V-RGBX: Video Editing with Accurate Controls over Intrinsic Properties

Ye Fang[1,2,*], Tong Wu[3✉], Valentin Deschaintre[2], Duygu Ceylan[2], Iliyan Georgiev[2],
Chun-Hao Paul Huang[2], Yiwei Hu[2], Xuelin Chen[2], Tuanfeng Yang Wang[2✉]

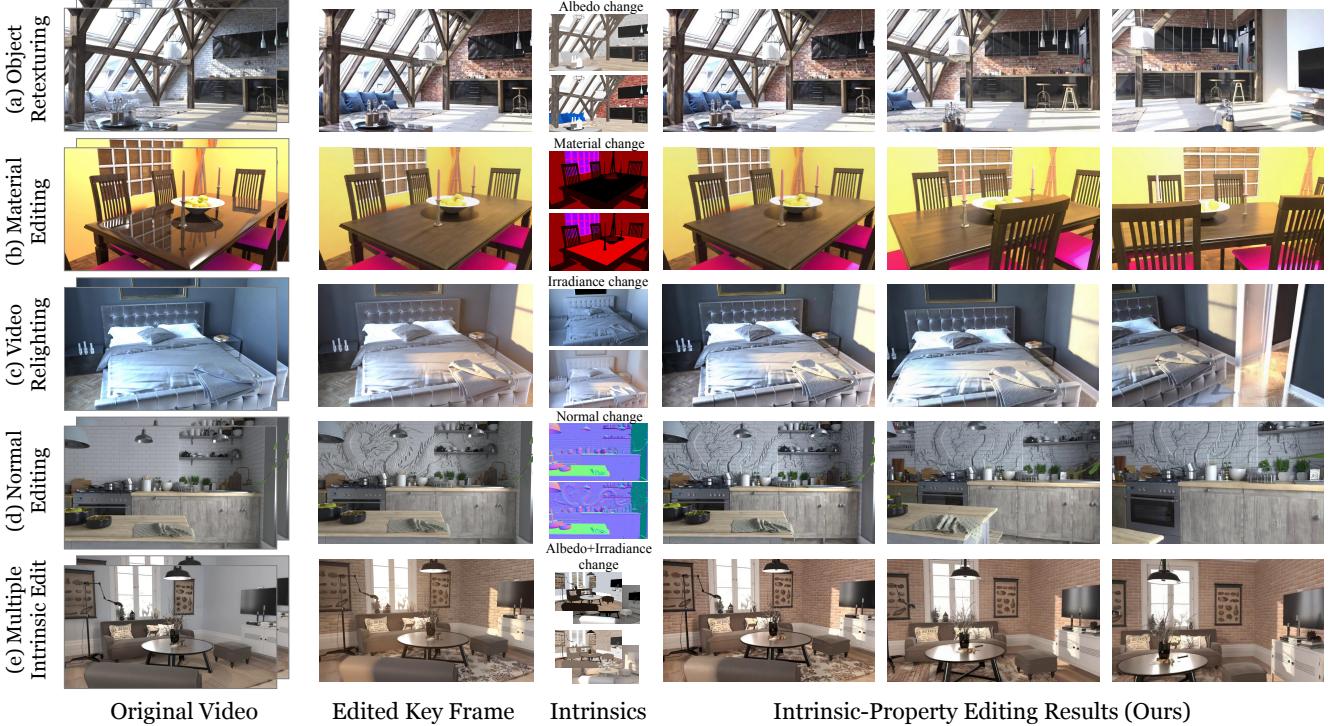[1]Fudan University   [2]Adobe Research   [3]Stanford University

Figure 1. **Overview.** Given a source input video and an edited keyframe obtained by manipulating various intrinsic properties, V-RGBX generates an edited video which propagates the edit in an intrinsic aware manner. V-RGBX is an end-to-end framework that understands intrinsic scene properties and uses them for generation to support tasks such as object retexturing, relighting, or material editing, etc.

Abstract

Large-scale video generation models have shown remarkable potential in modeling photorealistic appearance and lighting interactions in real-world scenes. However, a closed-loop framework that jointly understands intrinsic scene properties (e.g., albedo, normal, material, and irradiance), leverages them for video synthesis, and supports editable intrinsic representations remains unexplored. We present V-RGBX, the first end-to-end framework for intrinsic-aware video editing. V-RGBX unifies three key capabilities: (1) video inverse rendering into intrinsic channels, (2) photorealistic video synthesis from these intrinsic representations, and (3) keyframe-based video editing conditioned on intrinsic channels. At the core of V-RGBX is an interleaved conditioning mechanism that enables intuitive, physically grounded video editing through user-selected keyframes, supporting flexible manipulation of any intrinsic modality. Extensive qualitative and quantitative results show that V-RGBX produces temporally consistent, photorealistic videos while propagating keyframe edits across sequences in a physically plausible manner. We demonstrate its effectiveness in diverse applications, including object appearance editing and scene-level relighting, surpassing the performance of prior methods. Our project website is at: https://aleafy.github.io/vrgbx.

1

1. Introduction

Editing captured video sequences using large-scale generative models has witnessed remarkable progress in recent years. With the advent of powerful text-to-video and image-to-video diffusion models, users can now manipulate object appearance, scene layout, and motion dynamics through high-level language or visual instructions [8, 30, 47]. These advances have significantly expanded the scope of creative video synthesis and controllable generation.

However, direct control over intrinsic properties, such as albedo, irradiance, material, and shading, remains largely unexplored. These properties govern the physical realism and consistency of visual appearance, and fine-grained control over them is essential for many downstream applications, including relighting, material editing, and stylized video generation. For instance, one may wish to modify the shadeless texture of an object while preserving its lighting or change the environmental lighting without altering surface material. Achieving such effects requires explicit disentanglement of intrinsic factors from RGB observations and the ability to propagate intrinsic edits consistently across time — two capabilities absent in current video generation systems.

Despite the success of video diffusion models in controllable generation, existing approaches lack an explicit intrinsic representation space. Most prior works focus on appearance-level editing (e.g., texture or style transfer) or employ implicit latent conditioning that entangles illumination and material cues. Methods such as GenProp [30], VACE [23], DaS [15], or AnyV2V [26] extend diffusion models with auxiliary modality guidance (e.g., appearance, depth, flow, or semantics). However, these cues are injected directly into the pixel space, without disentanglement in the intrinsic domain. As a result, such approaches often fail to preserve key intrinsic properties consistently across frames after editing. Moreover, existing solutions typically apply conditioning in a global manner, based on a text prompt or a single reference frame, limiting their flexibility in practical video editing scenarios where multiple, localized edits may occur across different intrinsic modalities and time segments.

To address these limitations, we propose **V-RGBX**, an intrinsic-space video editing technique that enables keyframe-level control and propagation of intrinsic properties within a generative video model. Our framework begins with an intrinsic decomposition module (video RGB→X) that extracts physically interpretable modalities from input RGB frames, such as albedo, irradiance, normal, and depth. These modalities constitute a structured intrinsic editing space, where users can selectively modify any modality (e.g., changing albedo color or adjusting illumination) on sparse keyframes. At the core of our system lies a multi-modality conditioning diffusion transformer (DiT) that in-

tegrates intrinsic modalities within a unified generation process (video X→RGB). Rather than conditioning on pre-defined motion signals (e.g., DaS [15]), our model learns temporal dynamics from arbitrary modality inputs in an interleaving manner and propagates sparse keyframe edits throughout the sequence. Untouched modalities are randomly provided and preserved during generation, ensuring temporal and spatial consistency. Such a setup enables reliable multi-touch edits as user can provide multiple touches for different modalities over the sequence.

As shown in Fig. 1, this design enables a broad range of novel video editing capabilities with keyframe level control, including physically grounded relighting, albedo manipulation, and geometry-aware object insertion. Our approach achieves significant improvements in both fidelity and controllability over existing appearance-only baselines and establishes a foundation for physically consistent video generation beyond the RGB domain. Our main contributions are summarized as follows:

- We introduce V-RGBX, the first end-to-end intrinsic-aware video editing framework, which unifies video inverse rendering into intrinsic channels (video RGB→X), photorealistic video synthesis from intrinsic representations (video X→RGB), and keyframe-based video editing conditioned on intrinsic channels.

- We propose an interleaving conditioning mechanism for the DiT-based video model, enabling flexible intrinsic conditioning in both video X→RGB synthesis and keyframe-edit propagation.

- We present an intuitive video editing workflow that maintains temporal coherence while supporting keyframe-level, multi-modality control and propagation of intrinsic edits. We demonstrate its effectiveness through intrinsic-driven object appearance editing and scene-level relighting applications.

2. Related work

2.1. Video Diffusion Models

Diffusion models have become the leading paradigm for visual synthesis, surpassing GANs in fidelity and diversity by modeling complex data distributions via iterative denoising [12, 27, 33, 43, 45]. Building upon their success in image generation, recent works have naturally extended diffusion models to video generation [1, 2, 9, 16, 18, 20, 42, 46, 49]. Early video diffusion models [9, 10, 16, 18, 20, 42] adopt U-Net [40]-based architectures, incorporating temporal modules to capture frame-to-frame dependencies while maintaining spatial feature processing. More recent approaches [24, 46, 49] transition to diffusion transformers (DiT), leading to enhanced temporal coherence and overall visual quality. In this work, we build on the WAN

model [46], leveraging its strong prior and adapting them for intrinsic properties extraction and video editing.

2.2. Intrinsic-Aware Diffusion Models

There has been increasing focus on intrinsic image decomposition, composition, and editing. Existing methods [13, 41, 52] enables simple edits by modifying intrinsic components and recomposing them, while IntrinsicEdit [32] introduces a short residual optimization step that allows precise and controllable intrinsic attribute editing. However, these methods operate purely at the image level. When extended to videos, the introduction of the temporal dimension raises new challenges: edits applied to a single frame must be accurately propagated to subsequent frames while maintaining temporal consistency. DiffusionRenderer [29] enables video decomposition and recomposition but doesn't enable propagation of pixelwise editing. X2Video [22] is a video-level X→RGB method that mainly extends image-level models to the temporal domain, while it cannot directly propagate or edit intrinsic attributes frame by frame without a complete sequence of edited intrinsic properties.

To address this, our method builds upon a pretrained video diffusion framework [46] and proposes a single-frame interleaved conditioning strategy to fuse and propagate intrinsic controls across time. Moreover, a type-aware embedding is introduced to explicitly differentiate between various intrinsic types at each timestep, ensuring precise edits while achieving cross-frame consistency and stability in intrinsic-aware video synthesis.

2.3. Controlled Video Generation and Editing

Controlled video generation conditioned on various signals has recently attracted growing attention due to its wide range of applications. For example, a series of works have explored camera control [4–7, 17, 47], while other methods incorporate structural conditioning via point clouds, object tracking, or 3D-aware priors, which have proven effective in improving spatial consistency and trajectory alignment [14, 50, 51]. Beyond spatial control, several models have begun supporting action-based or scene-level conditioning [11, 35, 48], collectively advancing video generation toward a world simulator paradigm. Recent advances have also enabled general video editing [23, 30], supporting multiple conditioning modalities with different editing tasks in a unified framework. However, achieving more physically realistic rendering and editable generation remains challenging, as intrinsic-based tasks are still largely unexplored. In contrast, our framework can recover multiple intrinsic properties, including illumination, from video inputs and re-compose them through intrinsic-conditioned video generation. It thus provides a complete editing pipeline of RGB→X and X→RGB, allowing precise propagation and control of per-frame intrinsic edits across time.

3. Method

3.1. Overview

We propose **V-RGBX**, an intrinsic-aware video editing framework that enables precise and temporally consistent propagation of intrinsic edits. Unlike previous video diffusion models trained purely in the RGB domain, which often lead to entangled control over lighting, texture, and geometry, **V-RGBX** introduces an explicit *intrinsic-conditioned representation space*, ensuring that edits to one property do not unintentionally affect others.

Specifically, starting with an input video $V = \{v_1, v_2, \ldots, v_T\}$, the user selects a set of keyframes $\{v'_{i_1}, \ldots, v'_{i_k}\}$ and performs edits using Photoshop [3] or a text-to-image generator [34, 39]. Image-space inverse rendering techniques [31, 52] are then applied to identify which intrinsic modalities have been modified. Our goal is to propagate these edits across time, generating an edited video sequence $V' = \{v'_1, v'_2, \ldots, v'_T\}$, while preserving the untouched intrinsic properties and maintaining temporal coherence.

As illustrated in Fig. 2, **V-RGBX** consists of three main components: (1) (RGB→X) Starting from sequential RGB frames, an **inverse rendering model** $D(\cdot)$ estimates the corresponding intrinsic channels, such as **A**lbedo, **N**ormal, **M**aterial, and **I**rradiance, $D(V) = \{V_A, V_N, V_M, V_I\} \in \mathbb{R}^{T \times 3 \times H \times W}$, where T, H, and W denote the number of frames, height, and width of the video, respectively. The material channel consists of surface attributes such as roughness, metallic, and ambient occlusion; (2) After keyframe edits are applied, the modified intrinsic modalities of neighboring frames in $D(V)$ can no longer be directly used for conditioning during video generation. To address this, we combine the edited intrinsic channels at the keyframes with randomly interleaved, untouched intrinsic channels from other frames to construct a streamed *intrinsic conditioning* sequence $V'_X = \text{Sample}(D(V))$; (3) (X→RGB) At the core of our framework, a **forward rendering network** $R(\cdot)$ synthesizes the output video conditioned on both the streamed intrinsic conditioning and the edited keyframes $V' = R(\{v'_{i_1}, \ldots, v'_{i_k}\}, V'_X)$.

3.2. Inverse Rendering (RGB→X)

The inverse rendering model $D(\cdot)$ predicts intrinsic channels from the input video $V = \{v_1, \ldots, v_T\}$, including albedo, normal, material, and irradiance. We adopt a Diffusion Transformer (DiT) backbone [46] and condition the denoising process with $h_t = [x_t^z \| \mathcal{E}(V)]$, where x_t^z denotes the initial noisy latent, $\mathcal{E}(\cdot)$ is the frozen Wan-VAE encoder, and $\|$ represents channel-wise concatenation. The target modality name is encoded as a text prompt using
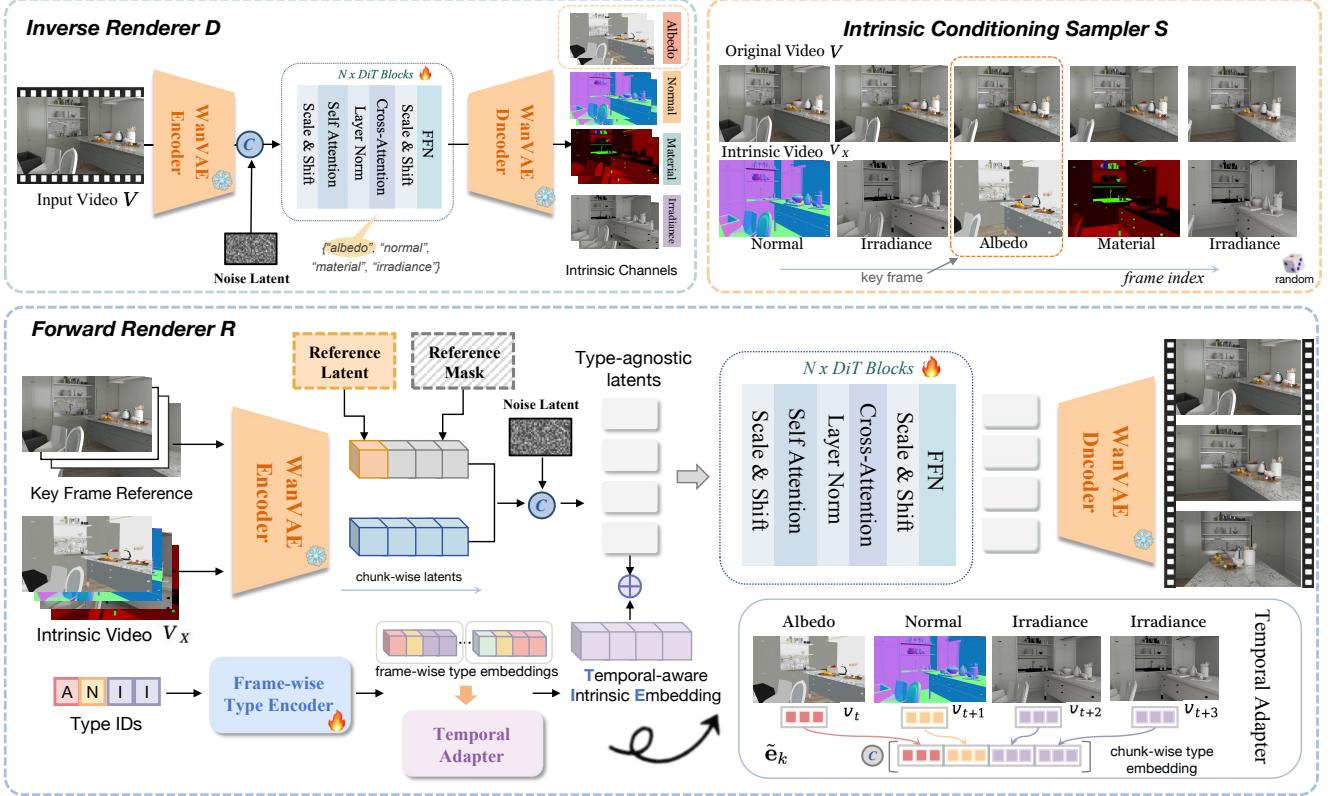
Figure 2. **The overall architecture of V-RGBX.** Our framework consists of three parts: (1) Inverse Renderer D, which decomposes the input video into albedo, normal, material, and irradiance channels; (2) Intrinsic Conditioning Sampler S, which interleaves edited keyframe intrinsics with non-conflicted random intrinsic frames to form a unified intrinsic conditioning video; and (3) Forward Renderer R, which integrates the intrinsic video, keyframe reference, and temporal-aware intrinsic embeddings to synthesize the output RGB video and consistently propagate intrinsic properties across time.

CLIP embeddings [38]. We fine-tune the backbone with the velocity-prediction objective [36] for improved training stability. The denoised latent is finally decoded by the frozen Wan-VAE decoder into a three-channel intrinsic image corresponding to the target modality.

3.3. Intrinsic-aware Conditioning

After keyframe edits are applied, the unedited intrinsic modalities of neighboring frames in $D(V)$ can no longer be directly used for conditioning during video generation, as they may introduce conflicts with the edited keyframes. Previous approaches (e.g., GenProp [30] and VACE [23]) handle such cases by inserting empty tokens to compensate for missing frames in the conditioning sequence. However, this strategy leads to substantial memory overhead in our setting, where multiple intrinsic channels are jointly modeled, and limits scalability to additional intrinsic modalities. We propose to ensure temporal coherence and cross-modality consistency by interleaving the decomposed intrinsic channels into a single conditioning sequence:

$$V_X' = \text{Sample}(\{V_A, V_N, V_M, V_I\}) = \{v_1^x, v_2^x, \ldots, v_T^x\},$$

where $\text{Sample}(\cdot)$ denotes a temporal multiplexing operation that alternates intrinsic modalities over time.

In the intrinsic-driven editing setting, we assume that one or more intrinsic modalities, denoted as $\mathcal{M}_t$, are modified at keyframe t. When sampling the conditioning signals for edited keyframes, we randomly draw from the edited modalities $\mathcal{M}_t$, while for the remaining frames, we sample from the *non-conflicted* intrinsic modalities of that frame. A modality is considered *conflicted* if it is affected by user edits in any keyframe, as its altered content may introduce inconsistencies when used as conditioning input. Formally,

$$v_t^x = \begin{cases} \text{RandomSample}(\mathcal{M}_t), & t \in \{v_{i_1}', ..., v_{i_k}'\}, \\ \text{RandomSample}(\{A,N,M,I\} \backslash \mathcal{K}_t), & \text{otherwise.} \end{cases}$$

This design encourages cross-modality propagation and enhances temporal stability. It naturally adapts to diverse attribute combinations and incomplete inputs. This provides a lightweight and extensible framework for intrinsic-aware video generation and editing applications.

4

3.4. Forward Rendering with Intrinsic-aware Conditioning

Given the edited RGB keyframes and the composed intrinsic conditioning sequence, our forward rendering model synthesizes a temporally coherent video that faithfully reflects the keyframe edits while preserving the intrinsic properties encoded in the conditioning inputs. We build upon WAN [46], a large-scale video generation backbone. Our model integrates the intrinsic conditioning sequence in a unified and interleaved sequence modulated with a modality-aware embedding and uses an edited keyframe for further conditioning.

Keyframe referencing. When keyframes are edited in the RGB domain, the edited keyframes can serve as visual guidance alongside the intrinsic video conditioning. To enable this, we align their temporal length with the original video sequence by extending the keyframes with empty tokens.

The extended sequence, Σ, is then processed by the Wan-VAE encoder to obtain a reference latent that aligns in shape with the backbone latent space. We concatenate the noisy latent x_t^z, the embedding of the conditioning interleaved intrinsic sequence, $\mathcal{E}_{\text{VAE}}(V_X')$, and the embedding of reference sequence $\mathcal{E}_{\text{VAE}}(\Sigma)$ along the channel dimension to compose the input to the diffusion model. Specifically,

$$\mathbf{z}_t = [x_t^z \| \mathcal{E}_{\text{VAE}}(V_X') \| \mathcal{E}_{\text{VAE}}(\Sigma)]. \tag{1}$$

By encoding the keyframes as reference signals and injecting them jointly with the intrinsic conditioning, the model learns to capture the overall visual content of the scene and compensates for information not explicitly represented in the intrinsic channels.

Temporal-aware intrinsic embedding. Following our backbone video generator, the encoder compresses every four consecutive frames into one latent chunk. However, these four conditioning frames may correspond to different intrinsic modalities. To embed an intrinsic conditioning sequence into the temporally compressed latent space of the DiT backbone, we propose a **T**emporal-aware **I**ntrinsic **E**mbedding (TIE) that *packs* per-frame modality embeddings within the chunk dimension, preserving both temporal order and modality identity.

Let each frame i be assigned a modality index $m_i \in \{\text{albedo}, \text{normal}, \text{irradiance}, \text{material}, \ldots\}$, we compute its embedding as

$$\mathbf{e}_i = \mathbf{W}\,\phi(m_i), \quad \mathbf{e}_i \in \mathbb{R}^d, \tag{2}$$

where $\phi(\cdot)$ is a one-hot modality indicator and the type encoder $\mathbf{W}$ is a learnable embedding matrix. Similar to [46],

we then construct a packed embedding for each latent chunk k via a temporal adapter,

$$\tilde{\mathbf{e}}_k = \begin{cases} [\mathbf{e}_1 \| \mathbf{e}_1 \| \mathbf{e}_1 \| \mathbf{e}_1], & k = 1, \\ [\mathbf{e}_{4k-3} \| \mathbf{e}_{4k-2} \| \mathbf{e}_{4k-1} \| \mathbf{e}_{4k}], & k > 1, \end{cases} \quad \tilde{\mathbf{e}}_k \in \mathbb{R}^{4d}, \tag{3}$$

where $\|$ denotes concatenation along the feature dimension.

After patchification, each latent chunk $\mathbf{z}_t^k \in \mathbb{R}^{H' \times W' \times 4d}$ is modulated by its corresponding packed embedding:

$$\tilde{\mathbf{z}}_t^k = \mathbf{z}_t^k + \gamma\,\tilde{\mathbf{e}}_k^*, \tag{4}$$

where $\tilde{\mathbf{e}}_k^*$ represents the spatial broadcasting of the modality embedding, and γ is a constant scaling factor empirically set to 1. This formulation enables each latent chunk to carry explicit modality information while maintaining temporal order within the chunk.

Training and Inference. Following the inverse rendering stage, we adopt the *v*-prediction objective [36] for training. For simplicity, text conditioning is omitted in this setting. During training, the keyframe reference is randomly dropped with a probability of $p_{\text{drop}} = 0.3$. At inference time, classifier-free guidance [19] is applied to the reference conditioning to balance fidelity and edit consistency.

4. Experiments

We extensively evaluate **V-RGBX** on a diverse set of synthetic and real-world datasets. Details of our experimental setup are provided in Sec. 4.1. We conduct comprehensive comparisons and ablation studies across three core tasks: (**RGB→X**) *inverse rendering* (Sec. 4.2), (**X→RGB**) *intrinsic-aware video synthesis* (Sec. 4.3), and *keyframe-based video editing* (Sec. 4.4). Finally, we demonstrate the versatility of our framework through a range of *intrinsic-driven video editing applications* in Sec. 4.5.

4.1. Experiment settings

We train **V-RGBX** on an internal synthetic dataset rendered from 127 Evermotion [52] interior scenes, producing 171K frames with paired supervision across RGB, albedo, normal, material, and irradiance channels. All models are trained at a resolution of 832×480 using 32 NVIDIA A100 (80GB) GPUs. Both the Inverse Renderer D and Forward Renderer R are initialized from the pretrained Wan 2.1 [46] T2V-1.3B DiT backbone and trained for 27K and 12K iterations, respectively, with a learning rate of 2×10^{-4}. The type encoder $\mathbf{W}$ is trained from scratch.

For evaluation, we select 85 videos from unseen Evermotion scenes and 85 videos from RealEstate10K [54] to assess performance on both synthetic and real-world data. To maintain consistency with baseline methods, only the
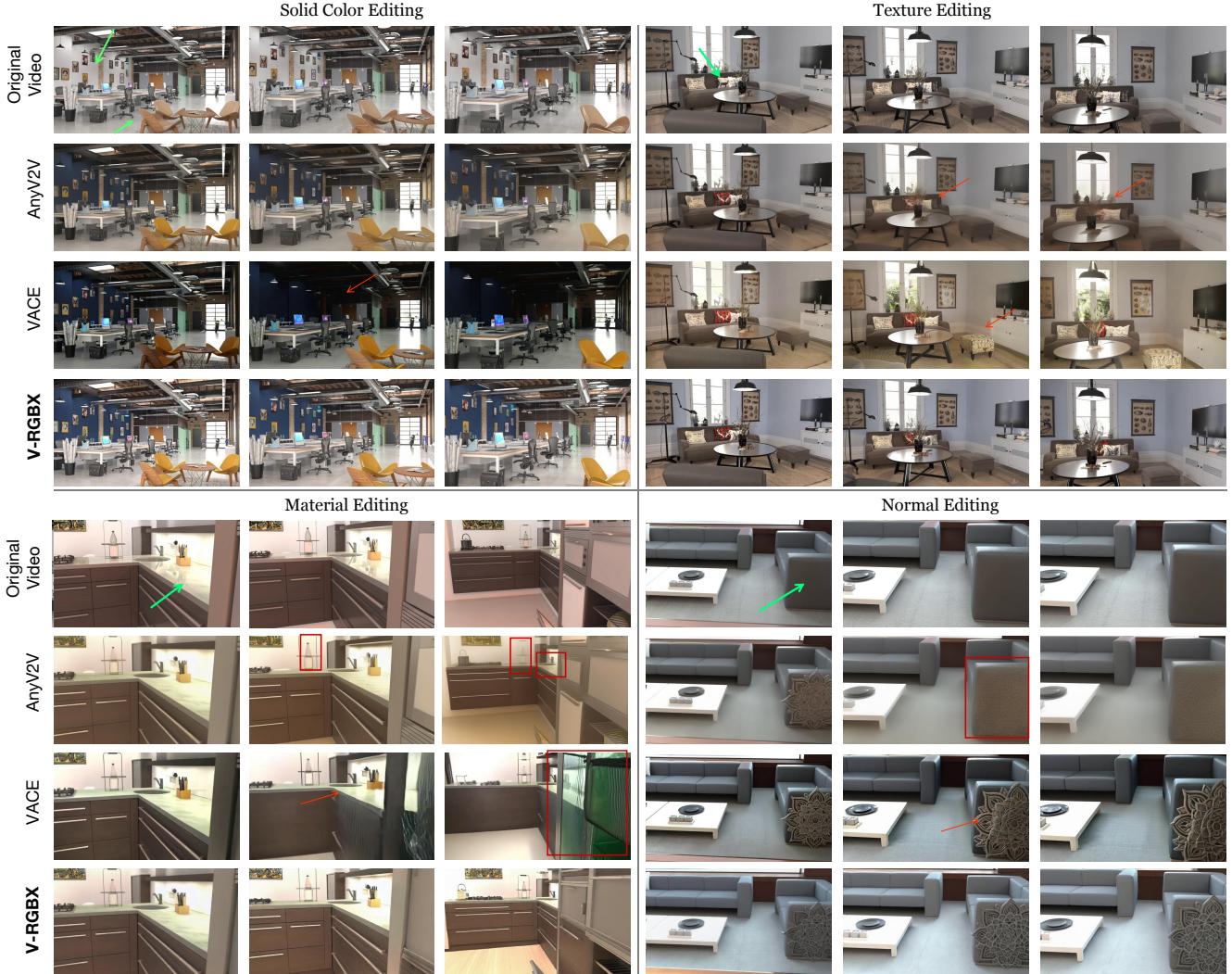
Figure 3. **Qualitative comparison on intrinsic-aware editing.** We show the results of keyframe-edit video propagation based on a single intrinsic channel. AnyV2V[25] and VACE[23] suffer from property drifting, exhibiting unexpected evolution over time, and fail to achieve accurate disentanglement among different intrinsic channels. In contrast, our method consistently propagates the intrinsic-aware edits throughout the sequence. (Green arrows mark the edited regions in the original video's keyframe, whereas red arrows and boxes denote artifacts generated by the baseline methods.)

first frame of each video is used as the keyframe input by default.

Metrics. We evaluate both forward and inverse rendering performance using PSNR, SSIM, and LPIPS [53]. For albedo evaluation, a three-channel scaling factor is applied via least-squares optimization before computing the metrics to account for global scaling ambiguity. To assess video generation quality, we use FVD [44], and for temporal coherence, we adopt the smoothness score from VBench [21].

4.2. Inverse rendering (RGB→X)

We compare our method against RGB↔X [52] and DiffusionRenderer [29] for intrinsic channel extraction on our

synthetic dataset. Since RGB↔X is an image-based approach, we perform inverse rendering on each frame independently. Consequently, its temporal consistency is not directly comparable to other video-based baselines. DiffusionRenderer does not estimate lighting, so we report results only on the remaining intrinsic channels. As shown in Table 1, **V-RGBX** outperforms other baselines in all intrinsic modalities for frame-wise pixel-aligned accuracy. We show qualitative comparisons of the predicted intrinsic channels in supplemental material, showing that our method maintains better temporal coherence than other baselines.
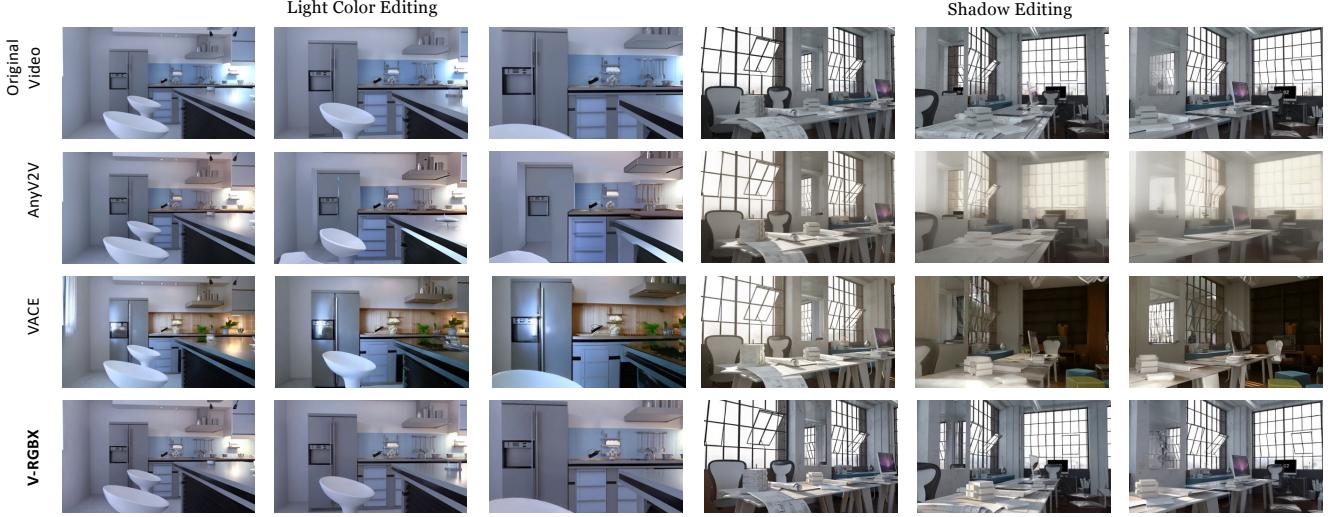
Figure 4. **Qualitative comparison on relighting.** We evaluate our method on light color and shadow editing tasks. AnyV2V exhibits geometry and appearance drifting as generation progresses, while VACE struggles to disentangle lighting effects, leading to unintended changes in other channels and even introducing new assets. Our approach naturally performs relighting in both tasks and produces results that closely match the ground truth.
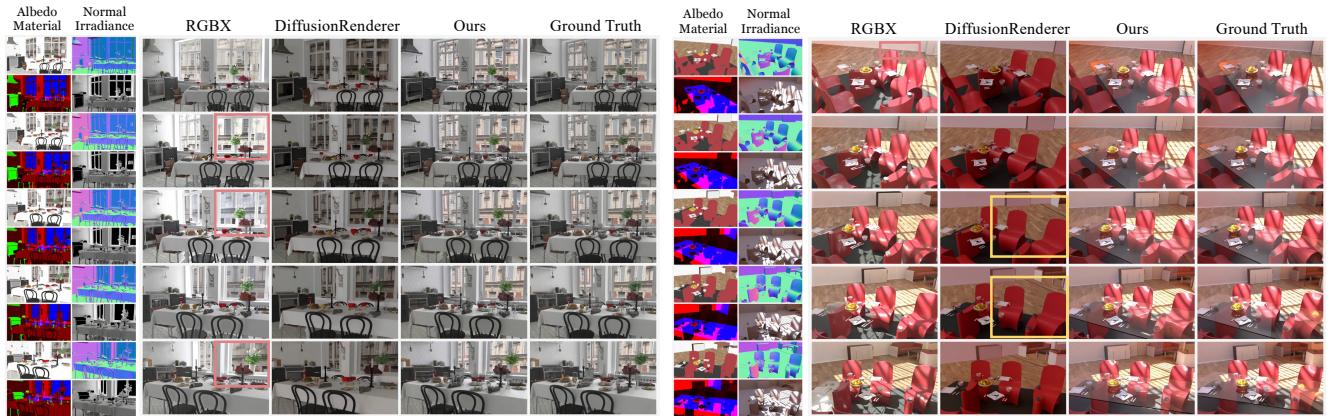

Figure 5. **Qualitative comparisons for the X→RGB task.** Pink frames highlight temporal or spatial inconsistencies observed in RGBX results, while yellow frames indicate inaccurate or missing shadow modeling produced by DiffusionRenderer. Our method achieves temporal coherence, reliable shadow and light modeling, and overall more faithful scene generation from X to video.

4.3. Intrinsic-aware video generation (X→RGB)

We first evaluate **V-RGBX** on the synthetic dataset, where ground-truth intrinsic channels are available. Under our interleaved conditioning setting, one intrinsic modality is randomly sampled for each frame, along with a randomly selected RGB frame as the reference image. Since our synthetic dataset only provides rendered irradiance maps, we follow [37] to estimate an environment map from the first frame of each video sequence for fair comparison with DiffusionRenderer [29]. As shown in Fig. 5, although the estimated environment maps are reasonable, Diffusion-Renderer fails to reproduce realistic reflections in the test scenes, resulting in misleading high smoothness scores (Ta-

ble 2).

We further conduct an ablation study to assess the impact of keyframe reference images by comparing V-RGBX variants with and without reference conditioning. As evidenced by both quantitative metrics and visual comparisons, incorporating a reference image helps the model better capture the visual style of the scene and compensates for information not explicitly represented in the intrinsic channels.

As a unified framework, **V-RGBX** can also be evaluated in an end-to-end manner via cycle consistency. On both synthetic and real datasets, we perform inverse rendering followed by forward rendering using our approach and the baseline methods, then compare the generated video se-

Table 1. **Quantitative evaluation on the RGB→X task.** We show that our method significantly outperforms previous approaches in terms of albedo, normal, and irradiance estimation.

Method	Albedo		Normal		Irradiance	
	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
RGBX	14.04	0.2872	19.44	0.1800	11.92	0.2994
DiffusionRenderer	17.40	0.3002	21.04	0.1817	-	-
V-RGBX (ours)	**17.73**	**0.2406**	**21.59**	**0.1407**	**19.94**	**0.2187**

quences with the original inputs. Consistent with previous settings, RGB↔X [52] operates in a frame-wise manner, while DiffusionRenderer uses environment maps estimated by [37]. As shown in Table 3, **V-RGBX** achieves the best overall performance among all baselines, demonstrating superior temporal coherence and frame-wise pixel-aligned accuracy on both real and synthetic datasets. DiffusionRenderer again reports spuriously high smoothness scores, as its forward-rendered outputs exhibit faded in qualitative evaluations. Additional qualitative comparisons are provided in the supplementary materials.

4.4. Keyframe based video conditioning

Our method supports flexible conditioning through intrinsic channels during inference. In this experiment, we evaluate **V-RGBX** under different conditioning schemes: (1) excluding a specific modality χ from the conditioning sequence, and (2) using χ only in the first frame, where in both settings, χ can be either albedo or irradiance. These settings simulate real editing scenarios in which certain modalities cannot be used due to conflicts with the edited keyframes. As shown in Table 4, when an RGB reference frame is provided, our trained model achieves quantitatively comparable performance as in Table 2 under different conditioning schemes, demonstrating the robustness and reliability of **V-RGBX** in real intrinsic-guided video editing scenarios.

4.5. Applications

V-RGBX enables keyframe-based, intrinsic-aware editing of input videos. In this evaluation, we use a text-driven image editing tool, NanoBanana [34], to modify selected keyframes. These edited keyframes are encoded as references and jointly condition the video generation process together with the intrinsic conditioning sampled from the video's intrinsic channels and keyframe intrinsics. In addition to Fig. 1, Fig. 3 and 4 present examples of appearance editing and scene relighting tasks, respectively. As shown, **V-RGBX** faithfully incorporates the keyframe edits while preserving the untouched intrinsic content from the original video frames, significantly outperforms the existing baselines. We include the input and output videos, edited keyframes, extracted intrinsic channels, and additional intermediate results in the supplementary material.

Table 2. **Quantitative evaluation on the X→RGB task.** We show that our method substantially surpasses previous approaches by achieving higher forward rendering accuracy, improved video quality, and smoother temporal consistency. Our method is further enhanced when the reference frame is provided.

Method	PSNR↑	SSIM↑	LPIPS↓	FVD↓	Smoothness↑
RGBX	16.53	0.7154	0.2417	1037.15	0.9469
DiffusionRenderer*	12.66	0.6475	0.3376	1015.09	**0.9883**
V-RGBX (w/o ref.)	21.48	0.7908	0.2064	401.62	0.9814
V-RGBX (ours)	**22.42**	**0.7952**	**0.1930**	**367.89**	0.9805

Table 3. **Quantitative evaluation on RGB→X→RGB cycle consistency.** We conduct comparisons on both synthetic and real-world data, where our approach achieves the best cycle consistency in inverse and forward rendering, demonstrating its robustness and high fidelity.

Dataset	Method	PSNR↑	SSIM↑	FVD↓	Smoothness↑
Evermotion	RGBX	15.29	0.7539	1099.04	0.9485
	DiffusionRenderer*	12.42	0.6311	1073.98	0.9803
	V-RGBX (ours)	**22.57**	**0.7985**	**367.61**	**0.9808**
RealEstate10K	RGBX	14.40	0.6411	2082.81	0.9307
	DiffusionRenderer*	12.53	0.6272	1643.05	0.9839
	V-RGBX (ours)	**17.88**	**0.7533**	**633.76**	**0.9870**

Table 4. **Quantitative comparison on different control strategies in the X→RGB task.** We drop one intrinsic channel (albedo or irradiance) during the X→RGB generation and find that our method still demonstrates strong robustness in rendering. When the missing channel is provided only for the first frame as reference, the results show a corresponding improvement, indicating that the model effectively propagates the guidance across the entire sequence.

Control Type X	Method	Albedo			Irradiance		
		PSNR↑	SSIM↑	FVD↓	PSNR↑	SSIM↑	FVD↓
Drop X channel	Ours (w/o ref)	17.18	0.7236	907.63	17.43	0.7350	702.16
	Ours	20.83	0.7623	549.71	21.70	0.7807	441.05
1st-Frame X-Guided	Ours (w/o ref)	20.17	0.7652	496.09	20.67	0.7775	461.53
	Ours	21.65	0.7738	427.56	21.82	0.7844	396.40

5. Discussion

In this paper, we present **V-RGBX**, an end-to-end framework for video editing with intrinsic-level control. While **V-RGBX** demonstrates strong performance in intrinsic-aware video editing, several limitations remain. First, the model is trained only on indoor synthetic datasets and may therefore struggle to generalize to out-of-distribution scenarios, such as outdoor scenes. Second, the current intrinsic conditioning samples exactly one modality per frame, which limits its ability to capture complex, multi-attribute edits on keyframes. Third, the framework relies on a pretrained video backbone [46], constraining scalability in both video length and real-time performance. Looking forward, the framework could be extended with long-range generation capabilities [28], enabling more flexible and persistent keyframe edits across time.

A. Appendix Overview

In this appendix, we provide additional details and results that are not included in the main paper due to the space limit. The attached video includes intuitive and interesting qualitative results of V-RGBX.

B. Workflow & Implementation Details

B.1. Video editing workflow explanation

Intrinsic decomposition and keyframe editing. As shown in Fig. S1, we first decompose the input RGB video into intrinsic channels, including albedo, irradiance, normal, and material. These intrinsic channels form a physically structured representation that separates appearance, illumination, and geometry, enabling more reliable and controllable video editing. Selected keyframes are edited with a text-driven image editing tool NanoBanana and then decomposed again to obtain their edited intrinsics, ensuring that user-intended modifications (e.g., material changes or relighting) are reflected in the intrinsic domain.

Intrinsic conditioning sampling. To propagate the edits beyond the keyframes, we employ an intrinsic conditioning sampler that aggregates both the original per-frame intrinsics and the edited intrinsic channels. The sampler constructs an interleaved intrinsic sequence V'_X, inserting edited intrinsic cues at the keyframe positions while preserving unmodified channels for all other frames. This provides a unified intrinsic sequence that encodes both preserved and edited content in a temporally aligned manner.

Forward rendering of edited content. The interleaved intrinsic video is then passed through our forward renderer R, which synthesizes the final edited RGB video. The edited keyframes provide both edited intrinsic cues and reference appearance keyframes, and conditioning on intrinsics leverages their structured, disentangled nature to support faithful and controllable propagation of edits. We show more qualitative results in Sec. D.4 and D.5.

B.2. RGB→X→RGB cycle workflow

We adopt an RGB→X→RGB cycle setup to assess how well the intrinsic representation retains the information needed for accurate reconstruction and for supporting reliable edit propagation. This evaluation setting provides a clear and comprehensive way to examine how appearance, geometry-related cues, and illumination are preserved when passing through each stage of our framework.

As illustrated in Fig. S2, an input RGB video is first decomposed by our inverse renderer into its intrinsic channels. The predicted intrinsic sequence is temporally consistent, and the intrinsic output of the first frame additionally serves as a keyframe to anchor the forward synthesis. These intrinsic channels are then fed into our forward renderer to reconstruct the video. By comparing the reconstructed sequence with the original input, as reported in Table 3, we evaluate how well the intrinsic space maintains pixel-level fidelity, structural detail, and temporal continuity with baseline methods. This cycle analysis also indicates the stability of intrinsic-based edits when propagated across frames. We show more qualitative results in Sec D.3.

B.3. Inference details

During inference of forward rendering, classifier-free guidance is applied to the reference branch while keeping V'_X as a shared condition:

$$\epsilon_{\text{CFG}} = \epsilon_\theta(z_t, \varnothing, V'_X) + s[\epsilon_\theta(z_t, v_{\text{ref}}, V'_X) - \epsilon_\theta(z_t, \varnothing, V'_X)], \tag{S1}$$

where the two terms denote predictions without/with the reference input, respectively. Following the notation in the main text, the reference is defined as $v_{\text{ref}} = \{v'_{i_1}, \ldots, v'_{i_k}\}$. This modulates reference-driven appearance while preserving the structural/physical priors encoded in V'_X. In our implementation, the guidance scale is set as $s = 1.5$.

C. Additional Experiments

In this section, we present additional ablation studies focusing on the two modules that most strongly affect the propagation behavior of our Forward Renderer: the Intrinsic Type Embedding (ITE) module and the Reference Condition. For each ablation, we remove the corresponding module and retrain the model from scratch under the same training iterations and hyperparameters as V-RGBX. Both quantitative and qualitative analyses are provided.

C.1. The Effectiveness of ITE Module

As shown in Tab. S1, comparing the first and second rows reveals that removing the ITE module—and thus relying solely on interleaved intrinsic conditioning—leads to consistent drops across all evaluation metrics. The qualitative results in Fig. S3 further show noticeable temporal flickering and color inconsistencies. We observe that when intrinsic channels are interleaved on a per-frame basis without explicit type disambiguation, the model may confuse modality identities across frames. Such confusion can couple signals across different channels in the color space, causing visually incorrect or unstable predictions.

After introducing the ITE module, all metrics improve, and the generated videos (Fig. S3, second column) exhibit much better temporal stability and appearance consistency. This confirms that ITE provides an effective mechanism for disentangling the intrinsic channels over time, reducing cross-channel conflicts and producing more reliable visual outcomes.
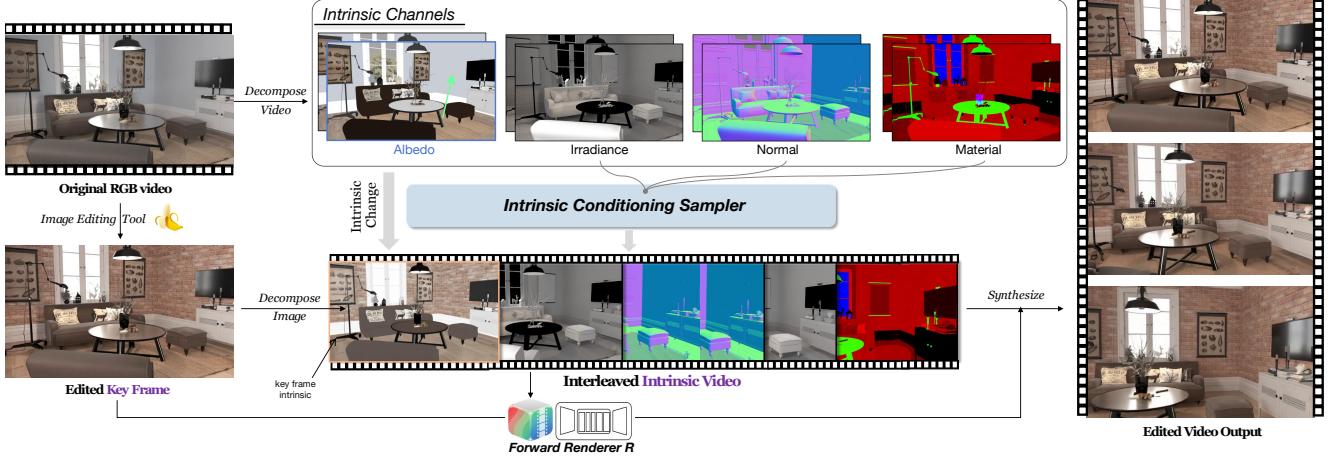
Figure S1. **Intrinsic-aware video editing workflow of V-RGBX.** Given an input video and edited keyframes, we decompose them into intrinsic channels, and the intrinsic conditioning sampler uses these representations to produce an intrinsic video. The forward renderer then synthesizes the final edited sequence using both the intrinsic video and the appearance cues provided by the edited keyframes.
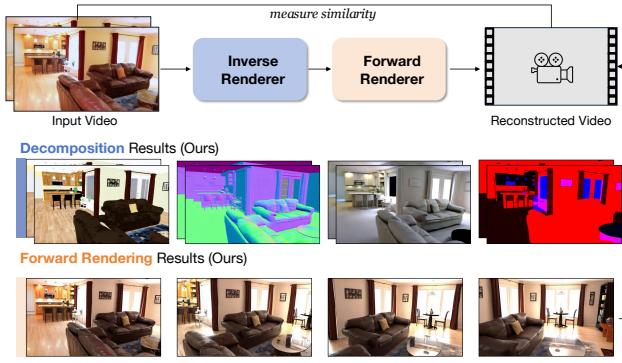

Figure S2. **Overview of RGB→X→RGB cycle workflow.** An input RGB video is first decomposed into intrinsic components by our inverse renderer, then reconstructed by the forward renderer using the predicted intrinsic sequence and a first-frame keyframe. The decomposition and forward-rendering results illustrate the quality of our intrinsic predictions and the rendered video.
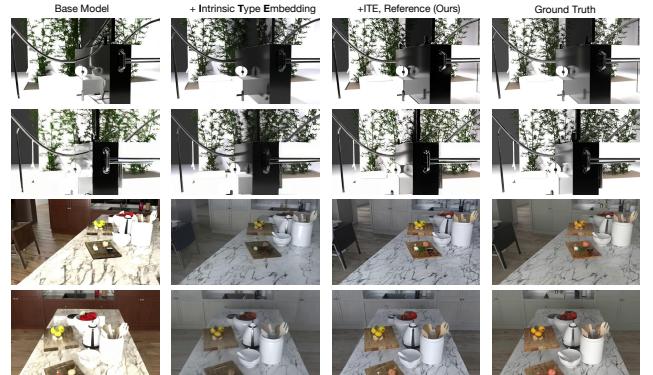

Figure S3. **Visual ablations on the Intrinsic Type Embedding (ITE) module and the reference condition.** Columns show the base model, adding ITE, adding both ITE and the reference condition (ours), and the ground truth. The intrinsic maps (top) and reconstructed RGB frames (bottom) illustrate that ITE reduces temporal and modality inconsistencies, while the reference condition further improves reflections and color fidelity.

C.2. The Effectiveness of Reference Condition

As shown in Tab. S1, adding the reference condition leads to further improvements over using ITE alone. Note that this experiment evaluates whether the model is trained with reference supervision, which differs from the evaluation in the main paper where we study whether reference images are provided at inference time. Here, the ablation aims to understand the contribution of the reference module itself.

The qualitative comparisons also show consistent gains: in the first two rows of Fig. S3, reflections become clearer and more coherent, while in the third and fourth rows, object colors and tones align more closely with the ground truth. These observations suggest that the reference condition offers important complementary cues that help correct

biases in the X→RGB mapping and significantly improve reconstruction fidelity.

D. Additional Qualitative Results

D.1. Additional RGB→X Results

In the main paper (Sec. 4.2), we have already reported quantitative results for the inverse-rendering task (RGB→X). Here, we provide additional qualitative results in Fig. S4 and Fig. S5, along with representative comparisons against baseline methods.

Fig. S4 and Fig. S5 show input videos from both synthetic and real scenes together with the intrinsic predictions

10

Table S1. **Quantitative ablations of the ITE module and the reference condition.** Adding ITE consistently improves reconstruction quality and temporal stability across all metrics, and further incorporating the reference condition yields the best overall performance, with noticeable gains in PSNR, LPIPS, FID, FVD, and smoothness.

Method	ITE	Key Reference	PSNR↑	LPIPS↓	SSIM↑	FID↓	FVD↓	Smooth.↑
Base (no added modules)	×	×	20.96	0.2372	0.7818	37.81	532.21	0.9769
+ ITE	✓	×	21.47	0.2149	**0.7994**	35.39	405.79	0.9802
+ ITE + Reference (ours)	✓	✓	**22.42**	**0.1930**	0.7952	**29.83**	**367.89**	**0.9805**

produced by V-RGBX, including albedo, normal, material, and irradiance channels. We further compare our method with RGBX and DiffusionRenderer in Fig. S6. Consistent with the quantitative findings, V-RGBX yields more stable albedo and normal reconstructions, while RGBX often exhibits temporal instability and color inconsistencies. DiffusionRenderer also shows some failure cases, such as collapsed normal maps and inaccurate color estimates. Moreover, our model demonstrates strong generalization ability, producing reliable intrinsic decompositions even under challenging real-world and outdoor lighting conditions.

D.2. Additional X→RGB Results

As discussed in Sec. 4.3, we evaluate the X→RGB task, and Fig. S8 provides additional qualitative examples together with comparisons against the baseline methods. These examples show that V-RGBX handles complex lighting effects and geometric structures more reliably, producing RGB sequences with more stable shading, reflections, and temporal coherence. Overall, the supplemental results further illustrate the robustness of our approach when generating videos from intrinsic representations.

D.3. Additional RGB→X→RGB Results

As discussed in Sec. 4.3, we quantitatively evaluate the RGB→X→RGB cycle to assess whether the intrinsic representation preserves sufficient information for accurate reconstruction and reliable edit propagation. In Figs. S9 and S10, we provide additional qualitative comparisons on both synthetic and real-world videos, showing the reconstructed sequences produced by our approach and the baseline methods. Our method achieves more stable temporal behavior and better preserves scene appearance across the full cycle.

D.4. Keyframe Editing Results

As described in Sec. 4.5, we demonstrate the intrinsic-aware video editing capability of V−RGBX in the main paper. To provide a clearer view of how the edits are propagated through our intrinsic pipeline, we include the complete set of intermediate results in Fig. S11. Specifically, we visualize the input video frames, the edited keyframes produced

by the NanoBanana tool, and the extracted intrinsic channels that jointly condition the generation process. These intermediate visualizations help illustrate how the edited albedo, normal, material, or irradiance attributes guide the final synthesis. Following the editing workflow shown in Fig. S1, V-RGBX takes the modified keyframes and intrinsic channels as conditioning signals and generates temporally consistent, intrinsically coherent outputs.

D.5. Real-world Challenging Cases

We provide additional demonstrations of our editing capability on diverse and challenging real-world scenarios. Please refer to the attached video for full results. Our evaluations cover real indoor scenes, self-captured videos, general object videos, and cases with complex lighting, showcasing robust intrinsic-aware editing performance across a wide range of real-world conditions.
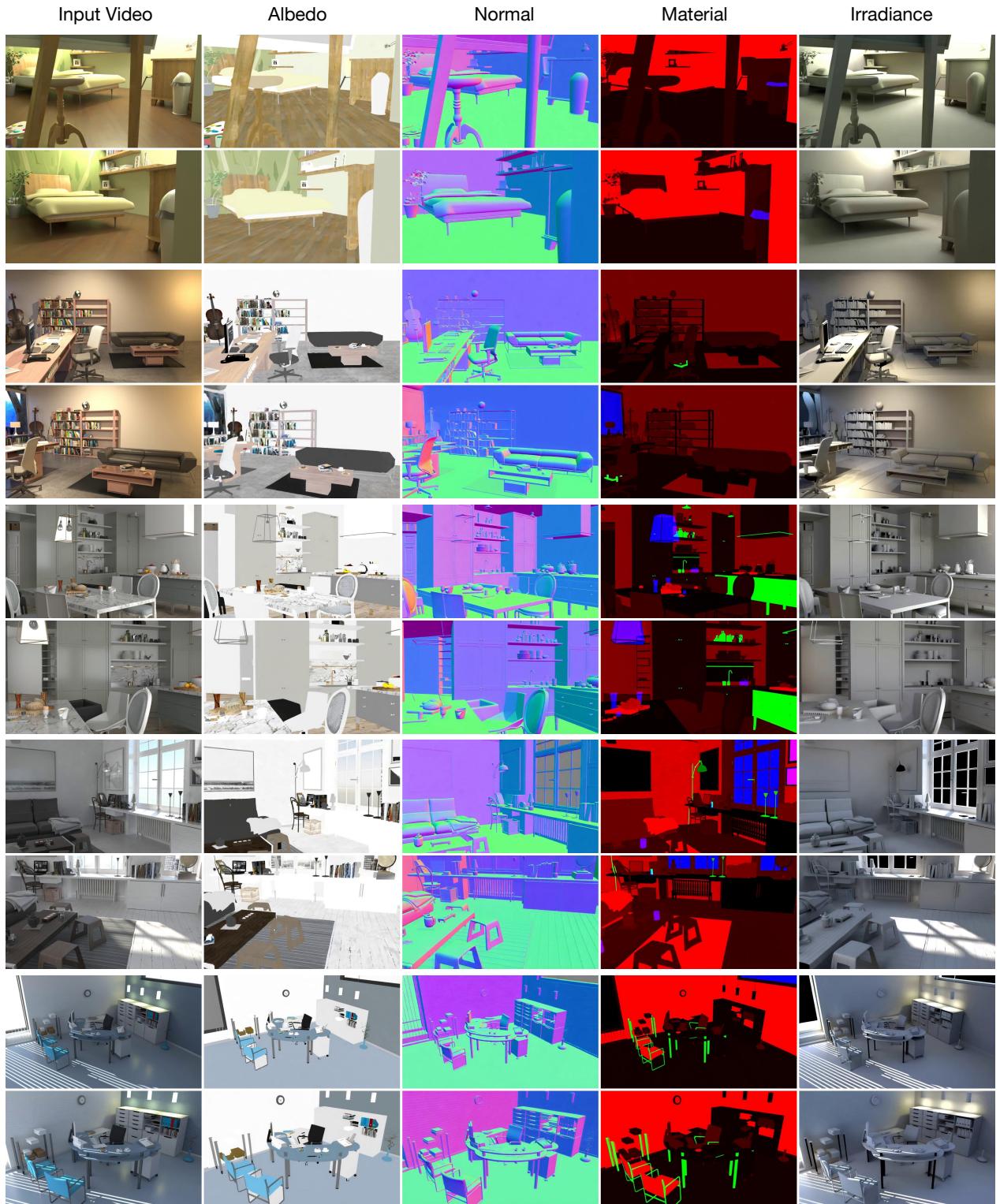
Figure S4. **RGB→X results on synthetic Evermotion scenes.** Given an input RGB video, V-RGBX decomposes it into albedo, normal, material, and irradiance channels. Each pair of rows shows two frames from the same video, and the second to fifth columns visualize the corresponding intrinsic channels, demonstrating spatially coherent and temporally stable decompositions across diverse indoor scenes.
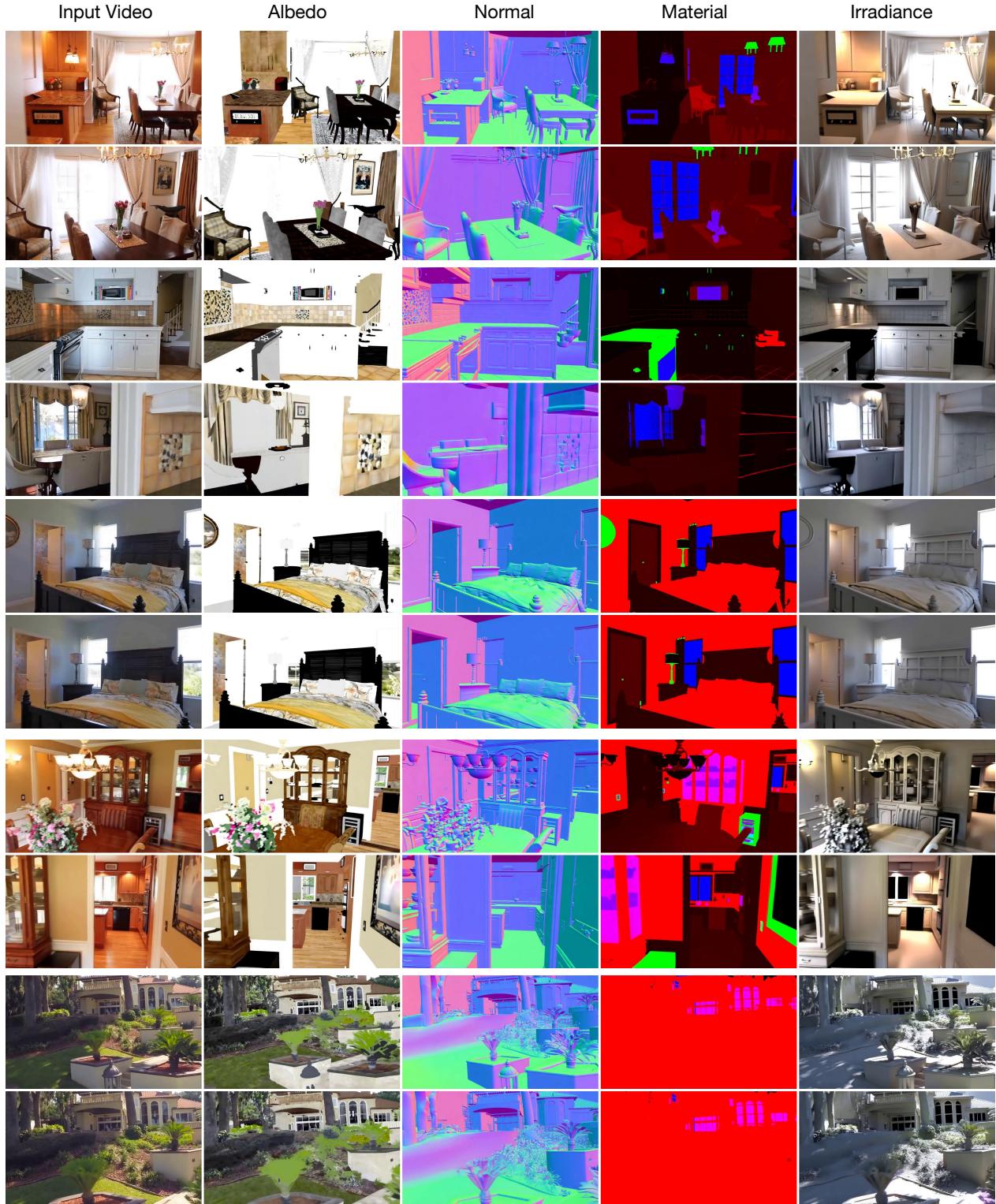
Input Video	Albedo	Normal	Material	Irradiance

Figure S5. **RGB→X results on real-world RealEstate10K videos.** Given an input RGB video, V-RGBX decomposes it into albedo, normal, material, and irradiance channels. Each pair of rows shows two frames from the same video, and the second to fifth columns visualize the corresponding intrinsic channels, demonstrating coherent and temporally stable decompositions under challenging and unseen real-world conditions, while also showing reasonable generalization to outdoor scenes.
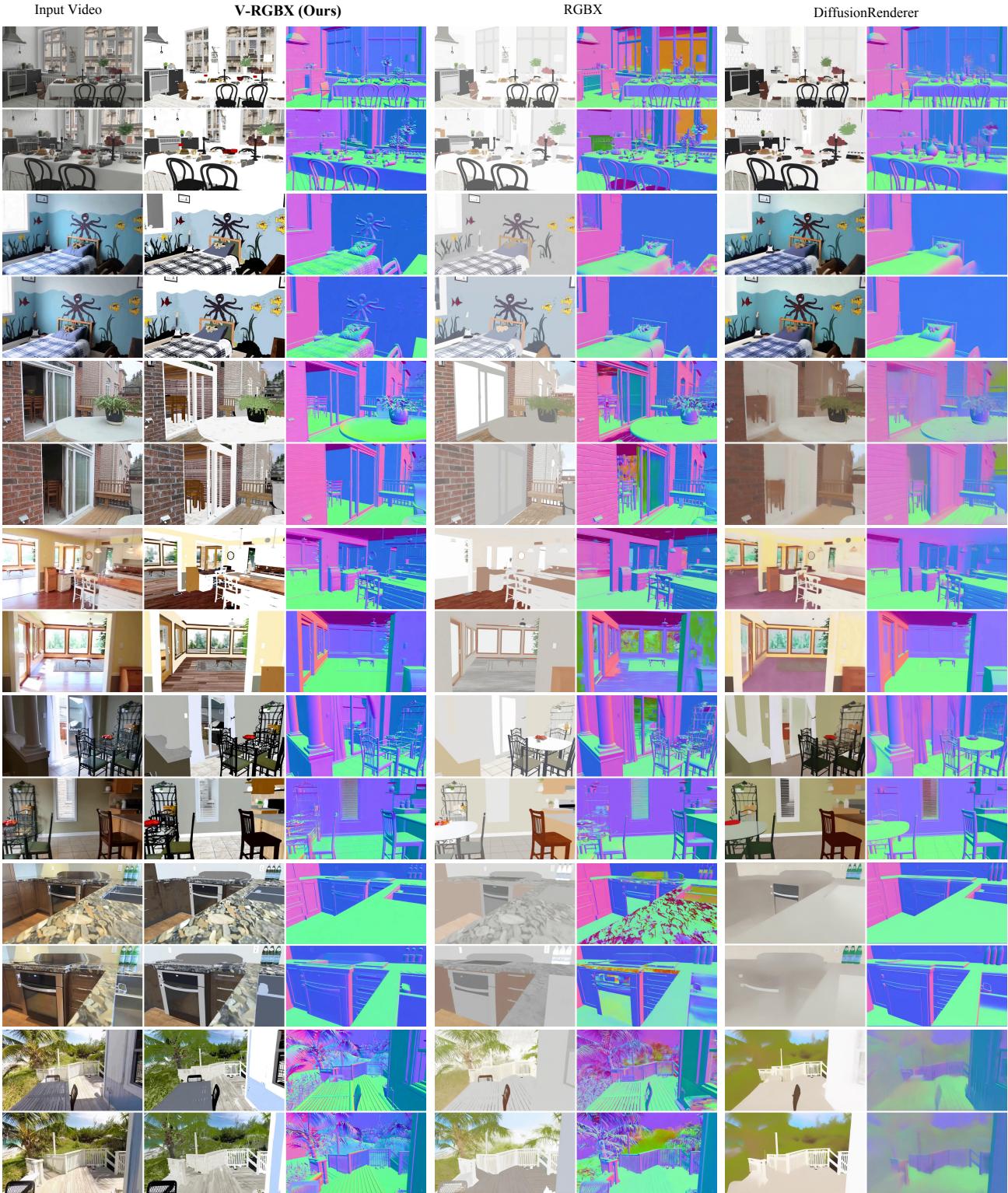
Figure S6. **Comparison of RGB→X decomposition results with baselines.** Each pair of rows shows two frames from the same input video (first column). For each method, the two columns visualize the predicted albedo and normal channels. Compared with RGBX and DiffusionRenderer, V-RGBX produces intrinsic decompositions with higher visual fidelity, more accurate albedo estimation, and more consistent normal predictions across frames.

Figure S7. **Comparison of irradiance decomposition with baselines.** The figure shows two different videos, with each pair of rows representing two frames from the same video. For each frame, the second and third columns show irradiance predictions from V-RGBX and RGBX. V-RGBX produces more accurate illumination and shadow modeling, resulting in clearer and more plausible irradiance maps.
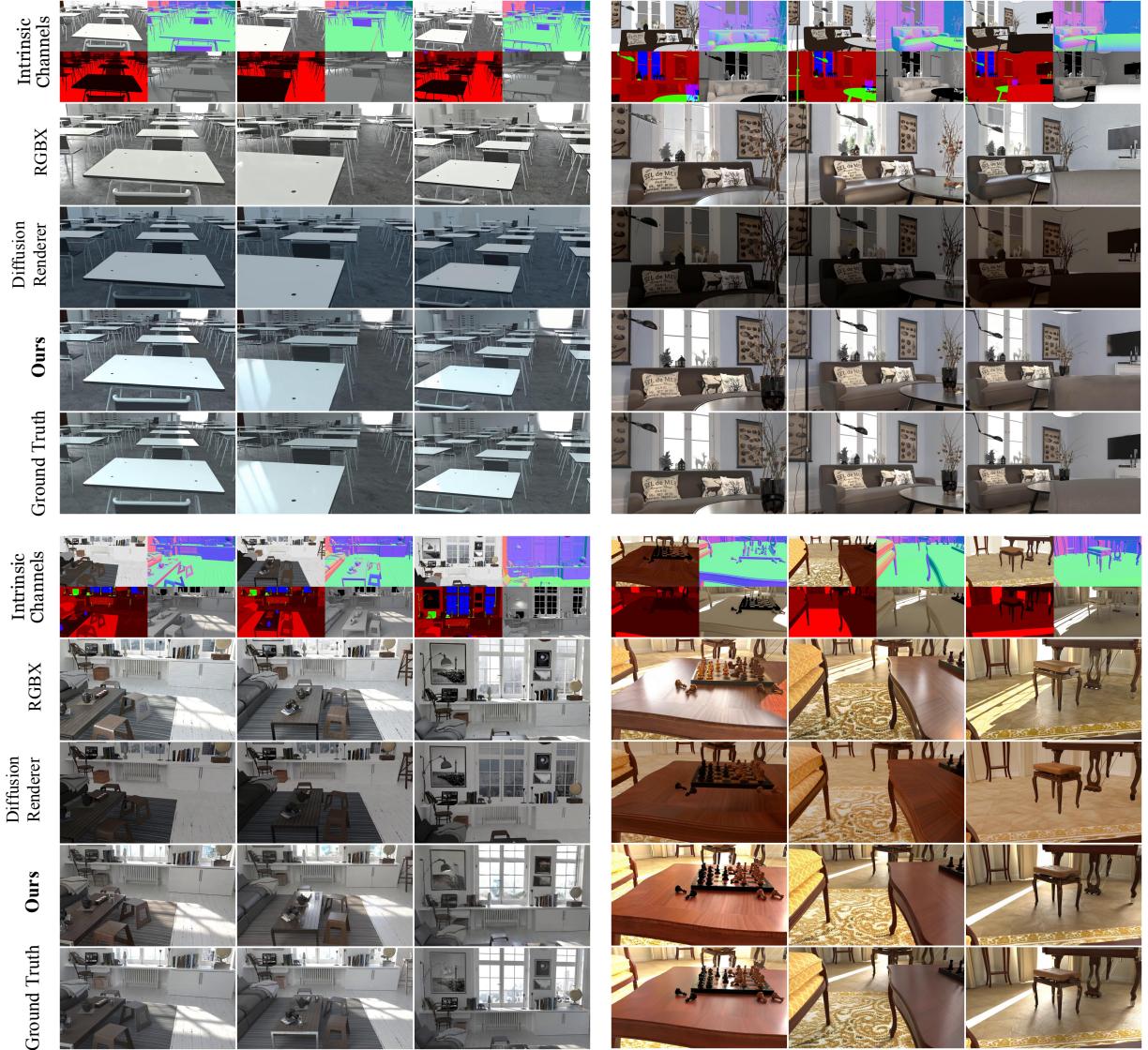

Figure S8. **More qualitative comparisons for the X→RGB task.** Each group of three columns shows three frames from the same video, while each row corresponds to a different method: intrinsic channels inputs, RGBX, DiffusionRenderer, our results, and the ground truth. The comparisons illustrate that our method performs better in scene appearance and temporal consistency across frames.
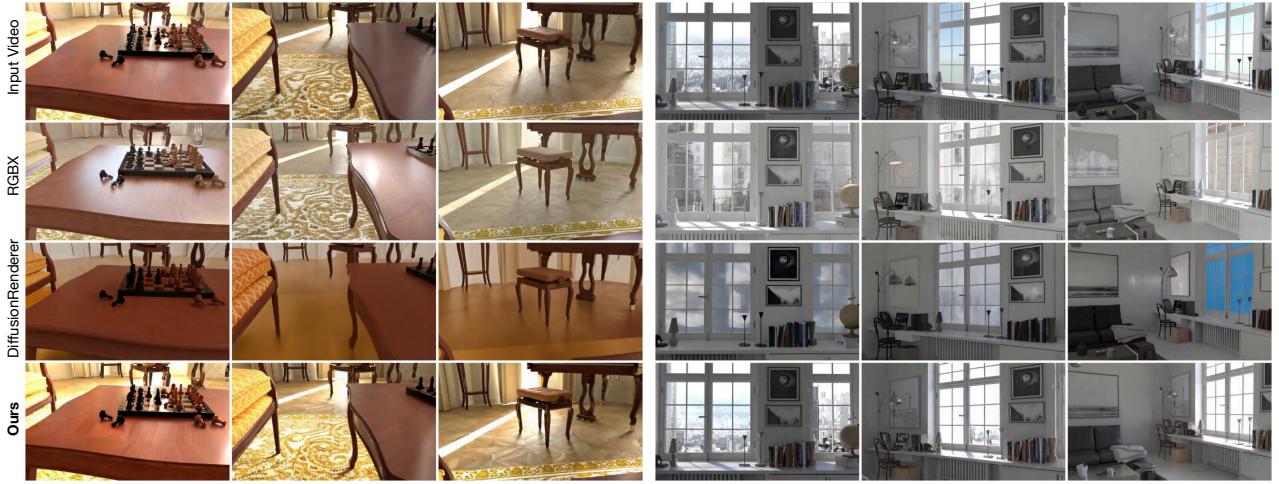
Figure S9. **RGB→X→RGB cycle results on the synthetic dataset.** Each row shows a different method (the first row is the input video as ground truth). Every three columns correspond to three frames from the same video. Our method produces reconstructions closest to the ground truth and better preserves scene appearance and structure throughout the sequence.
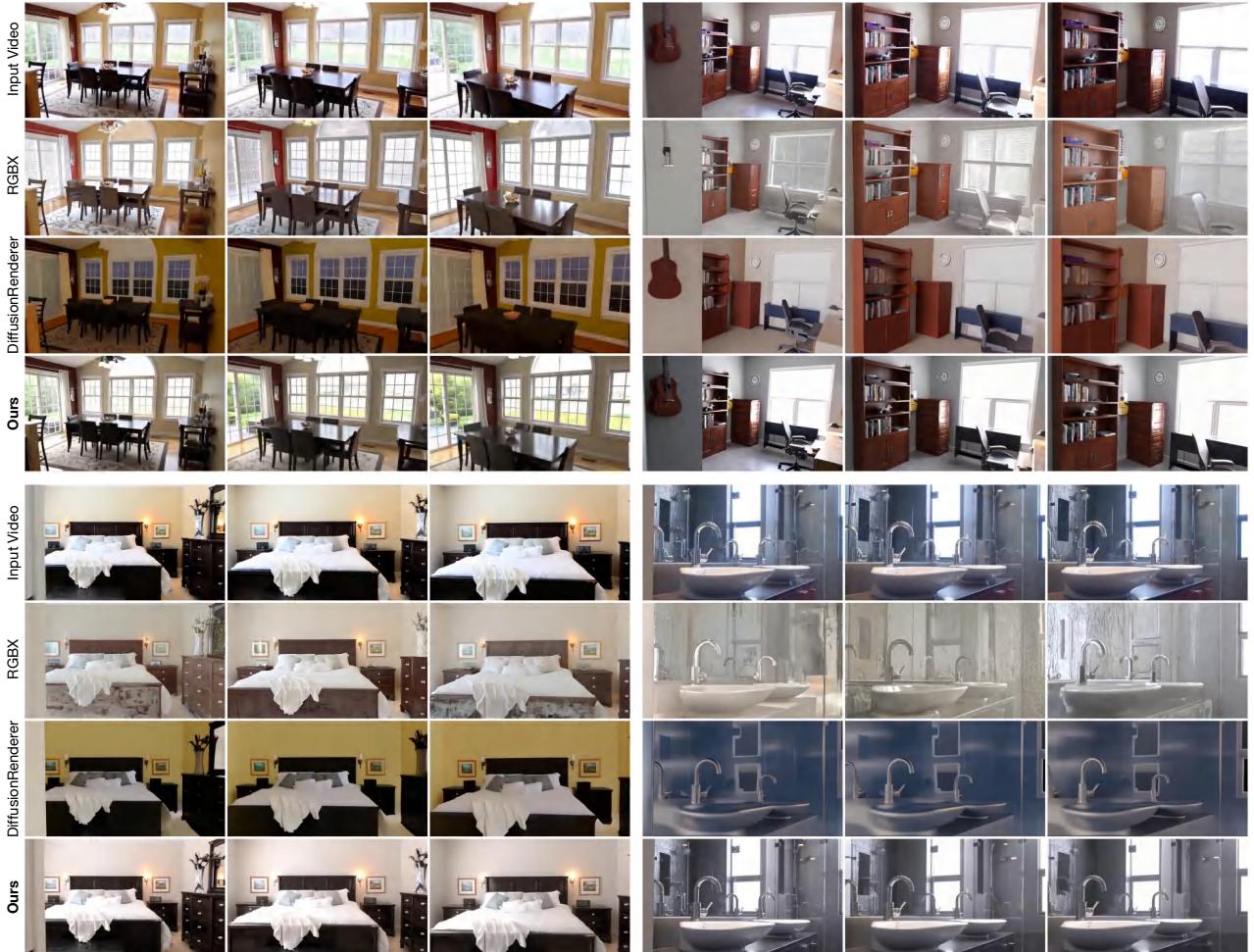

Figure S10. **RGB→X→RGB cycle results on the real-world dataset.** Each row shows a different method (the first row is the input video as ground truth). Every three columns correspond to frames from the same video. Our method gives a closer match to the ground truth.

Figure S11. **Intermediate results of the intrinsic-aware keyframe editing.** We visualize intermediate results used by V-RGBX across the following editing types: (1) solid color, (2) texture, (3) material, (4) normal, (5) light color, and (6) shadow editing. For each case, we show the original video frames, the edited keyframe produced by the NanoBanana tool, and the corresponding modified intrinsic channels (albedo, material, normal, or irradiance) that serve as conditioning inputs. These processes reveal how keyframe edits are translated into intrinsic-space modifications, which are then reliably propagated by V-RGBX to generate the final temporally consistent edited video.

References

[1] Sora. https://openai.com/sora/, 2025. 2

[2] Veo. https://deepmind.google/models/veo/, 2025. 2

[3] Adobe Inc. Adobe photoshop. https://www.adobe.com/products/photoshop.html, 2025. Version 2025, Accessed: 2025-11-11. 3

[4] Sherwin Bahmani, Ivan Skorokhodov, Guocheng Qian, Aliaksandr Siarohin, Willi Menapace, Andrea Tagliasacchi, David B Lindell, and Sergey Tulyakov. Ac3d: Analyzing and improving 3d camera control in video diffusion transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22875–22889, 2025. 3

[5] Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, David B. Lindell, and Sergey Tulyakov. Vd3d: Taming large video diffusion transformers for 3d camera control. In *ICLR*, 2025.

[6] Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. *arXiv preprint arXiv:2503.11647*, 2025.

[7] Jianhong Bai, Menghan Xia, Xintao Wang, Ziyang Yuan, Xiao Fu, Zuozhu Liu, Haoji Hu, Pengfei Wan, and Di Zhang. Syncammaster: Synchronizing multi-camera video generation from diverse viewpoints. In *ICLR*, 2025. 3

[8] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, et al. Lumiere: A space-time diffusion model for video generation. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. 2

[9] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 2

[10] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22563–22575, 2023. 2

[11] Julian Decart, Quinn Quevedo, Spruce McIntyre, Xinlei Campbell, Robert Chen, and Wachen. Oasis: A universe in a transformer. 2024. 3

[12] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *ArXiv*, abs/2105.05233, 2021. 2

[13] Alara Dirik, Tuanfeng Wang, Duygu Ceylan, Stefanos Zafeiriou, and Anna Frühstück. Prism: A unified framework for photorealistic reconstruction and intrinsic scene modeling. *arXiv preprint arXiv:2504.14219*, 2025. 3

[14] 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. *arXiv preprint arXiv:2501.03847*, 2025. 3

[15] Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, et al. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–12, 2025. 2

[16] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023. 2

[17] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. In *ICLR*, 2025. 3

[18] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*, 2022. 2

[19] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 5

[20] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022. 2

[21] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024. 6

[22] Zhitong Huang, Mohan Zhang, Renhan Wang, Rui Tang, Hao Zhu, and Jing Liao. X2video: Adapting diffusion models for multimodal controllable neural video rendering. *arXiv preprint arXiv:2510.08530*, 2025. 3

[23] 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. 2, 3, 4, 6

[24] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. 2

[25] Max Ku, Cong Wei, Weiming Ren, Huan Yang, and Wenhu Chen. Anyv2v: A plug-and-play framework for any video-to-video editing tasks. *arXiv*, 2024. 6

[26] Max Ku, Cong Wei, Weiming Ren, Huan Yang, and Wenhu Chen. Anyv2v: A tuning-free framework for any video-to-video editing tasks. *Transactions on Machine Learning Research*, 2024. Reproducibility Certification. 2

[27] Alex X. Lee, Richard Zhang, Frederik Ebert, P. Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *ArXiv*, abs/1804.01523, 2018. 2

[28] Zongyi Li, Shujie Hu, Shujie Liu, Long Zhou, Jeongsoo Choi, Lingwei Meng, Xun Guo, Jinyu Li, Hefei Ling, and

Furu Wei. Arlon: Boosting diffusion transformers with autoregressive models for long video generation. In *ICLR*, 2025. 8

[29] Ruofan Liang, Zan Gojcic, Huan Ling, Jacob Munkberg, Jon Hasselgren, Zhi-Hao Lin, Jun Gao, Alexander Keller, Nandita Vijaykumar, Sanja Fidler, and Zian Wang. Diffusion-renderer: Neural inverse and forward rendering with video diffusion models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3, 6, 7

[30] Shaoteng Liu, Tianyu Wang, Jui-Hsien Wang, Qing Liu, Zhifei Zhang, Joon-Young Lee, Yijun Li, Bei Yu, Zhe Lin, Soo Ye Kim, et al. Generative video propagation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17712–17722, 2025. 2, 3, 4

[31] Jundan Luo, Duygu Ceylan, Jae Shin Yoon, Nanxuan Zhao, Julien Philip, Anna Frühstück, Wenbin Li, Christian Richardt, and Tuanfeng Y. Wang. IntrinsicDiffusion: Joint intrinsic layers from latent diffusion models. In *SIGGRAPH 2024 Conference Papers*, 2024. 3

[32] Linjie Lyu, Valentin Deschaintre, Yannick Hold-Geoffroy, Miloš Hašan, Jae Shin Yoon, Thomas Leimküehler, Christian Theobalt, and Iliyan Georgiev. Intrinsicedit: Precise generative image manipulation in intrinsic space. *ACM Transactions on Graphics*, 44(4), 2025. 3

[33] Michaël Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *CoRR*, abs/1511.05440, 2015. 2

[34] NanaBanana. Nanabanana: Text-to-image stable diffusion generator. https://nanabanana.ai/, 2025. Accessed: 2025-11-11. 3, 8

[35] Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Christos Kaplanis, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, Stephen Spencer, Jessica Yung, Michael Dennis, Sultan Kenjeyev, Shangbang Long, Vlad Mnih, Harris Chan, Maxime Gazeau, Bonnie Li, Fabio Pardo, Luyu Wang, Lei Zhang, Frederic Besse, Tim Harley, Anna Mitenkova, Jane Wang, Jeff Clune, Demis Hassabis, Raia Hadsell, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 2: A large-scale foundation world model. 2024. 3

[36] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 4, 5

[37] Pakkapon Phongthawee, Worameth Chinchuthakun, Nontaphat Sinsunthithet, Varun Jampani, Amit Raj, Pramook Khungurn, and Supasorn Suwajanakorn. Diffusionlight: Light probes for free by painting a chrome ball. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 98–108, 2024. 7, 8

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 4

[39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[41] Sam Sartor and Pieter Peers. Teamwork: Collaborative diffusion with low-rank coordination and adaptation. In *ACM SIGGRAPH Asia Conference Proceedings*, 2025. 3

[42] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 2

[43] S. Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1526–1535, 2017. 2

[44] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. 2019. 6

[45] Carl Vondrick and Antonio Torralba. Generating the future with adversarial transformers. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2992–3000, 2017. 2

[46] 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. *arXiv preprint arXiv:2503.20314*, 2025. 2, 3, 5, 8

[47] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2, 3

[48] Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory, 2025. 3

[49] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 2

[50] Mark YU, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monoc-

ular videos via diffusion models. *arXiv preprint arXiv:2503.05638*, 2025. 3

[51] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024. 3

[52] Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Miloš Hašan. RGB↔X: Image decomposition and synthesis using material- and lighting-aware diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, New York, NY, USA, 2024. Association for Computing Machinery. 3, 5, 6, 8

[53] 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*, pages 586–595, 2018. 6

[54] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 5