PAIR-FREE COARSE-TO-REAL VIDEO GENERATION VIA REAL-TO-COARSE IMAGE SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Coarse-to-real (C2R) video generation converts coarse animation, simulation, or game-rendered videos into photorealistic videos while preserving scene layout, camera motion, and dynamics. Training such models requires aligned (coarse, realistic) video pairs, which are expensive to create and confine prior systems to narrow synthetic domains. We exploit a simple asymmetry: paired videos are hard to obtain, but paired real-to-coarse images are easy to construct with existing editing and 3D reconstruction tools. Using these image pairs, we train the DINO-R2C model to transfer realistic images to coarse images in DINO feature space, and apply it frame-wise to ordinary real videos to synthesize pseudo-coarse videos. The C2R# model then learns to generate each real video from its pseudo-coarse counterpart, free from needing (coarse, realistic) video pairs. At inference, the DINO-R2C model is discarded and actual coarse videos drive the C2R# model directly. Thus, our pair-free training scales with any real-video corpora rather than with paired synthetic videos.

1 INTRODUCTION

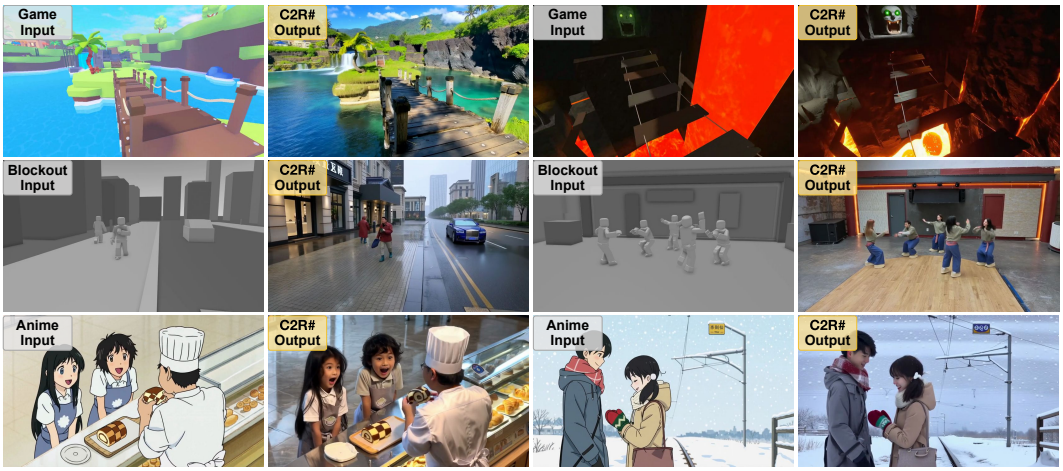

Figure 1: The C2R# model transforms game renders (top), geometry blockouts (middle), and stylized animations (bottom) into photorealistic videos.

Coarse-to-real (C2R) video generation uses coarse videos, such as game renders, geometry blockouts, or stylized animations, as the structural control for photorealistic video generation. The coarse input specifies scene layout, camera motion, and dynamics, while a modern video model produces the final visual appearance.

A major limitation of existing coarse-to-real systems is that they need paired training data. The original C2R paper (Gomez-Nogales et al., 2026) introduces a model that learns controllability from paired synthetic (coarse, realistic) videos. Such aligned video pairs are expensive to create, and the cost multiplies with every additional coarse modality, so paired-video supervision is difficult to scale.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Our key observation is that this bottleneck is asymmetric. Paired coarse-to-real videos are expensive, but paired real-to-coarse images are cheap. Modern image editing models and 3D fitting methods can turn diverse realistic images into aligned coarse counterparts. Image-level supervision is therefore highly accessible. However, using an image generator such as GPT-Image 2 (OpenAI, 2026) to edit every frame of a large video corpus into its coarse counterpart would incur substantial generation costs.

We exploit this asymmetry with an image-level real-to-coarse bridge, the DINO-R2C model. Trained on (coarse, realistic) image pairs, it predicts the DINO features (Siméoni et al., 2025) of the coarse counterpart of a realistic image. Applied to every frame of an ordinary real video, it synthesizes a pseudo-coarse video that is paired with its source. We train the C2R# model to generate the real video from this pseudo-coarse control video. At inference, the DINO-R2C model is discarded and an actual coarse video drives the C2R# model directly.

Using the image-level bridge poses two challenges. First, the DINO-R2C model is an image model, so its per-frame outputs are not temporally smooth. As a result, we apply camera augmentation during its training to reduce this jitter, and the C2R# model learns to tolerate the rest because its target is always the temporally coherent real video. Second, the predicted features can leak cues of the realistic input that no actual coarse video contains. Therefore, we block the leak by decoding the prediction to pixels with a frozen RAE decoder (Zheng et al., 2026) and re-encoding those pixels before they serve as control.

This formulation moves the expensive part of supervision from videos to images. Once the DINO-R2C model is trained for a coarse modality, any real-video collection becomes coarse-to-real training data, while supporting a new modality requires only new image pairs rather than a new paired-video dataset.

2 Related Work

Generative rendering from coarse structure. Turning cheap renders into realistic imagery predates diffusion models: Richter et al. (2022) enhance game frames with a G-buffer-conditioned network trained adversarially against unpaired photographs, and vid2vid (Wang et al., 2018) learns paired label-to-video synthesis. Recent work conditions image or video diffusion models on G-buffers (Beisswenger et al., 2025; Zeng et al., 2026), animated meshes (Cai et al., 2024), physics-engine states (Lin et al., 2026), or multimodal spatial maps (Abu Alhaija et al., 2025), and RealMaster (Cohen-Bar et al., 2026) lifts rendered videos to photorealism by propagating enhanced keyframes to build its own paired videos. NVIDIA's DLSS 5 (NVIDIA, 2026) conditions a real-time generative renderer on rendered frames and engine motion vectors to enhance photorealistic appearance. All of these obtain supervision from inside a rendering pipeline, from per-scene synthesis, or from authored video pairs. We assume only a coarse RGB video at test time and no paired videos at all.

The original C2R paper. Our direct predecessor (Gomez-Nogales et al., 2026) conditions a Wan2.1 backbone on DINOv3 features of the control video through a lightweight adapter. Its control pathway is trained mostly on unpaired real videos reconstructed from their own DINO features, with a small set of paired synthetic (coarse, realistic) videos anchoring controllability. Controllability therefore rests on scarce, domain-limited video pairs, and the adapter sees mostly features of real videos in training but features of coarse videos at test time. Our pair-free C2R framework keeps the objective and control interface but removes paired videos from training.

Controllable video diffusion. Video diffusion and flow-matching models can be guided by depth, edges, segmentation, or sparse keyframes through ControlNet-style branches (Zhang et al., 2023; Esser et al., 2023; Guo et al., 2024; Lin et al., 2025), by unified context blocks (Jiang et al., 2025; Wan, 2025), or by warped noise that carries motion (Burgert et al., 2025). For image generation, LooseControl (Bhat et al., 2024) generalizes ControlNet's depth conditioning to loosely specified scene boundaries and 3D object boxes. Instruction-based video editors such as Editto (Bai et al., 2025) are trained on large synthetic edit datasets and can perform sim-to-real edits, which is why we compare against them. Explicit structural controls preserve geometry strongly but can over-constrain generation or leak the appearance of the control input; as in the original C2R paper, the C2R# model

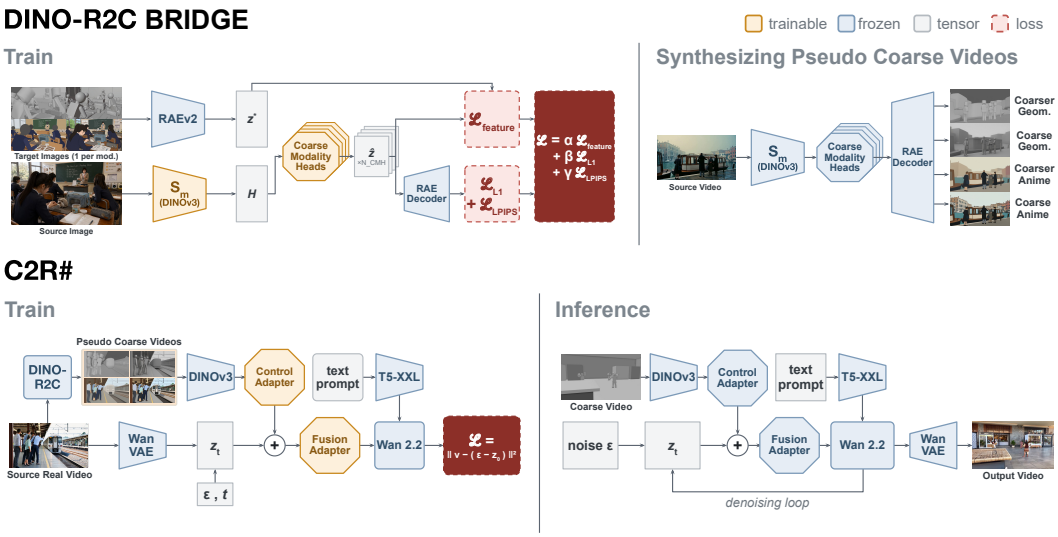

Figure 2: **Overview of the DINO-R2C and C2R# models. Top:** The DINO-R2C model is trained from paired realistic and coarse images using feature reconstruction and pixel-space reconstruction losses. Once trained, it is applied to real videos frame-wise to synthesize pseudo-coarse videos. **Bottom left:** The C2R# model is trained on these pseudo-paired videos. The pseudo-coarse video is encoded by the frozen DINO encoder E and injected into the frozen Wan2.2 backbone through trainable control and fusion adapters, while the original real video provides the training target. **Bottom right:** at inference, an actual coarse video is encoded directly by E and used to condition the same C2R# model; the DINO-R2C model is no longer required. Orange modules are trainable, blue modules are frozen, gray boxes are tensors, and red boxes denote training losses.

conditions on implicit DINO features, which carry layout and semantics while leaving appearance to the video model.

Self-supervised features as a cross-domain bridge. DINO features (Oquab et al., 2024; Siméoni et al., 2025) are dense and robust to appearance change, and several works exploit this for domain transfer. Concurrently, Control-DINO (Dominici et al., 2026) conditions an image-to-video model on DINOv3 features for stylization and video-from-3D, and Liu et al. (2026) use self-supervised features as an appearance-invariant bridge for unpaired image translation. Earlier unpaired video translation relied on cycle consistency (Zhu et al., 2017) with pseudo-supervision from synthetic motion (Wang et al., 2022), while InstructPix2Pix (Brooks et al., 2023) and Ditto (Bai et al., 2025) showed that strong image models can synthesize the paired data an editor needs. We combine both ideas: image models provide cheap (coarse, realistic) image pairs, an real-to-coarse model learned in DINO space and decoded through an RAE decoder (Zheng et al., 2026) lifts them to videos, and the resulting pseudo-paired videos supervise the coarse-to-real generator.

3 Method

3.1 Problem Setup

At inference time, the user provides an actual coarse video v^{c} from categories such as game renders, geometry blockouts, or stylized animations, together with a text prompt. We seek a generator that produces a realistic video while preserving the scene layout, camera motion, and dynamics of v^{c}. For training, we assume access to ordinary real videos v^{r} and to aligned (coarse, realistic) *image* pairs, but to no paired $(v^{\mathrm{c}}, v^{\mathrm{r}})$ videos. Throughout, superscripts r and c denote realistic and coarse data.

Our pair-free C2R framework has two models (Figure 2). The DINO-R2C model (Section 3.2) learns real-to-coarse image transfer from the image pairs and is then applied to every frame of the real videos, producing pseudo-paired videos (Section 3.3). The C2R# model (Section 3.4) is trained

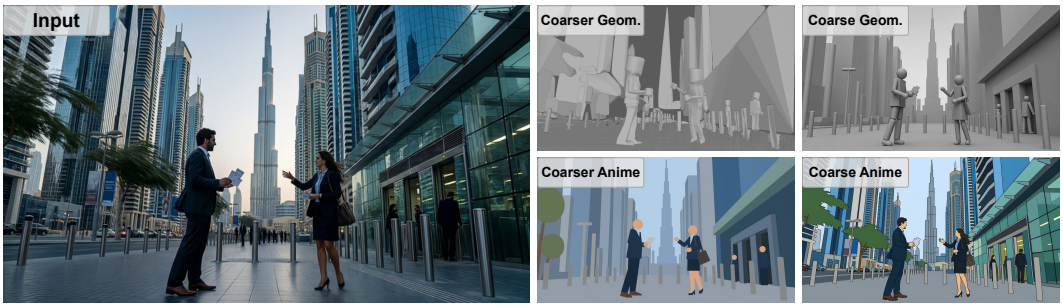

Figure 3: Example (coarse, realistic) image pairs. A realistic image (left) is paired with coarse images in four modalities (right): Coarser geometry, Coarse geometry, Coarser Anime, and Coarse Anime.

on these pseudo-paired videos to generate the real video from its pseudo-coarse counterpart; at inference it is driven directly by actual coarse videos.

3.2 Image-Level Real-to-Coarse Bridge: the DINO-R2C

Image pairs. We define a *coarse modality* as one visual form of coarse input that the C2R# model should support, and use the four modalities shown in Figure 3. Each coarse image preserves the subjects, layout, pose, and camera of its realistic counterpart. We generate realistic source images with GPT-Image 2 (OpenAI, 2026) and create their coarse counterparts through instruction-guided editing with the same model or 3D reconstruction and simplification. All images are produced at 1280×720, with 14,000 aligned (coarse, realistic) image pairs per modality (Appendix A.1). These image pairs are the only paired supervision in our framework.

Model and objective. Let E be a frozen DINO encoder (DINOv3-L/16 (Siméoni et al., 2025)) that maps an image to a dense feature grid. For each coarse modality m, the DINO-R2C model learns a mapping S_m from a realistic image to the DINO features of its coarse counterpart,

$$S_m(I^{\mathrm{r}}) \approx E(I^{\mathrm{c}}_m), \qquad S_m = H_m \circ B, \tag{1}$$

where B is a trainable DINOv3-L backbone shared across all four modalities and H_m is one of four modality-specific residual MLP heads. We learn the mapping in DINO feature space rather than in RGB because the self-supervised representation captures scene structure and layout while being relatively insensitive to the large appearance gap between realistic and coarse images; empirically, training also converges more easily in DINO space. To supervise the prediction in pixel space as well, a frozen RAE decoder D (Zheng et al., 2026) decodes it to a coarse image $\hat{I}^{\mathrm{c}}_m = D(S_m(I^{\mathrm{r}}))$. The training objective is

$$\mathcal{L}_{\mathrm{R2C}} = \lambda_{\mathrm{feat}} \left\| S_m(I^{\mathrm{r}}) - E(I^{\mathrm{c}}_m) \right\|_2^2 + \lambda_{\mathrm{rgb}} \left\| \hat{I}^{\mathrm{c}}_m - I^{\mathrm{c}}_m \right\|_1 + \lambda_{\mathrm{LPIPS}} \, \mathrm{LPIPS}\big(\hat{I}^{\mathrm{c}}_m, I^{\mathrm{c}}_m\big), \tag{2}$$

a DINO feature reconstruction loss, an RGB reconstruction loss, and the LPIPS perceptual loss (Zhang et al., 2018), with $\lambda_{\mathrm{feat}} = 1.0$, $\lambda_{\mathrm{rgb}} = 1.1$, and $\lambda_{\mathrm{LPIPS}} = 0.82$.

Camera augmentation. The DINO-R2C model is trained on image pairs but applied to every frame of a video, where consecutive frames differ by small viewpoint changes. To close this image-to-video gap, we apply the same random camera homography (crop, translation, rotation, shear, and perspective) to a realistic image and its coarse counterpart during training, so that the learned mapping is stable under the viewpoint changes seen between frames. Section 4.2 ablates the augmentation strength, and Appendix A.3 gives the exact bounds.

3.3 Constructing Pseudo-Paired Videos

Applying the DINO-R2C model to every frame of a real video v^{r} yields a pseudo-coarse video that remains paired with v^{r}. Because each frame is synthesized independently, the pseudo-coarse video carries some temporal jitter. We use it only as the control video and always use the temporally

coherent real video as the target, which teaches the C2R# model to tolerate minor jitter in the control video while generating temporally consistent output.

Decoding to RGB and re-encoding. The features $S_m(v^{\mathrm{r}})$ predicted by the DINO-R2C model already live in DINO space and could condition the C2R# model directly. However, they retain hidden cues of the realistic input that are absent when an actual coarse video is encoded at inference, a leakage that creates a train/test mismatch (Section 4.3). We therefore decode the prediction to coarse pixels with the frozen RAE decoder, which discards information that cannot be expressed as a coarse image, and re-encode those pixels with the same frozen encoder E used at inference. The training control features are thus

$$C_{\mathrm{train}} = E\big(D(S_m(v^{\mathrm{r}}))\big), \tag{3}$$

computed per frame and on the fly during C2R# training.

3.4 Coarse-to-Real Video Generation: the C2R#

Architecture. The C2R# model builds on a frozen pretrained text-to-video diffusion backbone, Wan2.2 T2V-A14B (Wan, 2025), whose VAE, text encoder, and denoiser all remain frozen. The denoiser is a mixture of two experts routed by diffusion timestep, so we train one control adapter per expert, active only when its expert is. Each control adapter takes the per-frame DINO control features C, projects them to an intermediate width, applies residual space-time convolution blocks with timestep conditioning, compresses time to the VAE latent frame rate, and maps the result to the denoiser width through a zero-initialized output head. At our operating resolution the DINO feature grid aligns one-to-one with the backbone's spatial token grid, so no learned spatial upsampling is needed. A zero-initialized fusion adapter then adds the control features after the frozen patch embedding, so at initialization the model is identical to the pretrained backbone. Architectural and training details are given in Appendix A.5.

Training objective. For a target real video v^{r} with latent $z_0 = \mathrm{VAE}(v^{\mathrm{r}})$, we sample a flow timestep σ and Gaussian noise ϵ, form $z_\sigma = (1-\sigma)z_0 + \sigma\epsilon$, and minimize the rectified-flow velocity loss (Liu et al., 2023)

$$\mathcal{L}_{\mathrm{FM}} = \big\| G\big(z_\sigma, \sigma, e_{\mathrm{text}};\, A(C_{\mathrm{train}}, \sigma)\big) - (\epsilon - z_0) \big\|_2^2, \tag{4}$$

where G is the frozen denoiser, e_{text} is the text embedding, and A denotes the trainable adapters, the only parameters that are updated. We use no control dropout: with a frozen backbone, steps without control provide no gradient to the adapters.

Inference. The user provides an actual coarse video v^{c} and a text prompt. The frozen encoder produces $C_{\mathrm{test}} = E(v^{\mathrm{c}})$, which conditions the same adapters and backbone used in training; generation starts from Gaussian noise, and the frozen VAE decodes the result. Neither the DINO-R2C model nor the RAE decoder is used at inference.

4 Experiments

We evaluate the DINO-R2C model (Section 4.2) and the C2R# model (Sections 4.3 and 4.4). Because real-world coarse videos have no photorealistic ground truth, the C2R# model is evaluated with reference-free quality, control-alignment, and VLM-judged realism metrics on three evaluation sets of 100 actual coarse videos each: game renders, geometry blockouts, and stylized animations. We report aggregate results across all 300 videos. Dataset construction and metric definitions are given in Appendices B.1 and B.3.

4.1 Training data.

We train a shared four-modality DINO-R2C model on 14,000 image pairs per modality: coarser geometry, coarse geometry, coarser anime, and coarse anime. GPT-Image 2 (OpenAI, 2026) generates the realistic images and edits them into coarse geometry and both anime modalities; coarser geometry targets are rendered from reconstructed and simplified 3D scenes. Images are generated at

Table 1: **DINO-R2C design ablations, video stability.** Evaluated on 1,000 real videos in the Coarse geometry modality at 256×256. Columns report means across videos; both flicker measures are per-video medians relative to source-video error. Bold/underlined values are best/second-best. Metrics are explained in Section 4.2 and Appendix B.3.

DINO-R2C Recipe	Perceptual flicker (relative) ↓	Smoothed-frame flicker (relative) ↓	Motion error (EPE) ↓	Motion magnitude ratio ≈ 1	Boundary stability (Dice) ↑	Content consistency (DINO) ↑
None, no LPIPS	2.2028	4.1565	2.2685	2.1909	0.7195	0.9816
None + LPIPS	2.1793	3.9131	1.8678	1.8304	0.7271	0.9877
$0.5\times$ + LPIPS	0.9314	1.7901	0.8819	<u>1.0844</u>	0.8345	0.9946
$1\times$, no LPIPS	**0.7965**	<u>1.6825</u>	0.9066	1.1504	0.8327	0.9914
$1\times$ + LPIPS	0.9023	1.7108	<u>0.8695</u>	1.0875	<u>0.8354</u>	<u>0.9946</u>
$2\times$ **+ LPIPS (ours)**	<u>0.8649</u>	**1.6294**	**0.8429**	**1.0797**	**0.8376**	**0.9948**

1280×720 using the procedure in Appendix A.1; larger training sets yielded only marginal improvements. For C2R# training, the DINO-R2C model synthesizes pseudo-coarse controls on the fly for 389,854 real videos from DL3DV (Ling et al., 2024), `SEKAI_HQ`, and `SEKAI_DRONE` (Li et al., 2026b). The adapted VACE 2.2 (Jiang et al., 2025) and Editto Sim2Real (Bai et al., 2025) baselines use exactly the same real videos and pseudo-coarse controls (Appendices A.4 and B.2).

4.2 DINO-R2C

Protocol. We evaluate each DINO-R2C model on 1,000 held-out images and 1,000 real videos. Because each coarse modality has a different target distribution, values are not comparable across modalities. All directly comparable ablations target the Coarse geometry modality and vary the camera-augmentation strength (none, $0.5\times$, $1\times$, $2\times$) and the inclusion of LPIPS (Table 1; image ablations are in Appendix B.5, Table 3). Image metrics compare predictions with the paired coarse target. Video metrics assess the stability of the pseudo-coarse video produced by applying the model to each frame of a real video.

Video stability metrics. We estimate motion with RAFT-Small (Teed & Deng, 2020). Perceptual flicker measures motion-aligned appearance changes using LPIPS (Zhang et al., 2018); smoothed-frame flicker measures broad color and shape changes using L1 on blurred frames. Both are divided by the corresponding source-video error, summarized by the median within each video, and averaged across videos. Motion error measures disagreement between estimated source and output motion vectors, while motion magnitude ratio indicates whether the output moves too little or too much. Boundary stability measures the overlap of motion-aligned outlines, and content consistency measures similarity between consecutive DINOv3 frame features (Siméoni et al., 2025). Appendix B.3 explains all six metrics and their interpretation.

Results. In Table 1, the selected recipe achieves the lowest smoothed-frame flicker and motion error, highest boundary stability and content consistency, and motion magnitude ratio closest to one. It ranks second in perceptual flicker. Its motion magnitude ratio of 1.0797 indicates approximately 8% higher estimated motion magnitude than the source; combined with the lowest motion error, this supports stronger source-motion agreement among the evaluated variants.

With LPIPS enabled, increasing augmentation from $1\times$ to $2\times$ improves all six reported video metrics. At fixed $1\times$ augmentation, adding LPIPS improves image perceptual reconstruction, video motion agreement, motion-magnitude preservation, boundary stability, and content consistency, but increases both relative flicker errors. These results support selecting $2\times$ augmentation with LPIPS as a balance between paired reconstruction fidelity and temporal behavior during per-frame video inference, rather than as the optimum for every individual metric. Image reconstruction results and their tradeoffs are reported in Appendix B.5.

Qualitative results. Image examples appear in Appendix B.5 (Figure 8); pseudo-coarse videos and additional image-pair results are provided in the supplemental HTML page.

Table 2: Comparison on 300 control videos: 100 game renders, 100 geometry blockouts, and 100 stylized animations. Every score averages 300 valid videos per method. "Rep." denotes Editto Sim2Real and VACE 2.2 fine-tuned on the same pseudo-paired training videos as C2R# (ours). Bold/underlined values rank first/second; arrows indicate preferred directions. Metric abbreviations and scales appear in Appendix B.3.

	DOVER		VEFX			VBench	Control alignment			VLM
Method	Tech. ↑	Aesth. ↑	RQ ↑	IF ↑	EE ↑	BG ↑	DINO ↓	Flow ↑	Cam. ↓	Realism ↑
C2R orig.	0.2647	<u>0.8033</u>	<u>3.5082</u>	2.8467	2.3468	0.9259	0.0226	0.6041	0.1733	**75.50**
Editto Global	<u>0.2648</u>	0.7700	3.4876	<u>2.9035</u>	2.3637	0.9338	0.0189	0.6897	0.1634	66.77
Editto Sim2Real	0.2288	0.7124	2.6722	1.4108	1.6995	0.9254	0.0196	0.5986	0.1788	52.62
Editto Sim2Real (rep.)	0.2504	0.7589	3.1011	1.7402	2.2450	0.9283	**0.0111**	0.7314	0.1574	58.52
VACE 2.2	0.2599	0.7768	3.4110	2.5446	2.3182	**0.9354**	0.0135	**0.7466**	**0.1491**	61.88
VACE 2.2 (rep.)	0.2487	0.7117	2.9695	2.4918	<u>2.5334</u>	0.9272	<u>0.0114</u>	<u>0.7347</u>	<u>0.1549</u>	51.96
C2R# (ours)	**0.2841**	**0.8169**	**3.5679**	**3.1823**	**2.5923**	<u>0.9343</u>	0.0147	0.6588	0.1600	<u>74.17</u>

4.3 C2R# Ablations and Efficiency

Effect of the RGB bottleneck. Without RAE decoding, C2R# model is trained directly on DINO-R2C model features predicted from real videos. These features may carry hidden realistic information that is unavailable when actual coarse videos are encoded at inference, creating a training–inference distribution shift. Decoding through the RAE acts as an information bottleneck that removes such hidden cues before the coarse pixels are re-encoded by DINO. Figure 4 shows the failure without this bottleneck. This illustrative variant is excluded from quantitative comparisons.

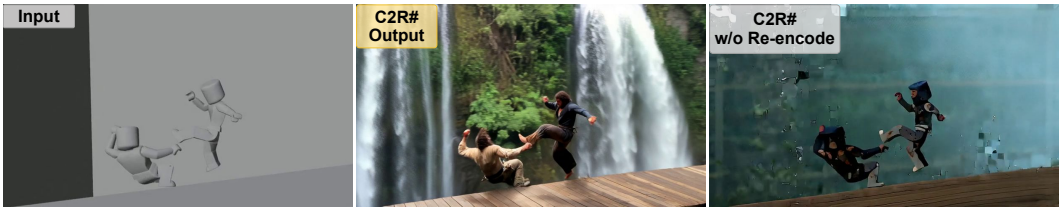

Figure 4: **Effect of the RGB bottleneck during training.** Matched frames show the control input (left), C2R# (center), and a variant conditioned directly on predicted DINO-R2C features, bypassing decoding and re-encoding (right). The latter produces visible artifacts.

Efficiency. With Wan2.2 frozen, the C2R# model trains only 180.54M control and fusion adapter parameters (0.63% of the counted model parameters). Appendix B.6 reports parameter comparisons (Table 4) and DINO-R2C training/inference costs (Table 5).

4.4 Comparisons

Protocol. We compare the C2R# model against the model from the original C2R paper, Editto Global, the released Editto Sim2Real, VACE 2.2, and versions of Editto Sim2Real and VACE 2.2 fine-tuned on exactly the same pseudo-paired training data as the C2R# model. All methods use the same 300 actual coarse videos and scene-specific text prompts. Table 2 reports DOVER technical and aesthetic quality (Wu et al., 2023), the three VEFX editing scores (Gao et al., 2026) (rendering quality, instruction following, edit exclusivity), VBench background consistency (Huang et al., 2024), three control-alignment measures (DINOv3 feature displacement (Siméoni et al., 2025), UFM-based optical-flow agreement (Zhang et al., 2025), and ViPE-based camera-pose error (Huang et al., 2025)), and a realism score judged by Qwen3.8-27B-FP8 (Qwen Team, 2026) on a 0–100 scale.



Figure 5: **Qualitative comparison with baseline methods.** Three groups of two rows show game renders (top), geometry blockouts (middle), and stylized animations (bottom). "Rep." denotes our Editto Sim2Real and VACE 2.2 reproductions, fine-tuned on the same pseudo-paired videos as C2R# (ours).

Results. C2R# leads DOVER technical and aesthetic quality and all three VEFX dimensions, ranking second in background consistency and VLM-judged realism. Relative to the model from the original C2R paper, the C2R# model improves nine of the ten reported metrics, including spatial correspondence, local-flow agreement, and camera-pose error. The model from the original C2R paper retains the higher realism mean (75.50 versus 74.17). The C2R# model exceeds every external baseline on this realism score, with Editto Global the closest at 66.77.

The results also show a quality–control tradeoff. The adapted Editto model achieves the lowest DINO displacement, while default VACE 2.2 leads local-flow agreement and camera-pose accuracy. Thus, the quality gains of the C2R# model do not imply the strongest preservation under every control metric. Additional plausible detail and secondary motion may contribute to source–output disagreement.

Appendix B.8 additionally compares first-attempt results with the commercial model Seedance 2.5 (ByteDance Seed, 2026) using the same control videos and text prompts.

Transfer of pseudo-paired training data. Adapting Editto Sim2Real with our generated pairs improves all ten reported aggregate metrics, including realism from 52.62 to 58.52. The effect on VACE 2.2 is mixed: adaptation improves DINO spatial correspondence and VEFX edit exclusivity, while reducing the remaining reported scores. These results support the utility of pseudo-paired

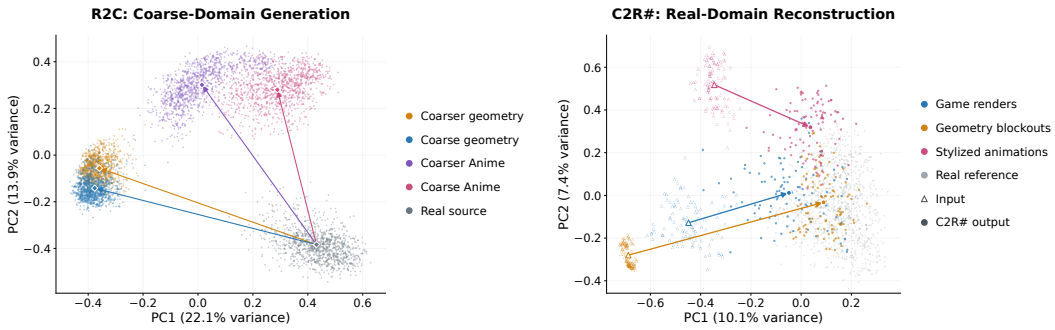

Figure 6: **Domain transfer in CSD-L style-feature space. Left:** DINO-R2C maps 1,000 real videos to four distinct pseudo-coarse style clusters. **Right:** C2R# maps actual coarse videos toward the real-reference distribution. Points denote videos; arrows connect input/output centroids. PCA is fitted per panel (Appendix B.4).

supervision beyond the C2R# model, while showing that its benefit depends on the recipient model and adaptation configuration.

Domain transfer. Figure 6 visualizes both models in the CSD-L style-feature space (Somepalli et al., 2024): the DINO-R2C model moves real videos into four distinct coarse style clusters, and the C2R# model moves each evaluation set of actual coarse videos toward the distribution of real reference videos.

Reference-image conditioning. A separately trained extension uses a reference image to guide appearance while the coarse video controls layout and motion (details and examples in Appendix B.7).

5 Limitations and Discussion

Our method relies on pseudo-coarse videos generated by DINO-R2C model to approximate the coarse inputs encountered at inference. Performance may therefore degrade when the gap between pseudo-coarse and actual coarse videos becomes large. Evaluation on actual coarse videos is also challenging because paired realistic ground truth is unavailable, requiring reference-free metrics and human judgment.

Our current implementation is constrained by the quality of the non-commercial backbone model and training data available. The frozen video prior also introduces a trade-off between realism and exact control alignment. When the input contains abrupt pose or motion changes, C2R# model may adjust the motion pace or synthesize intermediate states rather than following the discontinuity exactly. This can produce a more natural and temporally coherent video, but with reduced alignment to the input control (Figure 7). Future work includes closing the gap between pseudo-coarse and actual coarse features, strengthening reference-image control, and broadening evaluation on actual coarse videos.

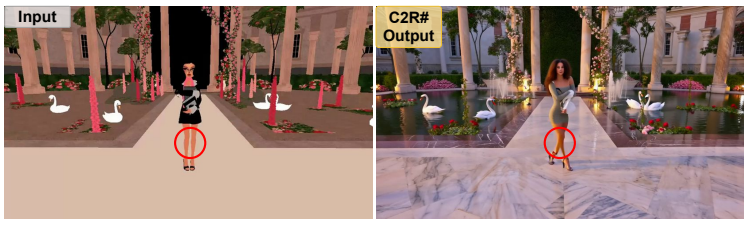

Figure 7: **Limitations under abrupt control changes.** Input (left) and C2R# output (right): abrupt control changes can lead to unwanted content and reduced alignment.

AI USE STATEMENT

We used generative AI models, including GPT-Image 2 and Wan 2.2, to create synthetic data as described in the data-generation and evaluation sections. AI agents assisted with constructing evaluation scenes in Roblox Studio. We also used AI assistants for manuscript editing, code assistance, and figure preparation. The authors reviewed and edited the AI-assisted text and figures and checked the AI-assisted code and generated data used in the experiments. The authors take responsibility for the final text, experimental results, and associated artifacts.

REFERENCES

Hassan Abu Alhaija, Jose Alvarez, Maciej Bala, Tiffany Cai, Tianshi Cao, et al. Cosmos-Transfer1: Conditional world generation with adaptive multimodal control, 2025.

Qingyan Bai, Qiuyu Wang, Hao Ouyang, Yue Yu, Hanlin Wang, Wen Wang, Ka Leong Cheng, Shuailei Ma, Yanhong Zeng, Zichen Liu, Yinghao Xu, Yujun Shen, and Qifeng Chen. Scaling instruction-based video editing with a high-quality synthetic dataset, 2025. URL https://arxiv.org/abs/2510.15742.

Ole Beisswenger, Jan-Niklas Dihlmann, and Hendrik P. A. Lensch. FrameDiffuser: G-buffer-conditioned diffusion for neural forward frame rendering, 2025.

Shariq Farooq Bhat, Niloy J. Mitra, and Peter Wonka. LooseControl: Lifting ControlNet for generalized depth conditioning. In *ACM SIGGRAPH 2024 Conference Papers*, 2024. doi: 10.1145/3641519.3657525. URL https://shariqfarooq123.github.io/loose-control/.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to follow image editing instructions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Ryan Burgert, Yuancheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, et al. Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

ByteDance Seed. Seedance 2.5. https://seed.bytedance.com/en/seedance2_5, 2026.

Shengqu Cai, Duygu Ceylan, Matheus Gadelha, Chun-Hao Paul Huang, Tuanfeng Yang Wang, and Gordon Wetzstein. Generative rendering: Controllable 4D-guided video generation with 2D diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Dana Cohen-Bar, Ido Sobol, Raphael Bensadoun, Shelly Sheynin, Oran Gafni, Or Patashnik, Daniel Cohen-Or, and Amit Zohar. RealMaster: Lifting rendered scenes into photorealistic video, 2026.

Edoardo A. Dominici, Thomas Deixelberger, Konstantinos Vardis, and Markus Steinberger. ControlDINO: Feature space conditioning for controllable image-to-video diffusion, 2026.

Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *International Conference on Computer Vision*, 2023.

Xiangbo Gao, Sicong Jiang, Bangya Liu, Xinghao Chen, Minglai Yang, Siyuan Yang, Mingyang Wu, Jiongze Yu, Qi Zheng, Haozhi Wang, Jiayi Zhang, Jie Yang, Zihan Wang, Qing Yin, and Zhengzhong Tu. VEFX-Bench: A holistic benchmark for generic video editing and visual effects, 2026. URL https://arxiv.org/abs/2604.16272.

Gonzalo Gomez-Nogales, Yicong Hong, Chongjian Ge, Peiye Zhuang, Marc Comino-Trinidad, Dan Casas, and Yi Zhou. Coarse-to-real: Generative rendering for populated dynamic scenes, 2026. URL https://arxiv.org/abs/2601.22301.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Maneesh Agrawala, Dahua Lin, and Bo Dai. SparseCtrl: Adding sparse controls to text-to-video diffusion models. In *European Conference on Computer Vision*, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixé, and Sanja Fidler. ViPE: Video pose engine for 3D geometric perception. Technical report, NVIDIA, 2025. URL https://research.nvidia.com/labs/toronto-ai/vipe/.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024. URL https://arxiv.org/abs/2311.17982.

Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One transformer to rule universal image segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://arxiv.org/abs/2211.06220.

Zhen Jiang, Zhen Han, Chengzhi Mao, Jing Zhang, Yingwei Pan, and Ziwei Liu. Vace: All-in-one video creation and editing. In *International Conference on Computer Vision*, 2025.

Dong-Yang Li, Wang Zhao, Yuxin Chen, Wenbo Hu, Meng-Hao Guo, Fang-Lue Zhang, Ying Shan, and Shi-Min Hu. Pixal3D: Pixel-aligned 3d generation from images, 2026a. URL https://arxiv.org/abs/2605.10922.

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

Guixu Lin, Zheng-Hui Huang, Siqi Yang, Ming-Hsuan Yang, Kaipeng Zhang, and Zhixiang Wang. Generative world renderer at the speed of play, 2026.

Han Lin, Jaemin Cho, Abhay Zala, and Mohit Bansal. Ctrl-Adapter: An efficient and versatile framework for adapting diverse controls to any diffusion model. In *International Conference on Learning Representations*, 2025.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22160–22169. IEEE, 2024.

Jiaming Liu, Felix Petersen, Yunhe Gao, Yabin Zhang, Hyojin Kim, Akshay S. Chaudhari, Yu Sun, Stefano Ermon, and Sergios Gatidis. Unpaired image-to-image translation via a self-supervised semantic bridge, 2026.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.03003.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Meta. SAM 3.1: Release notes. https://github.com/facebookresearch/sam3/blob/main/RELEASE_SAM3p1.md, 2026.

NVIDIA. DLSS 5: Generative neural rendering. https://research.nvidia.com/labs/adlr/DLSS5/, 2026.

OpenAI. GPT-Image-2. https://developers.openai.com/api/docs/models/gpt-image-2, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Qwen Team. Qwen3.8-27B-FP8: Model card. https://huggingface.co/Qwen/Qwen3.8-27B-FP8, 2026.

Stephan R. Richter, Hassan Abu AlHaija, and Vladlen Koltun. Enhancing photorealism enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Isaac Robinson, Peter Robicheaux, Matvei Popov, Deva Ramanan, and Neehar Peri. RF-DETR: Real-time detection transformer. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2511.09554.

Seyedmorteza Sadat, Otmar Hilliges, and Romann M. Weber. Eliminating oversaturation and artifacts of high guidance scales in diffusion models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.02416.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, et al. Dinov3, 2025.

Gowthami Somepalli, Anubhav Gupta, Kamal Gupta, Shramay Palta, Micah Goldblum, Jonas Geiping, Abhinav Shrivastava, and Tom Goldstein. Measuring style similarity in diffusion models. In *European Conference on Computer Vision*, 2024. URL https://arxiv.org/abs/2404.01292.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, 2020. URL https://arxiv.org/abs/2003.12039.

Team Wan. Wan: Open and advanced large-scale video generative models, 2025.

Kaihong Wang, Kumar Akash, and Teruhisa Misu. Learning temporally and semantically consistent unpaired video-to-video translation through pseudo-supervision from synthetic optical flow. In *AAAI Conference on Artificial Intelligence*, 2022.

Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. MoGe: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5261–5271, 2025. URL https://arxiv.org/abs/2410.19115.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Advances in Neural Information Processing Systems*, 2018.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861. URL https://www.ece.uwaterloo.ca/~z70wang/publications/ssim.html.

Haoning Wu, Erli Zhang, Liang Liao, Chaofeng Chen, Jingwen Hou, Annan Wang, Wenxiu Sun, Qiong Yan, and Weisi Lin. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In *IEEE/CVF International Conference on Computer Vision*, 2023. URL https://github.com/VQAssessment/DOVER.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything V2. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.09414.

Xitong Yang, Devansh Kukreja, Don Pinkus, Anushka Sagar, Taosha Fan, Jinhyung Park, Soyong Shin, Jinkun Cao, Jiawei Liu, Nicolas Ugrinovic, Matt Feiszli, Jitendra Malik, Piotr Dollár, and Kris Kitani. SAM 3D Body: Robust full-body human mesh recovery. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2602.15989.

Zheng Zeng, Marco Salvi, Lifan Wu, Jan Novák, Daqi Lin, Saeed Hadadan, Yichen Sheng, Robert Pottorff, Shiqiu Liu, Ravi Ramamoorthi, Ling-Qi Yan, and Miloš Hašan. RGBX-Next: Towards realistic generative rendering from G-buffers, 2026.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *International Conference on Computer Vision*, 2023.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Yuchen Zhang, Nikhil Keetha, Chenwei Lyu, Bhuvan Jhamb, Yutian Chen, Yuheng Qiu, Jay Karhade, Shreyas Jha, Yaoyu Hu, Deva Ramanan, Sebastian Scherer, and Wenshan Wang. UFM: A simple path towards unified dense correspondence with flow. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2506.09278.

Jixin Zhao, Zhouxia Wang, Peiqing Yang, and Shangchen Zhou. Precise object and effect removal with adaptive target-aware attention. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2505.22636.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, volume 2026, pp. 35791–35820, 2026.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision*, 2017.

A Implementation Details

A.1 Image-Pair Corpus

We use 14,000 aligned (coarse, realistic) image pairs for each of four modalities: coarser geometry, coarse geometry, coarser anime, and coarse anime. All realistic source images are generated with GPT-Image 2 (OpenAI, 2026). Both source images and coarse targets are generated at 1280×720 (720p). Larger per-modality training sets provided only marginal improvements, so we use the same 14,000-pair budget for all four modalities.

Coarse geometry, coarser anime, and coarse anime. We use GPT-Image 2 in image-editing mode to create these three types of coarse target from the realistic source images. Modality-specific instructions request the desired geometric simplification or anime stylization while preserving the source subjects, scene layout, poses, and camera viewpoint.

Coarser geometry. We reconstruct and simplify a 3D scene from each generated realistic source image, processing people and background separately. For images containing people, RF-DETR 2XL (Robinson et al., 2026) detects them, SAM 3.1 (Meta, 2026) extracts their masks, and SAM 3D Body (Yang et al., 2026) estimates their 3D bodies. ObjectClear (Zhao et al., 2026) removes the people and inpaints the background. OneFormer (Jain et al., 2023) separates ground, sky, and objects, with SAM 3.1 providing object-instance masks. We reconstruct objects with Pixal3D (Li et al., 2026a), use the masked MoGe (Wang et al., 2025) point cloud to determine their scene depth, and triangulate its ground points to recover the terrain, excluding sky regions. We simplify object meshes into primitives or convex parts and retarget the reconstructed human poses to blocky Roblox R15 characters using a deterministic MHR-to-Roblox mapping. Finally, we compose the simplified background and characters and render from the source camera, preserving scene layout, perspective, and occlusion to obtain the paired coarser-geometry target.

A.2 DINO-R2C Architecture and Training

The frozen encoder E is DINOv3-L/16, using affine-normalized final-layer patch tokens (layer index 23). At 1280×720, it produces a $45 \times 80 \times 1024$ feature grid. The DINO-R2C model has one trainable DINOv3-L backbone shared across all four modalities and four modality-specific residual token-MLP heads, each with hidden width 2048 and a zero-initialized output projection.

We first fine-tune the RAE decoder D (Zheng et al., 2026) for reconstruction at 1280×720 (720p), then freeze it during DINO-R2C training and pseudo-paired video synthesis. It maps the predicted features back to coarse RGB pixels. We optimize the DINO-R2C model with AdamW (Loshchilov & Hutter, 2019) using learning rate 3×10^{-5}, weight decay 0.05, and gradient clipping at 1.0. The shared four-modality model is trained for 20,000 iterations. Our selected model uses LPIPS and $2\times$ camera augmentation; Section 4.2 evaluates the effect of LPIPS and of the augmentation strength.

A.3 Camera Augmentation

We apply the same aspect-corrected camera homography to each realistic source image and its paired coarse target, preserving their alignment. The $1\times$ recipe uses retained crop area in $[0.75, 1.00]$, translation bounded by ± 0.06 in normalized image coordinates, rotation by $\pm 3.55°$, horizontal and vertical shear by $\pm 2.29°$, and perspective coefficients by ± 0.041. The $2\times$ recipe doubles these perturbation bounds and doubles the maximum removable crop area, giving a retained-area range of $[0.50, 1.00]$. The $0.5\times$ variant halves the perturbation bounds and retains crop area in $[0.875, 1.00]$. All variants use bilinear resampling without horizontal flips, photometric augmentation, or padding. The multiplier controls augmentation strength, not the number of training examples.

A.4 Pseudo-Paired Video Synthesis

Our training corpus contains 389,854 real videos drawn exclusively from DL3DV (Ling et al., 2024), `SEKAI_HQ`, and `SEKAI_DRONE` (Li et al., 2026b). We process clips at 1280×720, 16 fps, using 81 frames per clip. During training, the shared DINO-R2C model processes each frame independently

using the selected modality head, and the frozen RAE decoder produces the corresponding pseudo-coarse RGB video. The source real video provides its paired training target. These pseudo-paired videos are synthesized on the fly, with equal sampling budgets across the four coarse modalities. We apply the same resize-to-cover and center-crop transform to the pseudo-coarse control and real target to obtain 832×480 inputs.

A.5 C2R# Architecture and Training

The video backbone is Wan2.2 T2V-A14B, consisting of a 16-channel Wan2.1 VAE, a umT5-xxl text encoder, and two 40-layer, 5120-wide MoE DiTs with expert boundary $\sigma = 0.875$. We train two control adapters, one for each expert. Each adapter contains an input projection $1024 \to 512$, four residual 3D convolution blocks with dilations 1 and 2 and FiLM timestep conditioning, causal temporal compression from 81 input frames to 21 latent frames, and a zero-initialized output projection $512 \to 5120$. This is followed by a zero-initialized fusion adapter consisting of AdaLN and a depthwise convolution.

The C2R# model operates at 832×480. At this resolution, DINOv3-L/16 produces a 30×52 spatial feature grid. The VAE latent has shape $(16, 21, 60, 104)$ and, after 2×2 spatial patchification, the DiT token grid is $21 \times 30 \times 52$, so each DINO spatial cell corresponds directly to one DiT spatial token.

We train only the control and fusion adapters using the rectified-flow velocity loss of Section 3.4 and AdamW with learning rate 10^{-4}. The selected run optimizes 180,540,416 trainable adapter parameters for one epoch over 389,854 eligible records, corresponding to 12,183 optimization steps with global batch size 32.

A.6 Inference Settings

At inference, the C2R# model generates 81 frames at 16 fps and 832×480 resolution, using random seed 4321 for every video. We resize each input control video while preserving its aspect ratio to cover 832×480, then center-crop to this resolution, applying the same spatial transform to every frame.

We use a shifted-σ schedule with shift 12 and 50 denoising steps, with adaptive projected guidance (APG) (Sadat et al., 2025) at scale 10, $\eta = 0$, and momentum -0.5, adjusting the guidance radius separately for each evaluation set following that work. Each step uses two forward passes with the same coarse control: one with the scene prompt and one with the Wan2.2 default negative prompt. Guidance is applied to the denoised predictions, then converted back to velocity for the Euler update.

B Evaluation Protocol

B.1 Evaluation Sets

We evaluate on three sets of 100 coarse videos each, totaling 300 control videos. These sets cover game renders, geometry blockouts, and stylized animations and are used for all compared methods.

Sources of game renders. The 100 videos are recordings of renderings from Roblox games.

Construction of geometry blockouts. We use Roblox Studio to prototype scenes with an agent. This workflow allowed us to create all 100 evaluation scenes in under two hours in total and render them as coarse control videos.

Generation of stylized animations. We generate all 100 videos with Wan 2.2 T2V (Wan, 2025), using text prompts to produce anime-style animations.

Table 2 reports aggregate metrics across all 300 videos, with equal weight per video.

B.2 Compared Methods

The adapted VACE 2.2 and Editto Sim2Real baselines are fine-tuned on exactly the same pseudo-paired training data as the C2R# model: the same 389,854 real videos from DL3DV, `SEKAI_HQ`, and `SEKAI_DRONE`, paired with the corresponding pseudo-coarse controls synthesized by our shared DINO-R2C model. All three methods use the same four coarse modalities with equal sampling budgets and the same 832×480, 81-frame, 16-fps media format.

B.3 Metric Definitions

DINO-R2C image metrics (Table 3). Each realistic input image has a paired coarse target. All eight metrics compare the prediction with this target and measure how faithfully the model reproduces the intended coarse image. The model operates at 1280×720. Latent MSE uses the native $45 \times 80 \times 1024$ feature grid before decoding; the other metrics evaluate decoded images clipped to $[0, 1]$ and resized to 256×256 with bicubic interpolation. Table entries are means over the 1,000 held-out image pairs.

Latent MSE (↓). Mean squared error measures the average squared difference between the predicted feature grid and the target grid produced by the frozen encoder. It tests whether the model predicts the intended coarse representation before the RAE decoder turns it into pixels. Zero means identical feature grids; lower values indicate more accurate feature reconstruction. Its scale differs from pixel-space MSE, so the two columns should not be compared numerically.

RGB MSE (↓). We average the squared differences between corresponding output and target pixel values across all three color channels. This measures pixel-level color and appearance error, giving larger mistakes more weight because differences are squared. Zero means an exact pixel match. Small shifts in otherwise similar shapes can still produce substantial pixel error.

RGB L1 (↓). We average absolute, rather than squared, differences between corresponding RGB values. This also measures pixel-level reconstruction error, but gives less emphasis to a few large mistakes than MSE. With RGB values in $[0, 1]$, an L1 value of 0.1 means an average absolute channel-value difference of 0.1; it is not a percentage of incorrectly reconstructed pixels.

LPIPS (↓). Learned Perceptual Image Patch Similarity (Zhang et al., 2018) compares output and target using features from a frozen perceptual network. It measures differences in visual appearance, including texture and structure, that exact pixel comparisons may describe poorly. Despite its name, the reported quantity is a distance: lower values indicate greater perceptual similarity. It measures resemblance to the coarse target, not photorealism.

DINO patch cosine (↑). We extract DINOv3 patch features (Siméoni et al., 2025) from the decoded output and target, compute cosine similarity at corresponding spatial locations, and average across patches. This measures local agreement in scene content and structure. A value closer to one indicates more similar feature directions, independently of their magnitudes. It can remain high despite some color or texture differences and does not replace pixel-level fidelity measures.

Depth RMSE (↓). We estimate relative-depth maps for the output and target using Depth Anything V2 (Yang et al., 2024), align their scale and offset for each image pair, and compute the root mean squared difference. This measures agreement in the inferred scene geometry after removing arbitrary depth scale and offset. Lower values indicate closer geometric agreement. The score uses estimated relative depth, not measured distances, and is not an error in meters.

SSIM (↑). Structural Similarity (Wang et al., 2004) compares local brightness, contrast, and spatial structure between output and target. Our implementation uses luminance (BT.601 luma) and an 11×11 Gaussian window. Higher values, approaching one, indicate more similar local image structure. Because it operates on luminance, it does not directly measure color agreement.

Boundary F1 (↑). We extract Sobel edges from output and target, retain responses above 0.08, and match edges within a two-pixel tolerance. The F1 score combines precision (how many predicted

edges match target edges) and recall (how many target edges are recovered). Extra or missing outlines reduce the score; higher values indicate better boundary reconstruction. This compares outlines with the paired target, whereas Table 1's boundary stability measures their consistency between neighboring video frames.

DINO-R2C video metrics (Table 1). There are no ground-truth coarse target videos. We compare the temporal behavior of the pseudo-coarse output with the original real video, using 81 frames per clip resized to 256×256. RAFT-Small (Teed & Deng, 2020) estimates motion between neighboring source frames in both temporal directions. We use these correspondences to align neighboring source and output frames before comparing their appearance, compensating for motion present in the source. We retain only in-bounds correspondences for which following the forward flow and then the backward flow returns close to the starting pixel. This consistency mask reduces the influence of occlusions and unreliable motion estimates. Table entries are means of the corresponding per-video scores.

Perceptual flicker (relative, ↓). We measure LPIPS between neighboring output frames after aligning them using source motion. For each adjacent-frame comparison, this perceptual error is divided by the corresponding motion-aligned source-video error. We take the median of these ratios within each video to summarize typical temporal behavior, then average across videos. A value of one means source-level error; values below or above one indicate less or more error under this normalization. This is a temporal error ratio, not a reconstruction-accuracy percentage.

Smoothed-frame flicker (relative, ↓). We spatially blur frames and measure their absolute RGB difference (L1) after motion alignment. Blurring reduces sensitivity to fine texture and emphasizes changes in broad colors and shapes. As for perceptual flicker, each output error is divided by its corresponding source error, and per-video medians are averaged across videos. Both flicker columns therefore summarize typical behavior using the same aggregation rule. Lower values indicate less motion-aligned appearance instability; smoothing can itself reduce these errors, so they should be interpreted alongside the motion and content metrics.

Motion error (EPE, ↓). Endpoint error measures the mean distance between optical-flow vectors estimated independently from the output and source videos. Differences in either direction or magnitude increase the error. Lower values indicate closer agreement with source motion. Both motion fields are estimated from images; the source flow is a reference, not ground-truth motion annotation.

Motion magnitude ratio (≈ 1). For each video, we divide mean output flow magnitude by mean source flow magnitude. A ratio below one indicates attenuated motion, while a ratio above one indicates amplified motion; one is preferred. For example, a value of 1.0797 indicates approximately 8% greater estimated motion magnitude. This metric measures the amount of motion, not its direction, and complements motion error.

Boundary stability (Dice, ↑). We extract edge maps from neighboring output frames, align them using source motion, and measure their overlap with a soft Dice score. Higher overlap indicates that outlines persist more consistently across frames. This compares the output with its temporal neighbors, not with ground-truth coarse boundaries.

Content consistency (DINO, ↑). We spatially average DINOv3 features within each output frame and compute mean cosine similarity between consecutive frame descriptors. Higher similarity indicates more stable overall scene and subject content. Spatial averaging can hide local flicker, and an overly static output can also score highly, so this metric should be read together with the motion-preservation measures.

C2R# metrics (Table 2).

Perceptual quality and editing assessment. Table 2 separates output quality, editing success, control alignment, and realism. DOVER (Wu et al., 2023) technical quality (Tech.) measures technical visual quality, while its aesthetic branch (Aesth.) assesses aesthetic appeal. VEFX-Reward-32B (Gao et al., 2026) assesses the control video, editing instruction, and generated result on three

1–4 scales: rendering quality (RQ), including visual clarity, temporal stability, and artifacts; instruction following (IF), measuring execution of the requested edit; and edit exclusivity (EE), assessing unintended changes outside the requested transformation. Because coarse-to-real generation intentionally changes appearance throughout the scene, EE is interpreted relative to the editing request, rather than as a direct measure of spatial or motion alignment.

Temporal consistency and control alignment. VBench (Huang et al., 2024) background consistency (BG) measures the temporal consistency of generated background appearance. It does not directly measure agreement with source motion. The DINO metric measures normalized spatial displacement of matched DINOv3 features (Siméoni et al., 2025) between synchronized control and output frames; lower values indicate closer spatial correspondence among successfully matched features. Flow denotes local optical-flow agreement between the control and generated videos, estimated with UFM (Zhang et al., 2025) (UFM-Base-DINOv2G-init), with covisibility handling and scale normalization. Cam. denotes relative camera-pose error computed from ViPE-estimated trajectories (Huang et al., 2025), measuring camera preservation separately from local object motion. It is an error under the evaluation protocol, not an absolute distance in meters or an angle in degrees.

VLM-judged realism. We use Qwen3.8-27B-FP8 (Qwen Team, 2026) to assess generated videos sampled at 8 frames per second. The judge assigns four scores on a 1–5 scale: appearance, structural plausibility, temporal stability, and motion and physics. For video i, we compute

$$R_i = 25 \left(\frac{s_{i,\text{app}} + s_{i,\text{str}} + s_{i,\text{temp}} + s_{i,\text{motion}}}{4} - 1 \right). \tag{5}$$

This maps the mean rating to a 0–100 scale. All evaluation sets use photorealistic live-action footage as the target domain. The scene prompt provides context, but this assessment scores visible realism rather than prompt adherence or source-control similarity. In particular, plausible output motion does not necessarily imply preservation of the input trajectory.

B.4 Domain-Transfer Visualization

For Figure 6, each video is represented by averaging unit-normalized CSD-L style descriptors (Somepalli et al., 2024) from nine frames at 480p and normalizing the resulting vector. The left panel contains the same 1,000 real videos and their four pseudo-coarse versions; the right panel contains the 300 actual coarse control videos, their C2R# outputs, and 1,000 unmatched real reference videos. PCA is fitted independently for each panel with equal weight on every video; axes show the first two principal-component scores, and percentages give the variance explained by each component within that panel.

B.5 Additional DINO-R2C Results

Image ablations. Table 3 compares the six training recipes on 1,000 held-out Coarse geometry image pairs. The $2\times$ augmentation recipe with LPIPS achieves the best latent MSE, RGB L1, LPIPS, DINO patch cosine similarity, and depth RMSE. It does not optimize every reconstruction criterion: $1\times$ augmentation without LPIPS achieves the best RGB MSE and SSIM, while training without augmentation with LPIPS achieves the best boundary F1. Metric definitions are given in Appendix B.3.

Table 3: **DINO-R2C design ablations, image quality.** Evaluated on 1,000 held-out images in the Coarse geometry modality, comparing predictions with paired coarse targets. Latent MSE measures feature reconstruction; RGB MSE and L1 measure pixel errors; LPIPS and DINO patch cosine measure perceptual and local content agreement; depth RMSE, SSIM, and boundary F1 assess geometry, local image structure, and outlines. Latent MSE uses the native feature grid; other metrics use 256×256 images. Values are means across images. Bold and underlining mark the best and second-best values. Definitions are given in Appendix B.3.

DINO-R2C Recipe	Latent MSE ↓	RGB MSE ↓	RGB L1 ↓	LPIPS ↓	DINO patch cosine ↑	Depth RMSE ↓	SSIM ↑	Boundary F1 ↑
No Aug, no LPIPS	0.4443	0.0217	0.1138	0.3980	0.9055	0.4537	0.6225	0.3995
No Aug, w/ LPIPS	0.4406	0.0218	0.1137	0.3707	0.9121	0.4447	0.6122	**0.5150**
$0.5\times$ Aug, w/ LPIPS	0.4296	0.0214	0.1129	0.3700	<u>0.9128</u>	0.4380	0.6236	<u>0.4654</u>
$1\times$ Aug, no LPIPS	<u>0.4276</u>	**0.0211**	<u>0.1123</u>	0.4127	0.9020	0.4483	**0.6359**	0.2889
$1\times$ Aug, w/ LPIPS	0.4280	0.0212	0.1126	<u>0.3682</u>	0.9127	<u>0.4357</u>	0.6247	0.4644
$2\times$ **Aug, w/ LPIPS (ours)**	**0.4237**	<u>0.0212</u>	**0.1122**	**0.3659**	**0.9136**	**0.4352**	<u>0.6282</u>	0.4573

Qualitative results. Figure 8 shows realistic source images and decoded coarse predictions across the four modalities. Pseudo-Coarse videos, obtained by applying the DINO-R2C model to each frame of real clips, are provided in the supplemental HTML page, together with additional image-pair results.



Figure 8: Qualitative DINO-R2C results. Each example shows a realistic image and decoded coarse predictions for Coarser geometry, Coarse geometry, Coarser Anime, and Coarse Anime. The predictions illustrate modality-dependent appearance transfer while retaining scene structure.

B.6 Efficiency and Computational Cost

Training parameters. Because the Wan2.2 backbone stays frozen, the C2R# model trains only its control and fusion adapters. Table 4 distinguishes total model capacity from the parameters optimized during adaptation. The training procedure in the original C2R paper fine-tunes the diffusion backbone in Stage I, then freezes it and trains the added control and fusion modules in Stage II; totals reflect the modules present in each stage. Editto fine-tunes LoRA weights on an existing frozen VACE control branch, which remains included in its total parameter count. The VACE 2.2 trainable count follows its documented control-module training configuration. The C2R# model optimizes 180.54M adapter parameters, or 0.63% of the counted model parameters. For VACE 2.2 and the C2R#, counts sum both denoising experts, although only one expert is active at each denoising step.

Table 4: Training parameters. Totals include the diffusion backbone and control/fusion modules, including unmerged LoRA where applicable, but exclude frozen external text and feature encoders, the VAE, and separately trained DINO-R2C models. Counts include both denoising experts for VACE 2.2 and the C2R#. B and M denote billions and millions of parameters.

Method / training stage	Total params	Trainable params	Trainable (%)
C2R orig.: Stage I	14.288B	14.288B	100.00
C2R orig.: Stage II	14.333B	44.33M	0.31
Editto Global	17.460B	122.68M	0.70
Editto Sim2Real	17.460B	122.68M	0.70
VACE 2.2	34.675B	6.098B	17.59
C2R# (ours)	28.758B	180.54M	0.63

DINO-R2C efficiency. Table 5 reports the shared four-head model's parameter count, training hardware and cost, and inference performance at native 720p. The inference timings include the projector and frozen RAE decoder and exclude model loading, transfers, video codecs, and storage I/O.

Table 5: Model size and efficiency for a shared-backbone, four-head DINO-R2C architecture. Inference uses BF16 at 1280×720 and includes the projector and frozen RAE decoder, excluding model loading, transfers, video codecs, and storage I/O.

Quantity	Value
Model params	
Shared backbone + four heads	319.92M trainable
Training efficiency	
Hardware	3×8 B200 GPUs
Training iterations	20,000
Global batch size	192
Recorded training time	4h 02m 58s
Derived training cost	97.19 GPU-hours
Median training step time	0.703 s
Inference efficiency	
Hardware	1 B200 GPU
Image latency, batch 1	34.779 ms
Image latency, batch 8	199.303 ms/batch
Video length / frame chunk size	81 / 8
Median video latency	2.029 s
Video throughput	39.92 frames/s
Peak allocated GPU memory, video	4.820 GiB

B.7 Reference-Conditioned Generation

The C2R# model can additionally accept a reference image alongside the control video and text prompt, using separately trained reference-conditioned weights. The extension retains the control and fusion adapters for the control video and the original text conditioning. We encode the reference image with the frozen Wan VAE and append its tokens after the generation tokens. A temporal rotary-position offset of 100 separates the reference from the generated sequence, with reference-specific adaptive layer normalization for each expert. Rank-16 LoRA updates (Hu et al., 2022) (scaling parameter 16) adapt the query, key, value, and output projections of self-attention in all 40 blocks of both Wan experts. The original backbone weights remain frozen; the reference extension trains the adapters and LoRA parameters. The coarse video still controls layout and motion, while the reference image specifies additional appearance cues. Because this use case receives an extra input, we present it separately from the standard control-video-and-text comparisons.

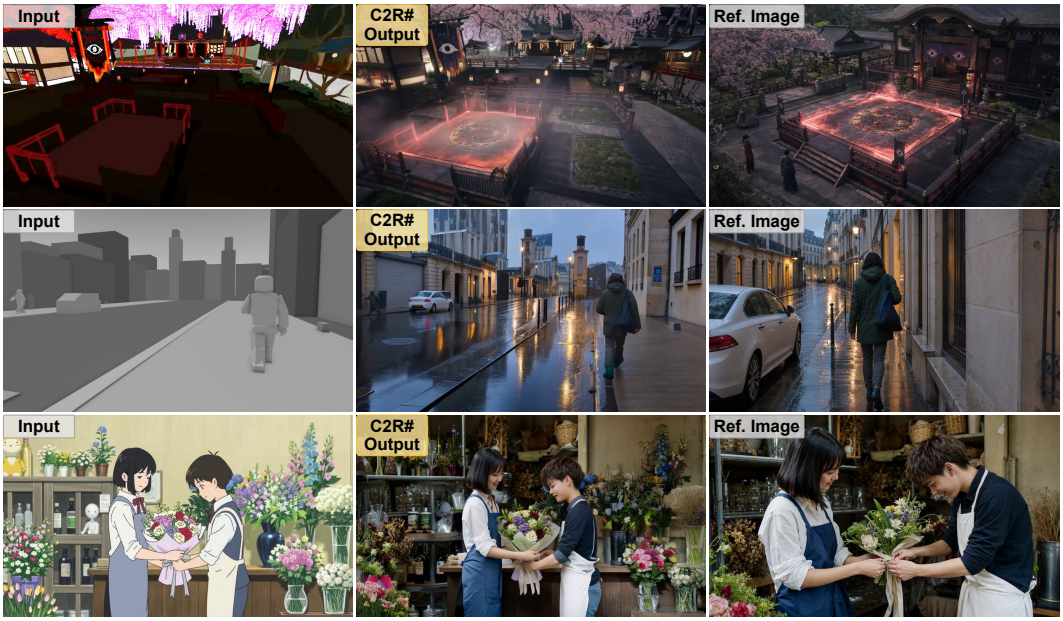

Figure 9: The C2R# model with a reference image. Each example includes a control video, the reference image used at inference, a text prompt, and the generated result. Examples cover game renders, geometry blockouts, and stylized animations. This extension uses separately trained weights and an additional image input, and is evaluated separately from the standard control-video-and-text setting.

B.8 Qualitative Comparison with Seedance 2.5

We compare the C2R# model with Seedance 2.5 (ByteDance Seed, 2026), a closed-source commercial video model, using the same control videos and text prompts on inputs drawn from our three evaluation sets. In our trials, when realistic domain transfer succeeds, Seedance 2.5 produces highly realistic results comparable in photorealism to the C2R# model. However, for this coarse-to-real task, we observed that it is prone to losing alignment with the input control video and can fail to produce the photorealistic appearance requested by the prompt. Figure 10 shows first-attempt examples of these failures: Seedance 2.5 changes the scene layout or retains visibly stylized or game-like elements, while the C2R# model more faithfully preserves the input structure and achieves the requested photorealistic appearance. We also found commercial editing expensive per video, making retries costly.

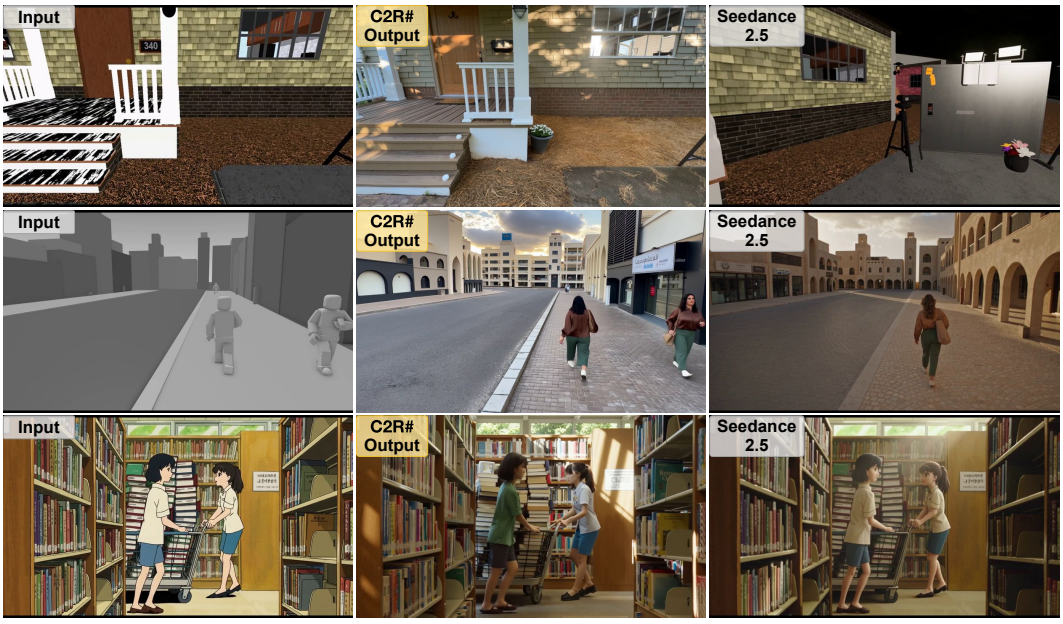

Figure 10: **First-attempt comparison with Seedance 2.5.** Columns show the input control, C2R# output, and Seedance 2.5 output. Rows show examples of game renders, geometry blockouts, and stylized animations, respectively. Both methods receive the same control video and text prompt, and the displayed results are from the first generation attempt. In these examples, the C2R# model produces a more photorealistic appearance, while Seedance 2.5 retains more of the coarse input style or changes the scene layout.