedestrians in Figure 4 are desired behaviour under user control, since C2R populates the background when the prompt asks for it (here, "people walking around") or when the control video contains coarse figures, and leaves the street empty otherwise.

4.2.2 Control feature injection strategy.

We inject DINO features from the control video into the first third of DiT blocks, which mainly influence structural control and global scene details. In Fig. 5 and Fig. 3, we compare alternative but computationally heavier strategies for integrating control features via projection heads: (0) direct addition of DINO features to noisy latents, (1) a single shared projection control head, and (10) dedicated projection control heads (1 per block). Surprisingly, direct addition (0 heads) achieves

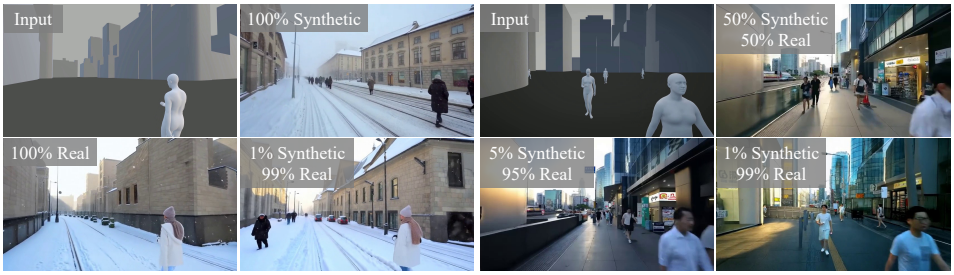

Figure 4: **Effect of the real-to-synthetic sampling ratio.** Synthetic-only training lacks diversity and weakly follows the control input, while real-only training improves structural alignment but reduces contextual richness. A strongly real-dominant ratio (99% real / 1% synthetic) provides the best trade-off, preserving control fidelity while enhancing detail generation. Increasing the proportion of synthetic data improves creativity but progressively weakens structural consistency, leading to hallucinations and deviations from the intended motion and layout. The left and right 2×2 blocks are two examples, each with its coarse input (top left) and three sampling ratios. Within a block the scene and the text prompt are fixed and only the ratio varies.

comparable quality to the 1-head approach, demonstrating that no extra learned projection is needed, training only the Adapter projection of the DINOv3 features to the diffusion latent space is enough. Meanwhile, per-block heads (10 heads) provide excessive capacity and produce reconstruction-like outputs, where what is reconstructed is *structure* rather than appearance, so that geometry, silhouettes and layout become pinned to the control input while colors and textures still follow the prompt. Since real data forms 99% of Stage II batches, structure is exactly recoverable on almost every batch, so the added capacity is optimized toward structural copying, which HSV decorrelation cannot prevent because what leaks is geometric rather than chromatic (Appendix A.5.3).This shows that simple feature addition is sufficient, and adding learnable capacity either provides no benefit or actively harms performance.



Figure 5: **Ablation of control feature injection strategies.** We compare direct additive injection after the DINO adapter (0 heads), a shared projection head (1 head), and per-block projection heads (10 heads). Direct addition is our preferred design, balancing control alignment and realistic detail synthesis while remaining computationally efficient, since it introduces no additional learnable parameters during training or inference. In contrast, additional heads increase conditioning capacity but can overfit the control signal and produce reconstruction-like outputs, in which structure is pinned to the input while appearance still follows the prompt. With 10 heads, the pedestrians keep the untextured silhouettes of the coarse input and the background retains its blocky geometry, even though the prompted buildings, lighting and ground colors are still rendered.

4.2.3 HSV Decorrelation

We study the effect of HSV decorrelation in the control branch during Stage II training. Without HSV decorrelation, control features retain low-level appearance cues, causing the model to inherit colors and textures from the input video, which is undesirable when appearance should be synthesized from text and learned priors. Applying a video-consistent HSV transformation suppresses appearance correlations while preserving structure and temporal coherence, encouraging the control signal to focus on spatio-temporal layout and preventing appearance leakage, as illustrated in Fig. 6.

4.3 Comparisons

To evaluate the effectiveness of C2R, we compare our framework against several recent controllable video generation and video-to-video translation baselines, including WAN VACE Jiang et al. [2025], Wan et al. [2025], WAN+ControlNet Wan et al. [2025], Zhang et al. [2023], Team [2025], Diffusion as Shader (DaS) Gu et al. [2025], Control-A-Video Chen et al. [2023c], and Seedance 2.0 Team Seedance et al. [2026]. These methods rely on different forms of conditioning, such as depth maps, tracking signals, or reference images, for video synthesis. In contrast, C2R operates directly on the coarse control video and prompt, without geometric preprocessing or reference-image conditioning.



Figure 6: **Effect of HSV decorrelation in the control branch.** Without HSV decorrelation, the model inherits undesired appearance cues from the control input. Video-consistent HSV decorrelation suppresses color and texture leakage while preserving structure and temporal coherence.

For a fair comparison, all methods are evaluated under the same target prompt and coarse low-poly block-based control setting shown in Fig. 7. WAN VACE uses depth sequences extracted from the input control video together with the text prompt. WAN+ControlNet corresponds to our finetuned WAN-ControlNet baseline trained on the same synthetic paired data as C2R, enabling direct conditioning on the input control video and text prompt without requiring depth or tracking extraction. Diffusion as Shader (DaS) requires additional tracking supervision computed with SpaTracker Xiao et al. [2024], as well as a reference image. We evaluate two DaS variants: DaS-FirstFrame, where the reference frame is generated with FLUX Black Forest Labs [2024b,a] following the original DaS pipeline, and DaS-NoFirstFrame, which uses the first frame of the control sequence as reference. Control-A-Video, like WAN VACE, relies on depth-conditioned video generation. Finally, we also compare against Seedance 2.0 using the same control video and prompt inputs.

As illustrated in Fig. 7, existing baselines struggle to translate highly coarse low-poly block characters into realistic video outputs. WAN VACE preserves the overall camera trajectory and scene structure, but its depth-based conditioning strongly constrains the generated humans to the coarse input geometry, resulting in rigid and insufficiently realistic appearances. WAN+ControlNet partially enables coarse-to-real transfer while maintaining strong structural adherence, but remains visually conservative and lacks expressive detail synthesis. Reference-image-based approaches such as DaS improve visual fidelity when using a generated reference frame (DaS-FF), but still suffer from inaccurate motion alignment, while DaS-NFF often produces reconstruction-like outputs tied to the coarse input appearance. Control-A-Video similarly preserves coarse identity cues, yet often fails to follow intended motion and camera dynamics while producing low-quality generations. Seedance 2.0 generates highly realistic imagery, yet frequently fails to preserve intended motion and scene alignment under coarse inputs, occasionally introducing visible reconstruction artifacts from the block-based signal. In contrast, C2R preserves layout, motion, and scene dynamics while generating photorealistic humans and environments with richer textures, lighting, and contextual detail.

Quantitative comparison. Table 1 complements the qualitative comparison with a head-to-head evaluation on identical inputs (50 control videos $\times$ 3 prompts for VE-Bench and VQAScore, and 50 control videos per input category for VBench and RAFT), with the full protocol and all configurations in Appendix A.4. Since comparing off-the-shelf checkpoints against a model trained on our data would not be a fair test, we additionally fine-tune WAN VACE, WAN+ControlNet and DaS on our data under the same regimes as C2R (100% synthetic, and 99% real + 1% synthetic), conditioned directly on the coarse control video. Control-A-Video offers no compatible training path, and Seedance 2.0 cannot be retrained and is compared qualitatively only. VE-Bench and VQAScore reward opposite failure modes and must be read jointly, since VE-Bench scores reconstruction highly whereas VQAScore scores uncontrolled generation highly, and off-the-shelf configurations fall into one corner or the other (VACE at 0.0492/0.7419 ignores the control signal, whereas DaS at 0.8333/0.0305 reconstructs rather than renders). Under matched 99%/1% training, C2R improves on VACE and WAN+ControlNet on both metrics simultaneously and attains far higher VE-Bench than DaS at comparable VQAScore, without depth, edge or tracking extraction, without a reference image, and without additional parameters at inference. It also ranks first in VBench Balanced Quality in every category and is the only method whose calibrated motion proximity stays low across all three types, with failure rates of 2%, 6% and 0% on humanoid, blocky and game inputs (Appendix A.4.5). Finally, the stronger structural adherence of WAN VACE and other models in Fig. 7 is not by itself an advantage, since coarse figures lack hair or clothing, and loose garments, accessories or hair must extend beyond their silhouette with dynamics the control signal does not contain.

To further explore these differences, Appendix Section A.5.1 presents the same evaluation protocol using humanoid control inputs instead of low-poly block characters. The results further show that C2R achieves a strong balance between visual richness, structural grounding, and motion alignment.

Model	VE-Bench↑	VQAScore↑	VBench BQ↑	RAFT prox.↓
VACE Base (depth)	0.0492	0.7419	0.7961	0.0756
VACE 100% synth.	0.1488	0.7126	<u>0.8371</u>	<u>0.0503</u>
VACE 99% real + 1% synth.	0.2554	0.1817	0.8138	0.1194
WAN+CN Base (depth)	0.0544	0.5637	0.7995	0.0512
WAN+CN 100% synth.	0.0941	0.3742	0.8306	0.0673
WAN+CN 99% real + 1% synth.	0.3199	0.1314	0.8145	0.0876
DaS Base	0.8333	0.0305	0.7116	0.1442
DaS 100% synth.	−0.0805	0.7566	0.8066	0.0919
DaS 99% real + 1% synth.	0.1513	0.3555	0.7982	0.0586
Control-A-Video (depth)	−0.7681	0.0679	0.7292	0.0945
C2R 99% real + 1% synth. (ours)	0.5317	0.3420	**0.8619**	**0.0445**

Table 1: **Quantitative comparison against baselines on identical inputs.** VE-Bench and VQAScore (50 control videos × 3 prompts) reward opposite failure modes and must be read jointly, while VBench Balanced Quality (BQ) and calibrated RAFT motion proximity are averaged over the humanoid, blocky and game categories (50 control videos each). "Base" is the public checkpoint with its native conditioning, and "100% synth." and "99% real + 1% synth." denote the baseline fine-tuned on our data under the same regime as C2R, conditioned directly on the coarse control video. Only the depth variants of the off-the-shelf baselines are shown, and all 16 configurations are reported in Appendix A.4. **Bold** and <u>underlined</u> mark the best and second-best VBench BQ and RAFT proximity.



Figure 7: **Baseline comparison on coarse low-poly block-based character control.** We compare C2R against WAN VACE, WAN+ControlNet, Diffusion as Shader (DaS), Control-A-Video, and Seedance 2.0 under the same text prompt and coarse control input. Existing methods struggle to translate coarse low-poly block characters into realistic videos, often producing weakly controlled or reconstruction-like outputs. In contrast, C2R transfers the coarse motion and layout into realistic humans and environments while preserving strong structural consistency and richer visual detail.

5 Limitations and Future Work

Our method relies on coarse 3D inputs to guide scene layout, camera motion, and human dynamics. When the input geometry is extremely sparse or abstract, it may not provide sufficient structural cues to precisely control the desired layout or camera trajectory. In such cases, text prompts alone may be ambiguous and insufficient to fully disambiguate camera motion or character movement, as illustrated in the Appendix Section A.5.6. Future work includes incorporating more explicit control signals, such as camera motion directions, speed profiles, or character movement constraints, to improve controllability. Additionally, the current framework operates in a non-autoregressive manner; extending the model to an autoregressive formulation could enable real-time generation and interactive applications.

Finally, we would like to note that no existing metric isolates adherence to the control input from appearance quality. VQAScore penalizes faithful control whenever the prompt and control disagree, while VE-Bench penalizes the appearance change that coarse-to-real generation must produce. Because the joint ceiling of the two metrics lies well below $(1, 1)$ (Appendix A.4.6), finding a completely suitable evaluation metric remains an open challenge, and developing a task-specific metric that scores control adherence separately from prompt realization is an important direction for future work. Likewise, the 99%/1% ratio is specific to our setup, and the transferable finding is that a small, carefully constructed paired set combined with abundant unpaired data suffices to enable a task for which paired supervision does not otherwise exist. On the data side, although our Stage-I prior is learned from an internal real-video corpus, Appendix A.1.5 shows that a publicly available urban dataset yields comparable quality, and Appendix A.4.5 quantifies failure rates per input type. Generalization beyond urban crowd scenes remains to be studied.

Acknowledgments and Disclosure of Funding

We thank Super Dimension (Jiaxing) InfoTech Ltd. for providing the 3D scanned human models used in this work. We also thank Dave Cardwell for the initial and insightful discussions on production-level requirements for crowd simulation; this project originated from those early conversations. The presented results are partially supported by the project "Inteligencia artificial para la industria 4.0: generación de datos, modelado avanzado optimización e interpretabilidad" (Acronym: IDEA-CM), with reference TEC-2024/COM-89, partially funded by the Comunidad de Madrid through the call for grants for collaborative R&D projects under the modality of "Programas de Actividades de I+D en Tecnologías 2024", according to Order 3177/2024.



References

abhayexe. City 3d models collection. https://sketchfab.com/abhayexe, 2024. Creative Commons licensed 3D assets; accessed January 29, 2026.

Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, page 408–416, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 9781450378253. doi: 10.1145/1186822.1073207. URL https://doi.org/10.1145/1186822.1073207.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report, 2023.

Black Forest Labs. FLUX. https://github.com/black-forest-labs/flux, 2024a.

Black Forest Labs. FLUX.1 Depth [dev]. https://huggingface.co/black-forest-labs/FLUX.1-Depth-dev, 2024b. Hugging Face model card. Accessed: 2026-05-07.

Angela Castillo, Jonas Kohler, Juan C. Pérez, Juan Pablo Pérez, Albert Pumarola, Bernard Ghanem, Pablo Arbeláez, and Ali Thabet. Adaptive guidance: Training-free acceleration of conditional diffusion models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(2):1962–1970, Apr. 2025. doi: 10.1609/aaai.v39i2.32192. URL https://ojs.aaai.org/index.php/AAAI/article/view/32192.

Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter1: Open diffusion models for high-quality video generation, 2023a.

Tsai-Shien Chen, Chieh Hubert Lin, Hung-Yu Tseng, Tsung-Yi Lin, and Ming-Hsuan Yang. Motion-conditioned diffusion model for controllable video synthesis. *arXiv preprint arXiv:2304.14404*, abs/2304.14404, 2023b.

Weifeng Chen, Yatai Ji, Jie Wu, Hefeng Wu, Pan Xie, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video diffusion models with motion prior and reward feedback learning. *arXiv preprint arXiv:2305.13840*, 2023c.

Jiaxin Cheng, Tianjun Xiao, and Tong He. Consistent Video-to-Video Transfer Using Synthetic Dataset. In *International Conference on Learning Representations (ICLR)*, pages 1–13, Vienna, Austria, 2024. OpenReview.net. URL https://openreview.net/forum?id=IoKRezZMxF.

Ruihang Chu, Yefei He, Zhekai Chen, Shiwei Zhang, Xiaogang Xu, Bin Xia, Dingdong WANG, Hongwei Yi, Xihui Liu, Hengshuang Zhao, Yu Liu, Yingya Zhang, and Yujiu Yang. Wan-move: Motion-controllable video generation via latent trajectory guidance. In *Annual Conference on Neural Information Processing Systems*, pages 1–29, Vancouver, Canada, 2025. NeurIPS Foundation. URL https://openreview.net/forum?id=lHW93LKaUk.

Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7312–7322, Paris, France, 2023. IEEE Computer Society. doi: 10.1109/ICCV51070.2023.00675.

Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara, Yusuf Aytar, Michael Rubinstein, Chen Sun, Oliver Wang, Andrew Owens, and Deqing Sun. Motion prompting: Controlling video generation with motion trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–12, Nashville, TN, USA, June 2025. IEEE Computer Society.

Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, Wenping Wang, and Yuan Liu. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In *ACM SIGGRAPH 2025 Conference Papers*, 2025. doi: 10.1145/3721238.3730607.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling Camera Control for Video Diffusion Models. In *International Conference on Learning Representations (ICLR)*, pages 1–32, Singapore, 2025. OpenReview.net. URL `https://openreview.net/forum?id=Z4ev0UYrk7`.

Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51 (5):4282, 1995.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video Diffusion Models. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:8633–8646, 2022.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Adobe Inc. Mixamo: Automatic rigging and animation service. `https://www.mixamo.com`, 2013. Accessed: 2026-01-29.

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*, pages 17191–17202, 2025.

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

Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. *ACM Trans. Graph.*, 21(3): 473–482, July 2002. ISSN 0730-0301. doi: 10.1145/566654.566605. URL `https://doi.org/10.1145/566654.566605`.

Xirui Li, Chao Ma, Xiaokang Yang, and Ming-Hsuan Yang. Vidtome: Video token merging for zero-shot video editing. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7486–7495, Seattle, WA, USA, 2024. IEEE Computer Society. doi: 10.1109/CVPR52733.2024.00715.

Zhen Li, Chuanhao Li, Xiaofeng Mao, Shaoheng Lin, Ming Li, Shitian Zhao, Zhaopan Xu, Xinyue Li, Yukang Feng, Jianwen Sun, Zizhen Li, Fanrui Zhang, Jiaxin Ai, Zhixiang Wang, Yuwei Wu, Tong He, Jiangmiao Pang, Yu Qiao, Yunde Jia, and Kaipeng Zhang. Sekai: A video dataset towards world exploration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2506.15675.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 366–384, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72673-6.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6), November 2015. ISSN 0730-0301. doi: 10.1145/2816795.2818013. URL https://doi.org/10.1145/2816795.2818013.

NVIDIA, Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, Prithvijit Chattopadhyay, Mike Chen, Yongxin Chen, Yu Chen, Shuai Cheng, Yin Cui, Jenna Diamond, Yifan Ding, Jiaojiao Fan, Linxi Fan, Liang Feng, Francesco Ferroni, Sanja Fidler, Xiao Fu, Ruiyuan Gao, Yunhao Ge, Jinwei Gu, Aryaman Gupta, Siddharth Gururani, Imad El Hanafi, Ali Hassani, Zekun Hao, Jacob Huffman, Joel Jang, Pooya Jannaty, Jan Kautz, Grace Lam, Xuan Li, Zhaoshuo Li, Maosheng Liao, Chen-Hsuan Lin, Tsung-Yi Lin, Yen-Chen Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Yifan Lu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Seungjun Nah, Yashraj Narang, Abhijeet Panaskar, Lindsey Pavao, Trung Pham, Morteza Ramezanali, Fitsum Reda, Scott Reed, Xuanchi Ren, Haonan Shao, Yue Shen, Stella Shi, Shuran Song, Bartosz Stefaniak, Shangkun Sun, Shitao Tang, Sameena Tasmeen, Lyne Tchapmi, Wei-Cheng Tseng, Jibin Varghese, Andrew Z. Wang, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Jiashu Xu, Dinghao Yang, Xiaodong Yang, Haotian Ye, Seonghyeon Ye, Xiaohui Zeng, Jing Zhang, Qinsheng Zhang, Kaiwen Zheng, Andrew Zhu, and Yuke Zhu. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, abs/2511.00062, 2025. URL https://arxiv.org/abs/2511.00062.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. arXiv:2304.07193.

Qwen Team. Qwen3.6-27b: Flagship-level coding in a 27b dense model. https://qwen.ai/blog?id=qwen3.6-27b, 2026. Alibaba Cloud. Accessed October 22, 2026.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6121–6132, Nashville, TN, USA, 2025. IEEE Computer Society. doi: 10.1109/CVPR52734.2025.00574.

Ludan Ruan, Yiyang Ma, Huan Yang, Huiguo He, Bei Liu, Jianlong Fu, Nicholas Jing Yuan, Qin Jin, and Baining Guo. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10219–10228, Los Alamitos, CA, 2023. IEEE Computer Society. doi: 10.1109/CVPR52729.2023.00985.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025. URL https://arxiv.org/abs/2508.10104.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*, pages 1–16, Kigali, Rwanda, 2023. OpenReview.net. URL https://openreview.net/forum?id=nJfylDvgzlq.

Shangkun Sun, Xiaoyu Liang, Songlin Fan, Wenxu Gao, and Wei Gao. Ve-bench: Subjective-aligned benchmark suite for text-driven video editing quality assessment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(7):7105–7113, 2025.

Super Dimension. Super dimension 3d human scanned models. http://en.superdimension.cn/, 2026. Website and 3D scanning service; accessed January 29, 2026.

Alibaba PAI Team. Wan2.1-fun-14b-control: Advanced multi-condition video generation weights. https://huggingface.co/alibaba-pai/Wan2.1-Fun-14B-Control, 2025. Hugging Face Repository.

Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, 2020.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025. URL https://arxiv.org/abs/2503.20314.

Dilin Wang, Hyunyoung Jung, Tom Monnier, Kihyuk Sohn, Chuhang Zou, Xiaoyu Xiang, Yu-Ying Yeh, Di Liu, Zixuan Huang, Thu Nguyen-Phuoc, Yuchen Fan, Sergiu Oprea, Ziyan Wang, Roman Shapovalov, Nikolaos Sarafianos, Thibault Groueix, Antoine Toisoul, Prithviraj Dhar, Xiao Chu, Minghao Chen, Geon Yeong Park, Mahima Gupta, Yassir Azziz, Rakesh Ranjan, and Andrea Vedaldi. WorldGen: From Text to Traversable and Interactive 3D Worlds, 2025. URL https://arxiv.org/abs/2511.16825.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20697–20709, Los Alamitos, CA, USA, 2024. IEEE Computer Society. doi: 10.1109/CVPR52733.2024.01956.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 1152–1164, Red Hook, NY, USA, 2018. Curran Associates Inc.

Wen Wang, kangyang Xie, Zide Liu, Hao Chen, Yue Cao, Xinlong Wang, and Chunhua Shen. Zero-shot video editing using off-the-shelf image diffusion models. *arXiv preprint arXiv:2303.17599*, abs/2303.17599, 2023.

Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan.Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer. In *International Conference on Learning Representations (ICLR)*, pages 1–30, Singapore, 2025. OpenReview.net. URL https://openreview.net/forum?id=LQzN6TRFg9.

Liping Yuan, Jiawei Wang, Haomiao Sun, Yuchen Zhang, and Yuan Lin. Tarsier2: Advancing large vision-language models from detailed video description to comprehensive video understanding, 2025. URL https://arxiv.org/abs/2501.07888.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3813–3824, Paris, France, 2023. IEEE Computer Society. doi: 10.1109/ICCV51070.2023.00355.

A Appendix

A.1 Implementation Details

A.1.1 Training Protocol

We follow the standard training protocol of latent diffusion and flow-matching video generation models. Videos are temporally sampled to 81 frames at 16 fps, resized and center-cropped to the 832×480 training resolution, and encoded into the latent space of the pretrained Wan 2.1 VAE. Text prompts are encoded with the frozen T5-XXL text encoder. At each training step, we sample a timestep t from the flow-matching schedule, perturb the clean latent $\mathbf{z}_0$ with Gaussian noise to obtain $\mathbf{z}_t$, and optimize the model to predict the corresponding velocity field using the objective in Eq. 3.

Training is performed in two stages. In Stage I, we fine-tune the diffusion backbone on real videos to adapt the pretrained model to our target urban video distribution, while keeping the VAE and text encoder frozen. In Stage II, we freeze the Stage-I backbone and train only the control pathway, namely the DINO feature adapter that projects temporally concatenated frame-wise features into the diffusion latent space. The DINO encoder itself remains frozen. Stage II uses the synthetic-real domain-hedging strategy described in Section 3.5, where paired synthetic data provides explicit coarse-to-real supervision and large-scale real videos regularize the control pathway toward the photorealistic distribution learned in Stage I.

We split the data into training and held-out validation sets using a 90–10 ratio for both the real and the synthetic data, ensuring that videos from the same source sequence do not appear in both splits. The validation set is used to monitor convergence, compare ablations, and inspect fixed validation generations throughout training, while final results are computed on a separate evaluation set. Although the flow-matching validation loss provides a useful convergence signal, checkpoint selection is based on both validation-loss trends and qualitative assessment of the realism–controllability trade-off, including temporal consistency, prompt alignment, control adherence, and appearance leakage.

We use an effective global batch size of 32. Both stages are optimized with AdamW using mixed-precision distributed training. Stage I uses a learning rate of 1×10^{-5}, while Stage II uses a learning rate of 1×10^{-4}. All ablations use the same optimizer settings, training resolution, data split, and validation protocol for fair comparison. Checkpoints are saved periodically during training. The complete two-stage training process takes approximately three weeks on a single node with eight NVIDIA H100 GPUs using data-parallel training.

A.1.2 Shared Feature Space

In Fig. 8, we demonstrate how DINOv3 features serve as the foundation for a common spatio-temporal feature space that bridges the domain gap between unpaired real videos and paired synthetic data. By mapping both real videos and coarse 3D renders into this shared representation, our pipeline ensures that control signals from vastly different origins are structurally compatible for joint training. The visualization of patch embeddings via a global PCA projection reveals that DINOv3 aligns corresponding semantic elements across the real and synthetic domains, effectively neutralizing appearance gaps. This shared space allows paired synthetic data to act as a structural anchor linking geometry to target appearance, while real videos populate the space to enrich the learned generative distribution.

The PCA colors in Figure 8 should be read as similarity in DINO feature space rather than as semantic labels. Each 16×16 patch feature is projected onto the first three principal components of a single global PCA computed over full frames of mixed real and synthetic samples, without foreground filtering, and displayed as RGB. Bilinear upsampling of the coarse patch grid produces the smooth gradients. Because full-frame DINO features encode global scene properties such as depth, scale and surface orientation (depth is linearly decodable from DINO patch embeddings), the leading components are dominated by these properties over large, semantically homogeneous regions. A near-field façade filling the frame and an open street with buildings receding into the distance therefore receive different colors even though both are "buildings". The coarse-to-real correspondence still appears in our global projection where elements are most comparable. Pedestrians fall in a similar color range in the real and coarse rows despite the large appearance gap, because compact, high-salience objects are dominated by object structure rather than by the low-frequency spatial signal of the background. PCA is used purely for visualization, and the adapter consumes the full patch embeddings. Figure 9 complements Figure 8 by showing the correspondence directly, following the qualitative patch-matching protocol of DINOv2 Oquab et al. [2024]. Mutual nearest neighbours under cosine similarity between the DINOv3 patch tokens of a coarse crop and those of its real

counterpart yield spatially consistent per-element correspondences between coarse body meshes and real pedestrians, and between coarse building volumes and real façades, despite the large appearance gap.

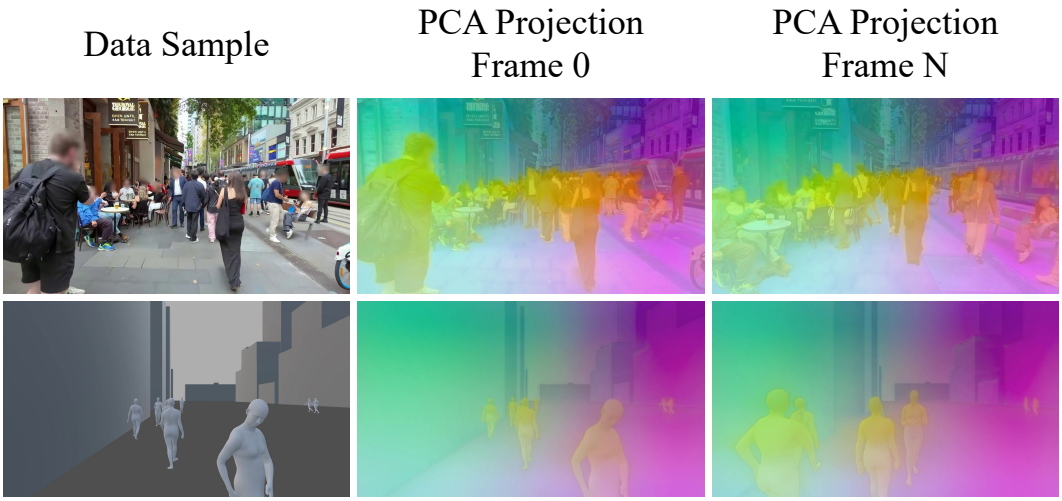

Figure 8: **DINOv3 features provide a domain-robust and temporally stable control signal.** We visualize patch embeddings using a *global* PCA projection computed on a mixed subset of real and synthetic samples, then reused across videos. Similar PCA colors across real and synthetic inputs indicate that DINOv3 aligns corresponding structural elements despite large appearance gaps. Color stability across time suggests temporal coherence even when features are extracted per-frame, supporting spatio-temporal control. Colors denote similarity in feature space under a single global PCA, not semantic labels (see the text and Figure 9).

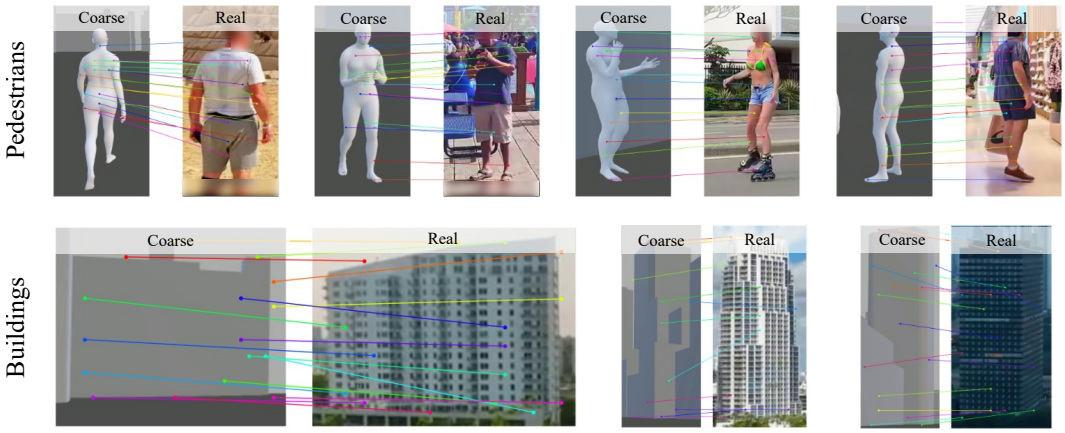

Figure 9: **Coarse-to-real patch correspondences of DINOv3 features.** Patch correspondences between coarse and real crops of pedestrians (top) and buildings (bottom), following the qualitative matching protocol of DINOv2 Oquab et al. [2024]. We extract last-layer patch tokens from a frozen DINOv3 ViT-B/16, keep object patches, and draw mutual nearest neighbours under cosine similarity (matches are spatially thinned for display).

A.1.3 Preventing Feature Leakage with Video-Consistent HSV Decorrelation

Problem. DINO features are expressive: if the control branch receives the same pixels as the VAE branch, the adapter can pass appearance information that encourages reconstruction rather than synthesis.

Solution. For real videos during Stage II, we feed the same video $\mathbf{x}$ to both branches, but apply a *video-consistent* random HSV transformation only to the control branch:

$$\mathbf{x}_{\text{ctrl}} = \text{HSV}(\mathbf{x}), \tag{8}$$

and compute $\hat{\mathbf{z}}_{\text{ctrl}} = A(\text{DINO}(\mathbf{x}_{\text{ctrl}}))$. By randomly shifting hue and scaling saturation/value consistently across frames, the adapter is forced to prioritize geometry and dynamics over pixel-level appearance. Compared to grayscale, HSV decorrelation avoids systematically biasing outputs toward muted colors and better preserves realistic color diversity. No augmentation is required at inference time.

In a nutshell, we apply a temporally consistent HSV decorrelation augmentation independently to each training video. For every video, a single random HSV transformation is sampled and applied uniformly across all frames to preserve temporal coherence while altering appearance statistics between videos. Specifically, we randomly sample a hue offset with magnitude between 20° and 90°, with equal probability of positive or negative rotation. In addition, saturation is randomly scaled by $\pm 15\%$ to $\pm 40\%$, and value (brightness) by $\pm 15\%$ to $\pm 30\%$. Each frame is converted from RGB to HSV space, transformed using the same per-video hue offset and saturation/value scaling factors, clipped to the valid range, and converted back to RGB before being used for training. This augmentation reduces appearance leakage while preserving the underlying structure and motion consistency of the control signal.

A.1.4 Post-Processing and Artifact Mitigation

Evaluation is done on raw outputs of the C2R model, without any postprocessing. For presentation purposes, a general watermark removal model was used in Figure 1.

A.1.5 Reproducibility with Public Data

To assess whether our contribution depends on privileged data, we trained a variant of C2R in which the internal real-video corpus is replaced by SEKAI Li et al. [2025], a publicly available large-scale dataset of urban walking footage, keeping the synthetic data, the two-stage protocol and all hyperparameters unchanged. This variant achieves comparable quality to the model reported in the paper, indicating that our contribution does not depend on privileged data. Any sufficiently large and diverse urban video corpus can serve as the Stage-I photorealistic prior, and the training distribution can be reproduced from public data.

A.2 Existing Video Models for Populated Urban Scene Generation

Fig. 10 illustrates the difficulty of generating scenes similar to those presented in our experiments using traditional text-to-video models without any control signal, even when provided with detailed prompts. In particular, prompts involving cities, crowds, or general human activities often lead to fast-forward-like scene dynamics, unstable motion, and limited temporal consistency. Moreover, character identity, layout, and camera trajectories remain extremely difficult to control reliably in purely prompt-driven generation. These limitations motivate the need for controllable video generation frameworks capable of leveraging the strong generative priors and world knowledge of large-scale video models while grounding them toward specific tasks, scenarios, and structural constraints.

A.3 Data Collection

We curate a large-scale dataset consisting of both synthetic and real-world videos to support training and evaluation.

For synthetic data generation, we use professional 3D content creation tools to simulate humans interacting within complex urban 3D environments. The scenes are of AAA visual quality, composed of fully textured city models with realistic geometry, materials, lighting, and post-processing effects, and populated with high-fidelity human assets obtained from detailed 3D scans Super Dimension [2026]. City environments were obtained from Sketchfab and created by Abhayexe under a Creative Commons license abhayexe [2024]. To enable paired supervision, each city environment is additionally converted into a corresponding coarse representation by approximating the volume of every building with simple flat geometric primitives (e.g., cuboids), producing a simplified city layout that preserves large-scale structure while removing fine visual detail. Vehicles are treated similarly, being replaced in the coarse domain by simple, untextured meshes that approximate their overall volume without appearance cues.

For character pairing, each high-quality scanned human model is matched with a coarse counterpart consisting of a simple unclothed human body mesh without hair or accessories. Both the full-quality and coarse characters are driven by the same motion data by retargeting identical Mixamo animations Inc. [2013], ensuring precise alignment of motion, pose dynamics, and overall body structure across



Figure 10: SORA (left column) and WAN (right column) show limited controllability over human motion and camera trajectories, and tend to generate similar viewing angles across populated urban scenes.

representations. This pairing strategy guarantees that differences between the full and coarse renders arise solely from appearance and geometric detail, rather than from motion discrepancies.

For each sample in the synthetic dataset, we randomly spawn a crowd of characters in a street region of the city, assign randomized textures and animations to the full-quality assets, and propagate the corresponding animations to their coarse counterparts. Two cameras with identical trajectories are then attached to a randomly selected character in the crowd and used to record a short replay sequence of 5 seconds. The full camera exclusively captures the high-quality assets, including detailed city geometry, vehicles, scanned characters, lighting, and post-processing effects, while the coarse camera records only the simplified elements: unclothed character body meshes, a ground plane, and the coarse volumetric city and vehicle representations rendered with a uniform, neutral material (e.g., grayscale). Each iteration produces one paired data sample consisting of a full-quality video and its corresponding coarse video, enabling learning of fine-grained visual detail from structured but minimal 3D representations.

For real data, we collect street-view video footage from cities spanning all five continents, covering a broad variety of urban environments. The data includes a wide range of viewpoints and motion characteristics, such as static and dynamic captures, first- and third-person perspectives, and varying viewing directions. We segment the videos into short clips to facilitate training.

We automatically generate textual annotations for both real-world videos and full-quality synthetic clips using Tarsier Yuan et al. [2025], a state-of-the-art video captioning system built on top of the Qwen large multimodal model Bai et al. [2023]. For each video, we employ a simple and generic prompt (e.g., "Describe the video in detail"), relying on the strong video–language understanding capabilities of Tarsier to produce rich, free-form captions without manual engineering of task-specific prompts.

The resulting captions consistently capture high-level scene context and fine-grained semantic attributes, including human actions and motion patterns, camera motion, clothing appearance, crowd density, weather conditions, environmental layout, architectural style, and location cues derived from the appearance of the city. This process yields expressive and diverse textual descriptions that reflect realistic real-world semantics and visual variability.

We apply this automatic captioning procedure to all real video clips in the dataset as well as to the full-quality synthetic renders. Coarse synthetic videos are not captioned separately; instead, each coarse

clip inherits the caption of its corresponding full-quality counterpart. Since both representations are perfectly paired in terms of motion, structure, and scene layout, the caption associated with the realistic render provides an accurate semantic description for the paired coarse input. This design allows the model to learn to condition generation on realistic, high-level textual supervision while operating on minimal and abstract 3D visual representations.

In total, our dataset contains 240K real-world video clips and 1.3K synthetic video clips, each with 5 seconds sampled at 16 fps.

A.4 Quantitative Comparison Protocol and Full Results

A.4.1 Protocol

All methods are evaluated on identical inputs. We use 50 control videos $\times$ 3 prompts for VE-Bench and VQAScore, and 50 control videos per input category (coarse humanoid, low-poly blocky and game capture) for VBench, RAFT and the failure analysis. Since comparing off-the-shelf checkpoints against a model trained on our data would not be a fair test, we additionally fine-tune WAN VACE Jiang et al. [2025], WAN2.1+ControlNet Team [2025] and Diffusion as Shader (DaS) Gu et al. [2025] on our data under the same regime as C2R, i.e., 100% synthetic and 99% real + 1% synthetic, conditioned directly on the coarse control video, which yields 16 baseline configurations in total (Table 2). "Base" rows denote the public checkpoint with its native conditioning signal extracted from the control video (RGB, depth or Canny).

Control-A-Video Chen et al. [2023c] was not fine-tuned, since its public repository offers no compatible training path and it generates 7 frames at 512×512 against 81 frames at 832×480 elsewhere. We report its three published variants (depth, Canny and HED). Seedance 2.0 Team Seedance et al. [2026] cannot be retrained (closed weights and pipeline), and running the full evaluation through its commercial API exceeds our academic budget. It is therefore included qualitatively only (Figures 7 and 12) as a reference for the default capabilities of a strong commercial model, and should not be read as a controlled comparison. DaS requires first-frame conditioning, so we supply the first frame of the same control video used by all methods (DaS-NFF) and exclude its generated-reference modality (DaS-FF), as it would grant information unavailable to the other methods.

A.4.2 Control–Prompt Trade-off

Table 2 reports VE-Bench and VQAScore for all configurations. The two metrics reward opposite failure modes and must be read jointly, since VE-Bench measures fidelity to the source and scores reconstruction highly, whereas VQAScore measures prompt alignment and scores uncontrolled generation highly. Off-the-shelf configurations fall into one corner or the other. Depth conditioning gives near-zero VE-Bench with high VQAScore (VACE 0.0492/0.7419 and WAN+CN 0.0544/0.5637), meaning the control signal is effectively ignored, while DaS Base gives the reverse (0.8333/0.0305), reconstruction rather than rendering, and Control-A-Video collapses on prompt adherence (best VQAScore 0.0679). Under matched 99%/1% training, C2R (0.5317/0.3420) improves on VACE (0.2554/0.1817) and WAN+CN (0.3199/0.1314) on both metrics simultaneously, and attains far higher VE-Bench than DaS (0.1513) at comparable VQAScore, without depth, edge or tracking extraction, without a reference image, and without additional parameters at inference. Notably, VACE and WAN+CN both move out of their degenerate corners under 99%/1% training, which highlights the applicability of the synthetic-real domain-hedging strategy to other backbones.

A.4.3 Visual Quality

Table 3 reports VBench Huang et al. [2024] Balanced Quality, the unweighted mean of six VBench dimensions (subject consistency, background consistency, motion smoothness, dynamic degree, aesthetic quality and imaging quality), per input category. C2R ranks first in all three categories (0.8707 / 0.8669 / 0.8479) and first overall (0.8619, $+0.0248$ over the closest baseline), led by imaging quality in every category. It also ties for the best dynamic degree on humanoid and blocky inputs, while on game inputs it leads subject consistency, aesthetic quality and imaging quality. Several baselines score higher in background consistency and motion smoothness, dimensions that often reward stable or relatively static synthetic content rather than realistic transformation. The advantage of C2R therefore primarily reflects stronger visual quality and dynamics without sacrificing temporal stability.

Model	VE-Bench↑	VQAScore↑
Wan2.1 VACE		
Base (RGB)	0.4882	0.2837
Base (depth)	0.0492	0.7419
Base (Canny)	−0.0252	0.3856
100% synth.	0.1488	0.7126
99% real + 1% synth.	0.2554	0.1817
Wan2.1 + CN		
Base (RGB)	0.0439	0.2568
Base (depth)	0.0544	0.5637
Base (Canny)	0.0086	0.0478
100% synth.	0.0941	0.3742
99% real + 1% synth.	0.3199	0.1314
Diffusion as Shader		
Base	0.8333	0.0305
100% synth.	−0.0805	0.7566
99% real + 1% synth.	0.1513	0.3555
Control-A-Video		
Depth	−0.7681	0.0679
Canny	−0.2649	0.0420
HED	0.3573	0.0025
C2R (ours) — real/synthetic ratio		
100% synth.	0.4675	0.5242
50% real + 50% synth.	0.4370	0.5120
70% real + 30% synth.	0.4560	0.4538
90% real + 10% synth.	0.5020	0.4438
95% real + 5% synth.	0.4809	0.4262
99% real + 1% synth.	0.5317	0.3420
100% real	0.8090	0.1743
C2R (ours) — injection heads		
0 head	0.5317	0.3420
1 head	0.5837	0.3536
10 heads	0.7893	0.0046

Table 2: **VE-Bench and VQAScore across all baseline configurations and C2R ablations.** All methods are evaluated on identical inputs (50 control videos × 3 prompts). "Base" denotes the off-the-shelf checkpoint with its native conditioning (RGB, depth or Canny extracted from the control video), and "100% synth." and "99% real + 1% synth." denote the baseline fine-tuned on our data under the same regime as C2R, conditioned directly on the coarse control video. Control-A-Video is reported in its three published variants without fine-tuning. The C2R rows repeat the ablation results of Figure 3(a) for reference.

Model	Humanoid ↑	Blocky ↑	Game ↑	Overall ↑
C2R 99/1 (ours)	**0.8707**	**0.8669**	**0.8479**	**0.8619**
VACE 100% synth	<u>0.8600</u>	<u>0.8548</u>	0.7965	<u>0.8371</u>
Wan+CN 100% synth	0.8566	0.8501	0.7853	0.8306
VACE Base (Canny)	0.8349	0.8084	<u>0.8068</u>	0.8167
Wan+CN 99/1	0.8497	0.8283	0.7655	0.8145
VACE 99/1	0.8296	0.8203	0.7915	0.8138
DaS 100% synth	0.8168	0.8137	0.7894	0.8066
VACE Base (RGB)	0.8166	0.7957	0.7977	0.8033
Wan+CN Base (Depth)	0.8241	0.8098	0.7646	0.7995
DaS 99/1	0.8052	0.7959	0.7936	0.7982
VACE Base (Depth)	0.8258	0.7642	0.7981	0.7961
Wan+CN Base (Canny)	0.8277	0.7825	0.7704	0.7935
Wan+CN Base (RGB)	0.8190	0.7630	0.7658	0.7826
CAV HED	0.6977	0.7451	0.7760	0.7396
CAV Canny	0.7012	0.7260	0.7659	0.7311
CAV Depth	0.6858	0.7287	0.7730	0.7292
DaS Base	0.7042	0.6787	0.7518	0.7116

Table 3: **VBench Balanced Quality by input category (higher is better).** Each value is the mean over 50 control videos of the category. Balanced Quality is the unweighted arithmetic mean of six VBench dimensions: subject consistency, background consistency, motion smoothness, dynamic degree, aesthetic quality and imaging quality. Overall is the equal-weight mean of the Humanoid, Blocky and Game columns. CAV videos are shorter than five seconds and therefore use standard short-video VBench, whereas all other models use VBench-Long with 2-second in-clip aggregation. **Bold** and <u>underlined</u> values mark the best and second-best result in each column.

Model	Humanoid ↓	Blocky ↓	Game ↓	Overall ↓
C2R 99/1 (ours)	0.0320	0.0453	0.0563	**0.0445**
VACE 100% synth	0.0300	0.0213	0.0994	<u>0.0503</u>
Wan+CN Base (Depth)	0.0664	0.0197	0.0675	0.0512
DaS 99/1	0.0649	0.0961	**0.0149**	0.0586
Wan+CN 100% synth	**0.0080**	0.0617	0.1322	0.0673
VACE Base (Canny)	0.0743	0.0175	0.1126	0.0681
Wan+CN Base (Canny)	0.0689	**0.0016**	0.1464	0.0723
Wan+CN Base (RGB)	0.0668	<u>0.0081</u>	0.1481	0.0743
VACE Base (Depth)	0.0997	0.0247	0.1023	0.0756
Wan+CN 99/1	<u>0.0135</u>	0.0444	0.2050	0.0876
DaS 100% synth	0.1101	0.1305	<u>0.0352</u>	0.0919
CAV Depth	0.0411	0.0886	0.1537	0.0945
CAV Canny	0.0425	0.1219	0.1552	0.1065
VACE 99/1	0.1141	0.0221	0.2221	0.1194
VACE Base (RGB)	0.0992	0.0535	0.2128	0.1218
CAV HED	0.0943	0.1318	0.1471	0.1244
DaS Base	0.1663	0.1937	0.0727	0.1442

Table 4: **RAFT motion-agreement proximity to the calibrated coarse-to-real reference (lower is better).** For each category we select 50 control videos and evaluate every model on the same cases. Values are the calibrated proximity $|\overline{s} - \mu_{\text{matched}}|$, where $\overline{s}$ is the mean RAFT flow-agreement score between each generation and its control video, and μ_{matched} is the reference obtained on our synthetic coarse-to-real pairs, whose motion is identical by construction ($\mu_{\text{matched}} = 0.5888$ for Humanoid, 0.5968 for Blocky and Game; mismatched pairs score 0.4796). The deviation penalizes both under-following and over-copying of the control motion. Overall is the equal-weight mean of the three category proximities. **Bold** and <u>underlined</u> values mark the best and second-best result in each column.

Input type	Failures	Rate
Coarse humanoid	1/50	2.0%
Low-poly blocky	3/50	6.0%
Game capture	0/50	0.0%

Table 5: **Failure rates by input type.** Each generation is rated 1–5 by a vision–language model (Qwen3.6-27B) on five axes, given the coarse control video, 10 frames of the generation and the text prompt: coarse-to-real success, control fidelity, prompt fidelity, temporal coherence and artifact-free quality. A generation counts as a failure if any axis scores ≤ 2, borderline if the minimum is 3, and a success if all axes are ≥ 4. Outcomes were strongly bimodal, with no borderline cases.

A.4.4 Motion Adherence

Table 4 reports RAFT Teed and Deng [2020] optical-flow agreement between each generation and its control video. The score has no natural ceiling at 1.0, since even a perfectly motion-aligned coarse/fine pair differs in texture and silhouette in ways that perturb flow estimation. We therefore calibrate it on our synthetic coarse-to-fine pairs, whose motion is identical by construction and which are the closest ground truth available for the coarse-to-real task (Section A.3). Matched pairs score 0.5888 (humanoid) and 0.5968 (blocky/game), while mismatched pairs score 0.4796. We report the absolute deviation from this reference, which penalizes both under-following and over-copying of the control motion. C2R attains the best overall proximity (0.0445) and, more importantly, is the only method that stays close to the reference across all three input types (0.0320 / 0.0453 / 0.0563). Baselines that match it in one category degrade sharply in another (WAN+CN 99/1 scores 0.0135 on humanoid but 0.2050 on game inputs, and VACE 100% synth. scores 0.0300 vs. 0.0994). This consistency is the more relevant property for our setting, where the method is meant to generalize across human templates and asset formats (Section 1).

A.4.5 Failure Analysis

To quantify failure frequency, each generation (50 per input category) is assessed by an open-weight vision–language model (Qwen3.6-27B Qwen Team [2026]) given the coarse control video, 10 frames of the generated video and the text prompt. The model assigns independent 1–5 ratings on five axes, which are coarse-to-real success (whether crude CG becomes convincing real imagery rather than retaining synthetic geometry or appearance), control fidelity (subject count and identity, layout, pose, trajectory and camera motion), prompt fidelity, temporal coherence (flicker, identity drift, deformation and temporal collapse), and artifact-free quality (malformed anatomy, warped or duplicated objects, blur). Scores of 1–2 denote serious failure, 3 usable but clearly problematic, and 4–5 minor or no problems. A generation counts as a failure if any axis scores ≤ 2, borderline if the minimum is 3, and a success if all axes are ≥ 4. Outcomes were strongly bimodal, with no borderline cases. Table 5 reports the failure rates per input type, and Figure 11 shows one representative case per input category.

A.4.6 On the Joint Reading of VE-Bench and VQAScore

Both metrics penalize correct behaviour in our setting, for structurally opposite reasons. VQAScore measures prompt–video alignment, but in coarse-to-real generation the prompt governs appearance (clothing style, time of day, weather) while the coarse input governs layout (camera path, building footprints, where and how many people walk). If the prompt asks for a crowded market street and the control video specifies three people on a wide avenue, faithful control necessarily makes the video less prompt-matching, and a model that ignores the control signal scores higher. Table 2 shows this directly, since the highest VQAScores belong to the configurations with the weakest structural correspondence (VACE 100% synth. reaches 0.7126 VQAScore vs. 0.1488 VE-Bench, and DaS 100% synth. reaches 0.7566 vs. -0.0805). VE-Bench penalizes the other half. It favors conservative edits that closely preserve the source, so its highest values (100% real at 0.8090, 10 heads at 0.7893) belong to the configurations most strongly influenced by the control signal, which in both cases degrades the output rather than improving it. For a coarse input the correct output must diverge from the source in appearance while preserving global structure, so the top of the VE-Bench range is not attainable by a model performing the task properly.

The joint ceiling therefore lies well below $(1, 1)$, and neither metric tells us where. More fundamentally, the two metrics are not measures of independent axes. VE-Bench conflates preserving structure with not changing appearance, while VQAScore conflates realizing the prompt with ignoring the con-

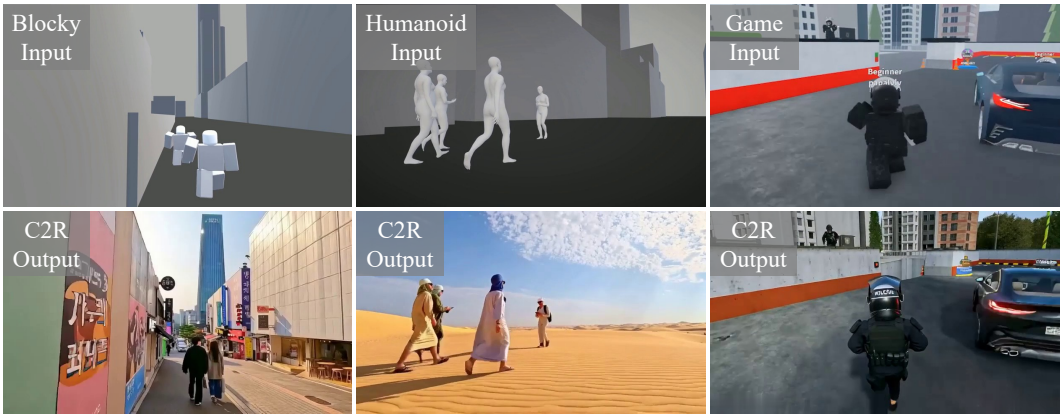

Figure 11: **Representative failure cases per input type.** One example per input category, judged with the protocol of Section A.4.5. Left, low-poly blocky input. With several very coarse characters in the scene, the model can confuse their depths and render two characters that walk at different distances side by side. Middle, coarse humanoid input. The text prompt can override the structure. Here a prompt that says nothing about the buildings replaces the background with a desert, although the people still follow the motion of the control. Right, game capture. The judge flagged no failure in this category, but non-humanoid characters can yield flatter bodies and a slightly altered perspective, and their coarse motion may not transfer exactly, as with the legs here, which are in a different gait phase than the input. All three cases fail on control fidelity. Failure rates are reported in Table 5.

trol input, which is why the frontier in Figure 3 looks like a trade-off. A metric that scores adherence to the control input separately from appearance quality and prompt realization, so that both can be high at once, is beyond the scope of this work and we leave it as future work. We also note that the 99%/1% ratio could be specific to our setup and that other coarse-to-real or cross-domain tasks would likely land elsewhere. What we expect to transfer is the finding that a small, carefully constructed paired synthetic set, combined with large amounts of available unpaired data, suffices to enable a task for which paired supervision does not otherwise exist, provided the control representation lets both domains share a feature space.

A.5 Extra Experiments

A.5.1 Additional Comparison Examples

To further evaluate the performance of C2R, we provide an extended qualitative comparison in Figure 12 using coarse humanoid control inputs instead of low-poly block characters. Following the same evaluation methodology as in Section 4.3, we compare C2R against several recent controllable video generation baselines, including WAN VACE, WAN+ControlNet, Diffusion as Shader (DaS), Control-A-Video, and Seedance 2.0, under the same target prompt and control setting.

Similarly to the block-based comparison, the evaluated methods rely on different conditioning modalities to guide generation. WAN VACE uses depth-conditioned video generation, whereas WAN+ControlNet corresponds to our finetuned WAN-ControlNet baseline trained on the same synthetic paired data as C2R, enabling direct conditioning on the humanoid control sequence and text prompt. DaS requires tracking supervision computed with SpaTracker together with a reference image; we evaluate both DaS-FirstFrame, where the reference frame is generated using FLUX following the original DaS pipeline, and DaS-NoFirstFrame, which uses the first frame of the control sequence as reference. Control-A-Video, like WAN VACE, uses depth-conditioned video generation. Seedance 2.0 is also evaluated using the same control video and prompt inputs.

As illustrated in Fig. 12, the humanoid control setting is significantly easier than the highly coarse low-poly block scenario, since the input already provides more human-like structural priors and motion cues. Nevertheless, important differences between methods remain apparent. Depth- and tracking-based approaches generally maintain the input motion structure and coarse character consistency, although they can still fail to preserve fine identity details across the sequence. Their outputs often remain visually conservative, with limited photorealism and weak environmental dynamics, where backgrounds remain largely static despite foreground motion. While reference-image-based methods improve visual fidelity, they still inherit many of the limitations of depth- and tracking-conditioned generation, resulting in constrained scene evolution and less expressive video synthesis.

WAN+ControlNet improves overall structural and temporal adherence compared to off-the-shelf controllable baselines, but still produces visually conservative outputs with limited realism and detail synthesis. Seedance 2.0 generates visually appealing frames, yet frequently fails to accurately follow the intended camera trajectory and motion structure.



Figure 12: **Baseline comparison on coarse humanoid character control.** We compare C2R against WAN VACE, WAN+ControlNet, Diffusion as Shader (DaS), Control-A-Video, and Seedance 2.0 under the same target prompt and humanoid control setting. While the humanoid input provides stronger structural priors than the low-poly block scenario, existing baselines still struggle to accurately preserve motion structure, camera dynamics, and temporal coherence across the generated sequence. In contrast, C2R achieves the best balance between structural controllability and photorealistic synthesis, generating realistic humans and environments while faithfully following the input motion and layout. The bottom row shows zoomed crops of the windows marked on the top row. C2R inpaints the hairstyle requested in the text prompt and adds accessories such as umbrellas when the prompt specifies them, whereas WAN VACE and WAN+ControlNet reproduce the bald, tightly bounded silhouette of the coarse input.

In contrast, C2R successfully preserves the humanoid motion, camera dynamics, and scene layout while generating realistic humans, detailed clothing, rich urban textures, and coherent lighting conditions, even hallucinating fine contextual details such as umbrellas that are not explicitly provided in the coarse control input. Compared to the low-poly block experiments, all methods benefit from the more informative humanoid control input; however, C2R consistently achieves the best balance between structural controllability and photorealistic synthesis. These results further support the effectiveness of our synthetic-real domain-hedging strategy across varying levels of control signal abstraction.

A.5.2 Structural Fidelity versus Realism

Close adherence to the input geometry is not the objective of C2R, which instead seeks plausible appearance and fine-scale dynamics consistent with the structure the input specifies. Hair and loose garments are the clearest case. Since the coarse figures carry neither (Section A.3), closely reproducing their silhouette means reproducing a bald, tightly bounded figure, whereas a coat or flowing hair must extend beyond the input silhouette and move with secondary dynamics the control signal does not contain, as in Figure 12, where the model adds umbrellas absent from the coarse input. Adherence and reconstruction lie on one axis, and the dense per-frame conditioning of WAN+ControlNet sits near the structural-reconstruction end. The zoomed crops in Figure 12 pair the control signal with the corresponding generations of WAN VACE, WAN+ControlNet and C2R so that the structural-fidelity/realism trade-off can be inspected directly.

A.5.3 Structural Drift and Control Capacity

Why higher synthetic ratios cause structural drift. As described in Section A.3, our synthetic pairs are aligned in motion, camera and pose, but deliberately not in structure, since the coarse render omits the clothing, hair and façade detail present in the target. Each pair therefore teaches a large coarse-to-fine delta, and this is the point of the method. Hair and loose garments with secondary dynamics, and detailed façades, are among the most expensive elements to author, simulate and render in traditional pipelines, and C2R lets the user provide only minimal 3D and delegate that detail

to the renderer. Real data carries the opposite signal. In Stage II the control branch and the target are the same video, differing only by the HSV transformation (Section A.1.3), so the structural delta is zero and the adapter is supervised toward tight correspondence. The mixing ratio therefore sets how much deviation the adapter learns to treat as expected. As the synthetic fraction rises, the average supervised delta grows and the model invents structure beyond what appearance detail requires, at the cost of alignment with the control input, which is the drift visible in Figure 4.

Why additional control capacity produces reconstruction-like outputs. On a coarse render the target is underdetermined, so part of the loss is irreducible however expressive the adapter, whereas on real data, which forms 99% of Stage II batches, the control branch and the target are the same video and structure is exactly recoverable on almost every batch. Added capacity is therefore optimized toward structural copying rather than toward the coarse-to-real mapping, so geometry, silhouettes and layout become pinned to the control input while colors and textures continue to be synthesized from the prompt. What is reconstructed is structure, not appearance. HSV decorrelation cannot prevent this, because it closes the appearance channel but deliberately leaves structure intact, as structure is the signal the control branch exists to carry, and what leaks is geometric rather than chromatic.

A.5.4 Adaptive Coarse-to-Fine Control

Fig. 13 shows that our method adapts control signals at different levels of geometric detail, from very coarse to relatively fine inputs. In all cases, the model respects the provided structure and adapts the generated content to the fidelity of the input geometry. Both examples are generated under a prompt that explicitly specifies the appearance of the pedestrians (curly afro hair and a long leather jacket). The control video sets layout, body motion and camera, while hairstyle and clothing follow the text, which separates input-controlled structure from text-controlled appearance.

APG versus CFG. Adaptive Prompt Guidance Castillo et al. [2025] permits higher guidance scales than classifier-free guidance (CFG) without the color oversaturation CFG produces at comparable strength. The middle and right columns of Figure 13 are generated with the same guidance scale of 15, the same prompt and the same seed, with only the guidance method changed. CFG oversaturates colors and lighting, whereas APG yields more natural results while still enforcing both conditioning signals, the text prompt and the control video.



Figure 13: **Coarsening levels for driving signal, with APG versus CFG.** Given very coarse geometry (top), our model inpaints many details. Given fine geometry (bottom), it successfully follows the richer input signal. Both rows use the same control signals and a prompt specifying curly afro hair and a long leather jacket. The middle and right columns are generated with APG and with CFG at the same guidance scale of 15 and the same seed. CFG oversaturates colors and lighting, whereas APG does not.

A.5.5 Applications

Our framework is agnostic to the specific human and scene templates used in the control video and generalizes well beyond the training distribution. By operating on coarse 3D renderings and applying HSV augmentation during training, the model reduces dependence on appearance cues such as color, texture, and rendering style. This allows robust handling of inputs from different game engines, simulation pipelines, and lighting conditions while preserving camera motion and human trajectories.

As shown in Fig. 14, our method transforms low-poly game videos into realistic outputs while faithfully following the input motion and interactions. Notably, the input character is cartoon-

styled and visually far from the neutral human models used during training, yet the generated video reproduces the same running and jumping motion with improved realism. Visual collision artifacts present in the low-poly input are also mitigated, resulting in more plausible contact between the character and the environment.



Figure 14: **Turn a low-poly Roblox game video into real-style.** *Left:* The character runs forward and jumps over a low wall. *Right:* The character throws a bomb and runs away.

A.5.6 Illustrating Limitations

Fig. 15 illustrates how our method can fail to maintain structural integrity when relying on extremely sparse or abstract 3D inputs. While these coarse inputs are intended to guide scene layout, camera motion, and human dynamics, a lack of sufficient geometric detail can result in a loss of precise control over the desired layout or trajectory. Failure rates and representative failure cases per input type are given in Section A.4.5.

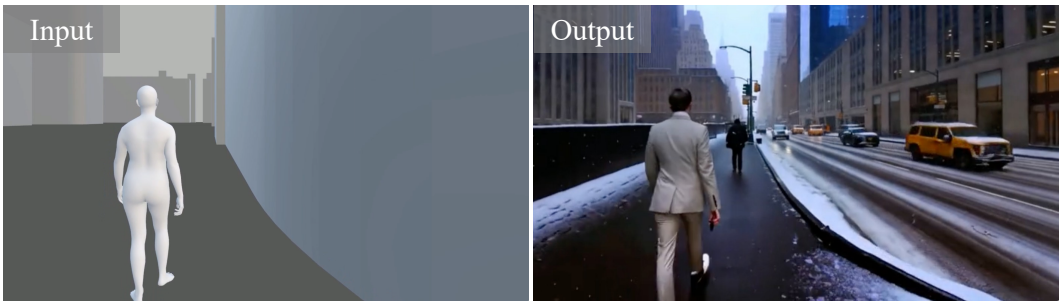

Figure 15: **Limitation.** Very coarse input 3D structure might not be sufficient to guide the architecture and city layout as expected.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The Abstract and Introduction (Section 1) explicitly state our core contributions, including the proposed generative rendering framework, the two-stage synthetic-real domain-hedging training strategy, and the system capabilities, accurately reflecting the scope of the evaluation presented in the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We dedicate a section to Limitations (Section 5) and future work where we explicitly discuss failure cases related to extremely sparse input geometry and the lack of real-time generation capabilities due to the non-autoregressive nature of the framework.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: Our paper focuses on empirical system design and methodology for generative rendering and does not introduce any new theoretical results or formal proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide comprehensive details regarding our architecture, training stages, objective functions, and evaluation metrics in the Method 3 and Experiments sections. Furthermore, as stated in the abstract, we will release the pre-trained model weights to allow the community to reproduce and verify our generation results, even though the raw training data remains internal.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release the code upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All the details are specified in section 3 Method and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Our section 4 (Experiments) demonstrates significant improvement on control and improved video quality compared to state-of-the-art.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide computer resources requirements and details in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper does not introduce new societal impact compared with baseline models.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: The base model weights (Wan 2.1) are released with their own safeguards and usage restrictions. In our training data preparation, we filtered out NSFW prompts and unsafe content. Any released code or model components will be accompanied by usage guidelines that prohibit deceptive, impersonation, privacy-violating, or harmful applications. We will also provide recommended negative prompts and content-filtering guidance to reduce unsafe generations.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all models and datasets mentioned in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not introduce new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: Our research relies entirely on automated quantitative metrics and does not involve any crowdsourcing or experiments with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

Justification: This study does not involve human subjects or study participants, making IRB approval not applicable to our work.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes].

Justification: We explicitly describe our use of the Tarsier (Qwen) large multimodal model for automated dataset captioning in the appendix, as well as the use of a pretrained T5 encoder for text conditioning in the method section. We also use an open-weight vision–language model (Qwen3.6-27B) as an automatic judge in the failure analysis of Appendix A.4.5, with the full rating rubric described there.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.