VideoScene: Distilling Video Diffusion Model to Generate 3D Scenes in One Step

Supplementary Material

1. More Discussion of Preliminaries

In this section, we provide more preliminaries about the diffusion model, consistency model [28], and contextual bandit [30].

1.1. Diffusion Model

Diffusion models [11, 24, 25, 27] generate data by progressively introducing Gaussian noise to the original data and subsequently sampling from the noised data through several denoising steps. Let $p_{\text{data}}(x)$ denote the data distribution, The forward process is described by a stochastic differential equation (SDE) [27] given by

$$\mathrm{d}\mathbf{x}_t = \mu(\mathbf{x}_t, t)\mathrm{d}t + \sigma(t)\mathrm{d}\mathbf{w} \tag{1}$$

where $t \in [0, T]$, $T > 0$ denotes a fixed time horizon, $\mu(\cdot, \cdot)$ and $\sigma(\cdot)$ represent the drift and diffusion coefficients,respectively, and $\{\mathbf{w_t}\}_{t\in[0,T]}$ is the standard Brownian motion. An important property of this SDE is the existence of an associated ordinary differential equation (ODE), known as the Probability Flow (PF) ODE [27], which deterministically describes the distribution's evolution

$$\mathrm{d}\mathbf{x}_t = \left[\mu(\mathbf{x}_t, t) - \frac{1}{2}\sigma^2(t)\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)\right]\mathrm{d}t \tag{2}$$

where $\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)$ is the *score function* of the intermediate distribution $p_t(\mathbf{x}_t)$. For practical purposes [13], a simplified setting is often adopted, where $\mu(\mathbf{x}_t, t) = \mathbf{0}$ and $\sigma(t) = \sqrt{2t}$. This yields the intermediate distributions $p_t(\mathbf{x}) = p_{\text{data}}(x) \otimes \mathcal{N}(\mathbf{0}, t^2\mathbf{I})$, where $\otimes$ convolution operation. Let $\pi(\mathbf{x}) = \mathcal{N}(0, T^2\mathbf{I})$ and after a sufficient noise adding process, the final distribution $p_T(\mathbf{x})$ will be closed to $\pi(\mathbf{x})$. Sampling involves solving the empirical PF ODE:

$$\frac{d\mathbf{x}_t}{dt} = -t\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t) \tag{3}$$

starting from a sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, T^2\mathbf{I})$ and running the ODE backward procedure with Numerical ODE solver like Euler [26] and Heun [13] solver, we can obtain a solution trajectory $\{\hat{\mathbf{x}}_t\}_{t\in[0,T]}$ and thus get a approximate sample $\hat{\mathbf{x}}_0$ from the data distribution $p_{\text{data}}(\mathbf{x})$. The backward process is typically stopped at $t = \epsilon$ to avoid numerical instability, where ϵ is a small positive number, and $\hat{\mathbf{x}}_\epsilon$ is treated as the final approximate result.

1.2. Consistency Model

Consistency model [28] is a novel class of models that supports both one-step and iterative generation, providing a trade-off between sample quality and computational efficiency. The consistency model can be trained either by distilling knowledge from a pre-trained diffusion model or independently, without relying on pre-trained models. Formally, given a solution trajectory $\{\hat{\mathbf{x}}_t\}_{t\in[0,T]}$ sampled from Eq. 2, we define the *consistency function* as $\mathbf{f} : (\mathbf{x}_t, t) \mapsto \mathbf{x}_\epsilon$. a consistency function exhibits a self-consistency property, meaning that its outputs remain consistent for any pair of $(\mathbf{x}_t, t)$ points that lie along the same PF ODE trajectory. The goal of a consistency model is to approximate this consistency function $\mathbf{f}$ with $\mathbf{f}_\theta$. Given any consistency function $\mathbf{f}(\cdot, \cdot)$, it must satisfy $\mathbf{f}(x_\epsilon, \epsilon) = x_\epsilon$, implying that $\mathbf{f}(\cdot, \epsilon)$ acts as the identity function. This requirement is referred to as the *boundary condition*. It is imperative for all consistency models to adhere to this condition, as it is pivotal to the proper training of such models. There are several simple way to implement the *boundary condition*, for example we can parameterize $\mathbf{f}_\theta$ as

$$\mathbf{f}_\theta(\mathbf{x}, t) = c_{\text{skip}}(t)\mathbf{x} + c_{\text{out}}(t)F_\theta(\mathbf{x}, t) \tag{4}$$

where $c_{\text{skip}}(t)$ and $c_{\text{out}}(t)$ are differentiable functions such that $c_{\text{skip}}(\epsilon) = 1$ and $c_{\text{out}}(\epsilon) = 0$. This parameterization ensures that the consistency model is differentiable at $t = \epsilon$, provided that $F_\theta(x, t)$, $c_{\text{skip}}(t)$, and $c_{\text{out}}(t)$ are all differentiable, which is crucial for training continuous-time consistency models. Once a consistency model $\mathbf{f}_\theta(\cdot, \cdot)$ is well-trained, samples can be generated by first sampling from the initial distribution $\hat{x}_T \sim \mathcal{N}(0, T^2\mathbf{I})$, and then evaluating the consistency model for $\hat{x}_\epsilon = \mathbf{f}_\theta(\hat{x}_T, T)$. This generates a sample in a single forward pass through the consistency model. Additionally, the consistency model can be evaluated multiple times by alternating between denoising and noise injection steps to improve sample quality, thus offering a flexible trade-off between computational cost and sample quality. This multi-step procedure also holds significant potential for zero-shot data editing applications.

1.3. Contextual Bandit Algorithm

The Multi-Armed Bandit (MAB) problem, originally introduced by [30], is a fundamental model in sequential decision-making under uncertainty. It is named after the analogy of a gambler trying to maximize rewards from multiple slot machines (or "arms"), each with an unknown probability distribution of payouts. At each step, the agent must decide which arm to pull, aiming to maximize the cumulative reward over time. The core difficulty lies in addressing the exploration-exploitation dilemma: the agent needs to explore different arms to learn their reward dis-

Algorithm 1 3D-Aware Leap Flow Distillation

Input: 3D dataset $\mathcal{D}$, initial model parameter θ, learning rate η, one-step ODE solver $\Phi(\cdot)$, distance metric $d(\cdot,\cdot)$, EMA rate μ, noise schedule α_t, σ_t, timestep interval k, diffusion optimization timesteps T', and encoder $E(\cdot)$
Repeat
  Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$ and $t_{n+1} \in [0, T']$
  Sample $(I^i_{Input}, c^i) \sim \mathcal{D}, \quad i = \{0, 1\}$
  $t_n \leftarrow t_{n+1} - k$
  Render images $\{I_{\text{Render}}\}_{\tau=1}^{\mathcal{T}} = g(S(I^0_{\text{Input}}, c^i), o(c^i)), \quad i = \{0, 1\}$
  $\mathbf{x}^r_0 \leftarrow E(\{I_{\text{Render}}\}_{\tau=1})$
  $\mathbf{x}^r_{t_{n+1}} \leftarrow \alpha_{t_{n+1}} \mathbf{x}^r_0 + \sigma_{t_{n+1}} \boldsymbol{\epsilon}$
  $\hat{\mathbf{x}}^{\phi}_{t_n} \leftarrow \mathbf{x}^r_{t_{n+1}} + (t_n - t_{n+1}) \Phi(\mathbf{x}^r_{t_{n+1}}, t_{n+1}; \phi)$
  $\mathcal{L}_D(\theta, \theta^-; \Phi) \leftarrow d(\mathbf{f}_\theta(\mathbf{x}^r_{t_{n+1}}, t_{n+1}), \mathbf{f}_{\theta^-}(\hat{\mathbf{x}}^{\phi}_{t_n}, t_n))$
  $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \mathcal{L}_D(\theta, \theta^-; \Phi)$
  $\theta^- \leftarrow \text{stopgrad}(\mu\theta^- + (1-\mu)\theta)$
Until convergence

tributions while simultaneously exploiting the best-known arm to achieve immediate gains.

Balancing exploration and exploitation is crucial because exploration uncovers potentially better options, while exploitation ensures short-term performance. Overemphasizing exploration can waste resources on suboptimal choices, whereas overly exploiting known options risks missing higher rewards. This trade-off is central to the design of MAB algorithms.

MAB problems can be broadly categorized into two types: context-free bandits and contextual bandits. Context-free bandits have been extensively studied, with popular algorithms such as the ϵ-greedy strategy [18] and the Upper Confidence Bound (UCB) algorithm [1]. These approaches assume that the rewards are solely determined by the arm selection, without considering additional information. In contrast, contextual bandits extend this framework by incorporating side information, or "context," to model the expected reward for each arm. Contextual bandits leverage the context as input features, enabling a more nuanced understanding of the reward function. For instance, algorithms like LinUCB [20] and Thompson Sampling for linear models [2] assume that the expected reward is a linear function of the context. However, in practice, this linearity assumption often fails for complex, non-linear environments.

To overcome this limitation, many works [3] have integrated deep neural networks (DNNs) with contextual bandit frameworks, significantly enhancing their representation power. In our approach, we adopt a convolutional neural network (CNN) for contextual bandit algorithm to dynamically determine the optimal denoising timestep in video inference. Specifically, we model this problem as follows:

- State Representation: The environment state is defined by the input video latent $\mathbf{x}^r_0$, capturing structural and perceptual details of the video.
- Agent and Policy: The agent is modeled using a CNN policy network, which employs a probabilistic policy $\pi_\psi(t|\mathbf{x}^r_0; \psi)$ to select the timestep t.
- Action Selection: The action corresponds to the choice of a timestep $t \in [0, T']$, representing the level of noise to add during the denoising process.
- Reward Signal: Feedback is provided in the form of a reward signal, defined as the negative mean squared error loss ($\mathcal{L}_{MSE}$) between the denoised output and the ground truth. This reward quantifies the quality of the denoising process for the chosen t.
- Policy Update: The policy network updates its parameters ψ using the observed rewards, gradually learning to select the optimal t for different contexts.

By framing timestep selection as a contextual bandit problem, our method adaptively balances structural preservation and artifact correction during video inference, achieving robust and high-quality results across diverse scenarios.

2. Additional Implementation Details

We present the pseudo-code for 3D-aware distillation in Algorithm 1. We also provide details for additional experiments.

Datesets. To further validate our strong generalizability, we test our method on the NeRF-LLFF [22], Sora [6], and more challenging outdoor datasets Mip-NeRF 360 [4] and Tank-and-Temples dataset [16]. For video-to-3D application, we also evaluate our method on the Mip-NeRF 360 [4] and Tank-and-Temples dataset [16].

Video Metrics. We utilize VBench [12] to evaluate the performance of our model by comparing it against several state-of-the-art, open-source video frame interpolation models. VBench provides a comprehensive analysis of video generation quality by decomposing it into 16 distinct evaluation metrics, enabling a detailed and multi-faceted as-

Figure 1. Visual results of the generative ability. We highlight the generated regions in the red boxes in the novel generated views.

sessment of model performance. For our evaluation, we focus on key metrics such as Aesthetic Quality, Subject Consistency, and Background Consistency, which offer critical insights into the visual appeal and temporal coherence of the generated frames.

Aesthetic Quality measures the visual appeal of the generated video. Utilizing the LAION aesthetic predictor [17], it gauges the artistic and aesthetic value perceived by humans for each video frame. This score reflects various aesthetic dimensions, such as the layout, the richness and harmony of colors, photorealism, naturalness, and the artistic quality of the video frames. Subject Consistency assesses whether the appearance of a subject remains visually stable and coherent across all frames of a video. This metric is computed by evaluating the similarity of DINO [7] features extracted from consecutive frames. Background Consistency evaluates the temporal consistency of the background scenes by calculating CLIP [23] feature similarity across frames.

To comprehensively evaluate the quality of the generated videos, we included additional metrics from VBench in the supplementary material, including Motion Smoothness, Dynamic Degree, and Imaging Quality. Motion Smoothness measures the fluidity of motion within the generated video, evaluating how well the movement follows realistic, natural trajectories. This metric assesses whether the video adheres to the physical laws governing motion in the real world. By utilizing motion priors in the video frame interpolation model [21], it quantifies the temporal smoothness of the generated motions. Dynamic Degree is estimated by RAFT [29] to indicate the temporal quality of generated videos. In our setting of still-scene video generation, an excessively high Dynamic Degree indicates unnecessary motion of objects within the scene, while an overly low Dynamic Degree suggests prolonged static periods interrupted by abrupt changes in certain frames. Both scenarios are undesirable outcomes for our task. Imaging Quality refers to the level of distortion present in the generated frames and is assessed using the MUSIQ [14] image quality predictor, which has been trained on the SPAQ [10] dataset.

Implementation Details for Video-to-3D Application. For the video-to-3D application, we evaluate the 3D reconstruction performance of our method on the Mip-NeRF 360 [4] and Tanks-and-Temples datasets [16]. Starting with two input images and corresponding camera poses estimated from DUSt3R [32], we first generate a continuous video sequence interpolating between the two frames. From this sequence, we extract intermediate frames by sampling every seventh frame, resulting in seven new views from novel perspectives. These sampled frames are then processed using InstantSplat [9] for Gaussian optimization-based 3D reconstruction from the generated novel views.

To assess the quality of our approach, we compare it against SparseNeRF [31], the original 3DGS [15], and DNGaussian [19], with per-scene optimization serving as the benchmark. For quantitative evaluation, we report standard novel view synthesis (NVS) metrics, including PSNR, SSIM [33], and LPIPS [38].

3. Additional Experiments and Analysis

3.1. More Visual Results

We present additional visual results of our VideoScene framework in Fig. 9, showcasing its performance across di-

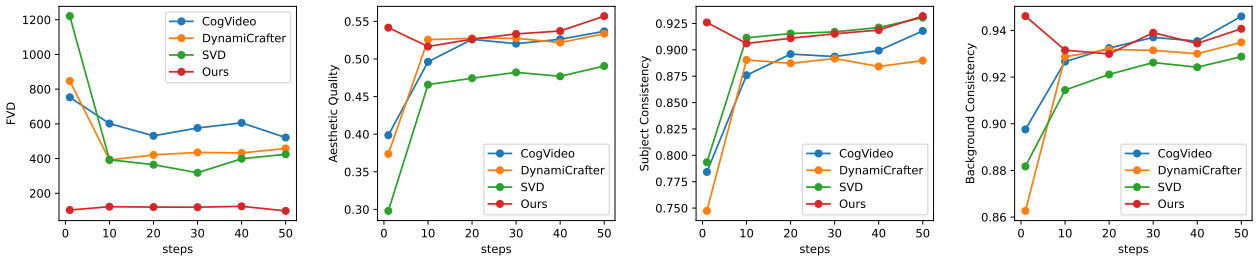

Figure 2. **Quantitative comparison across steps.** We evaluate the results of CogVideo, DynamiCrafter, Stable Video Diffusion (SVD), and VideoScene across 1, 10, 20, 30, 40, and 50 steps. VideoScene not only outperforms the other methods but also demonstrates remarkable consistency, with its 1-step results closely approximating its 50-step results, whereas other methods exhibit a significant decline in performance over fewer steps.

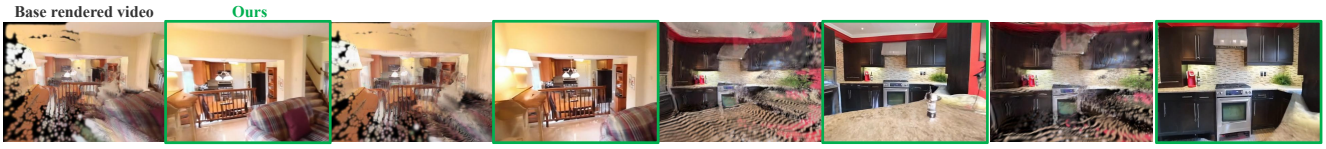

Figure 3. Comparisons with base renderings with severe artifacts.

verse datasets, including NeRF-LLFF [22], Sora [6], Mip-NeRF 360 [4], and Tanks-and-Temples [16]. These examples highlight the strong generalization capability of our method, effectively adapting to novel and out-of-distribution scenarios, whether indoor or outdoor.

We also provide additional visual results in Fig. 1 to further illustrate the generative capability of our model. When the input consists of two images with significantly different viewpoints, the intermediate regions often lack direct coverage by either input image. In such cases, a model must rely on its generative ability to synthesize these unseen areas. As highlighted by the red boxes in Fig. 1, VideoScene successfully generates novel content for these unseen regions. This demonstrates not only the strong generative capacity of our model but also its ability to generalize effectively while maintaining high fidelity in reconstructing previously unobserved areas.

3.2. More Quantitative Comparison Results

We provide comprehensive quantitative comparisons with baseline methods in Fig. 2, 6. In Fig. 2, we evaluate the performance of CogVideo [35], DynamiCrafter [34], Stable Video Diffusion (SVD) [5], and our VideoScene across different inference steps. The results demonstrate that VideoScene not only surpasses other methods in generation quality but also achieves results comparable to their 50-step outputs in just one step. In contrast, the one-step outputs of other methods fall significantly behind their 50-step counterparts, highlighting the efficiency and effectiveness of our approach.

Table 1. Quantitative comparison on Mip-Nerf 360 and Tank-and-Temples datasets. We report the quantitative metrics with two input views for each scene.

Method	PSNR↑	SSIM↑	LPIPS↓
Mip-NeRF 360			
3DGS	10.36	0.108	0.776
SparseNeRF	11.47	0.190	0.716
DNGaussian	10.81	0.133	0.727
InstantSplat	11.77	0.171	0.715
Ours	**13.37**	**0.283**	**0.550**
Tank and Temples			
3DGS	9.57	0.108	0.779
SparseNeRF	9.23	0.191	0.632
DNGaussian	10.23	0.156	0.643
InstantSplat	10.98	0.381	0.619
Ours	**14.28**	**0.394**	**0.564**

In Fig. 6, we further evaluate our method across multiple dimensions using metrics from VBench [12], providing a more systematic and holistic validation of our generative quality. Notably, the Dynamic Degree metric assesses both the dynamic motion of individual objects in the scene and overall camera motion. Our method carefully balances these aspects, preserving consistent camera motion while minimizing unstable object movements, resulting in a well-

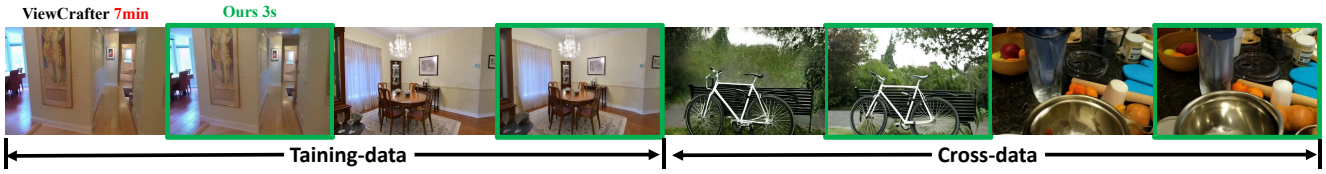

Figure 4. Comparisons with 3D-aware diffusion model ViewCrafter.

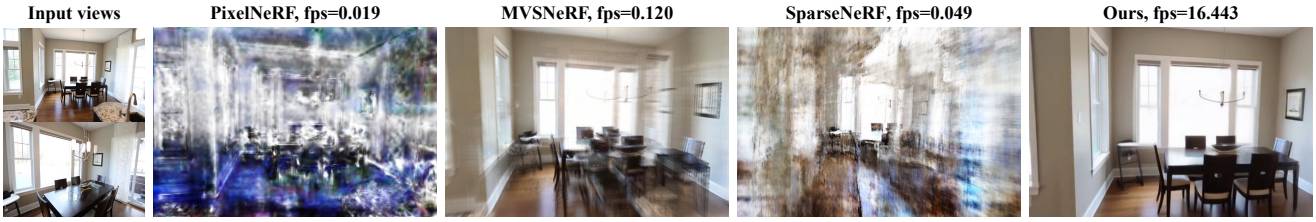

Figure 5. Comparisons with NeRF-based methods.

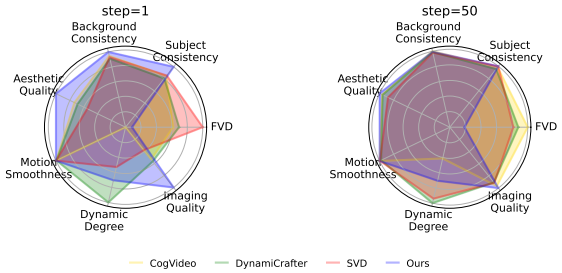

Figure 6. **Quantitative comparison across additional dimensions.** We further evaluate the 1-step and 50-step results by incorporating additional dimensions from the VBench metrics.

balanced intermediate Dynamic Degree value. In comparison, DynamiCrafter exhibits higher values due to its inability to maintain relative object stability, leading to excessive motion. Conversely, CogVideo shows lower values, as it often produces videos with prolonged static periods interrupted by abrupt transitions, particularly between the first and second halves. These observations underscore the robustness and balanced performance of our approach.

3.3. More Qualitative Comparison Results

In Fig. 3, we compare our VideoScene with MVSplat base renderings to show its effectiveness. In Fig. 4, we compare with another 3D-aware diffusion model [37], and in Fig. 5, we show more visual comparison with NeRF-based methods [8, 31, 36].

3.4. Video-to-3D Applications

We evaluate the geometric consistency of our generated frames to assess their suitability for downstream tasks such as 3D reconstruction. For this purpose, we utilize InstantSplat [9], a 3D Gaussian Splatting (3DGS) method built on DUSt3R [32], which generates Gaussian splats from sparse, unposed images. Using this approach, we use VideoScene to generate video frames from given two input views and optimize the generated frames for 3D Gaussian representations. We also compare our method against existing per-scene optimization techniques, including Instantsplat [9], DNGaussian [19], 3DGS [15], and SparseNeRF [31]. The results, presented in Tab. 1 and Fig. 7, demonstrate that our approach effectively preserves the geometric structure of the scene, avoiding issues such as the multi-face problem. Furthermore, our method exhibits strong generative capabilities, reconstructing regions beyond the coverage of input views.

Table 2. User study on the layout stability, smoothness, visual realism, and overall quality score in a user study, rated on a range of 1-10, with higher scores indicating better performance.

Method	Layout Stability	Smoothness	Visual Realism	Overall Quality
Stable Video Diffusion [5]	6.48	7.29	6.75	7.13
DynamiCrafter [34]	7.02	7.01	6.02	6.68
CogVideoX [35]	7.83	7.53	7.33	7.50
Ours	**8.39**	**8.91**	**9.52**	**8.82**

3.5. User Study

For the user study, we show each volunteer five samples of generated video using a random method. They can rate in four aspects: (1) *layout stability.* Users assess whether the scene layout in the video is spatially coherent and consistent. (2) *smoothness.* Users observe whether the frame rate is stable, whether actions are smooth, and whether there are any stuttering or frame-skipping issues. (3) *visual realism.* Users rate the similarity between the generated video and a real video. (4) *overall quality*. All aspects are on a scale of 1-10, with higher scores indicating better performance. We collect results from 30 volunteers shown in Table 2. We find users significantly prefer our method over these aspects.

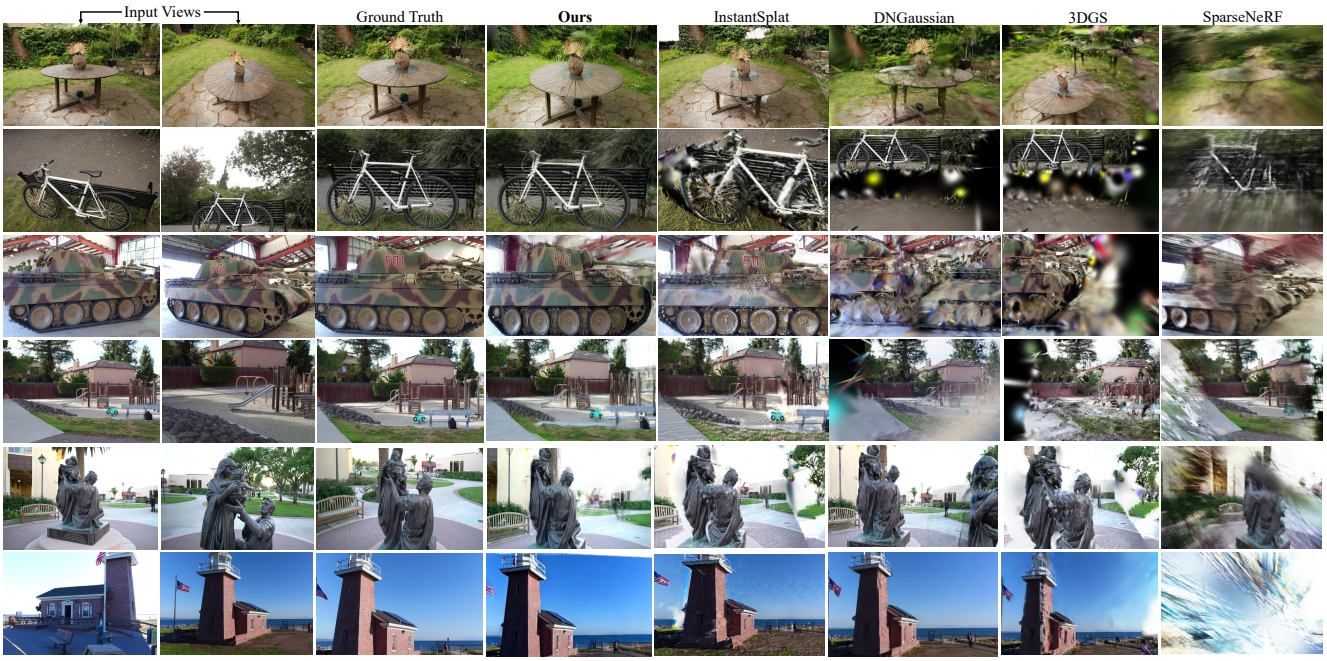

Figure 7. Qualitative comparison on Mip-Nerf 360 and Tank-and-Temples. With two sparse views as input, our method achieves much better reconstruction quality compared with baselines.

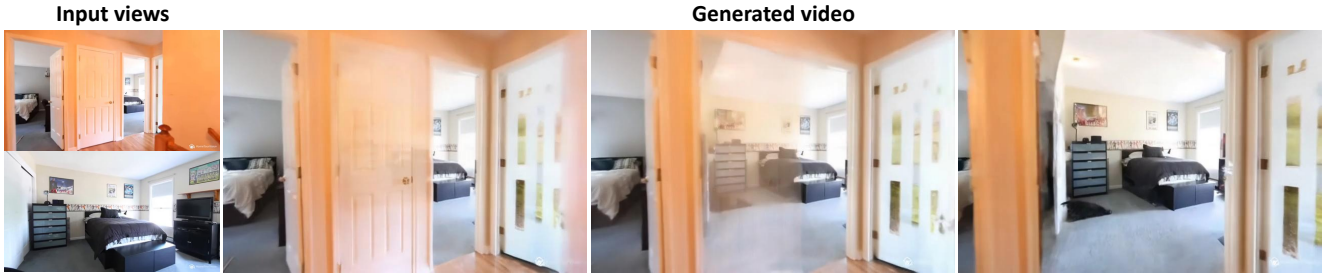

Figure 8. Fail case of passing directly through the closed door.

3.6. Failure Case

Significant semantic disparities between input views lead to failure cases (see Fig. 8). The generated video passes directly through the closed door rather than navigating around it to enter the room.

4. More Discussions

4.1. Discussion of Empirical Runtime

We provide the runtime comparison in Tab 3. DynamiCrafter is efficient due to smaller model size and lower frame rates and resolutions, but ours is still much faster.

4.2. Discussion of limited computational resources

Tab. 4 presents the comparison of memory costs on a single A100. The primary consumer of computational resources is the video diffusion model itself, which is inherently unavoidable. Leap flow distillation, as a strategy for video training, incurs a computational cost comparable to that of video diffusion training, without introducing significant additional overhead.

Table 3. Empirical runtime comparisons.

Method	SVD	DynamiCrafter	CogVideoX-5B	ViewCrafter	VideoScene (**Ours**)
Runtime (s)	933.89	21.14	179.45	206.13	**2.98**
Frames	25	16	**49**	25	**49**

Table 4. Comparison on memory costs.

Description	Video Backbone (CogVideoX)	Leap Flow Distillation	DDPNet	Total
Training Cost	$\sim$ 66 GB	$\sim$ 10 GB	$\sim$ 0.02 GB	$\sim$ 76 GB

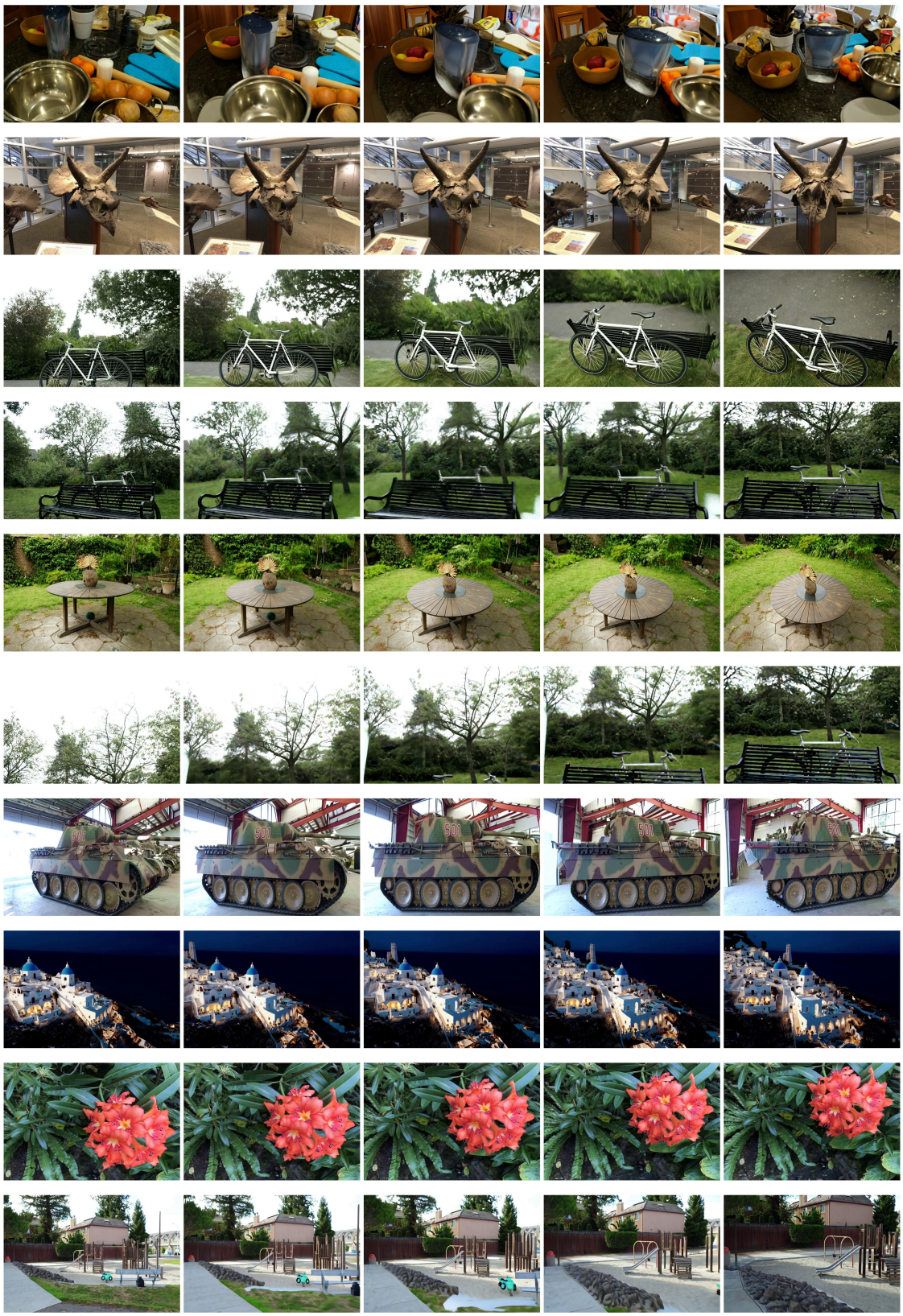

Figure 9. **Visual results of VideoScene.** We show visual results on NeRF-LLFF [22], Sora [6], Mip-NeRF 360 [4], and Tank-and-Temples dataset [16] datasets. The first and last columns represent the input views, while the intermediate columns depict the generated views.

References

[1] Rajeev Agrawal. Sample mean based index policies by o(log n) regret for the multi-armed bandit problem. *Advances in applied probability*, 27(4):1054–1078, 1995. 2

[2] Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *ICML*, pages 127–135. PMLR, 2013. 2

[3] Yikun Ban, Jingrui He, and Curtiss B Cook. Multi-facet contextual bandits: A neural network perspective. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 35–45, 2021. 2

[4] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 2, 3, 4, 7

[5] 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. 4, 5

[6] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators, 2024. 2, 4, 7

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, 2021. 3

[8] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14124–14133, 2021. 5

[9] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds. *arXiv preprint arXiv:2403.20309*, 2, 2024. 3, 5

[10] Yuming Fang, Hanwei Zhu, Yan Zeng, Kede Ma, and Zhou Wang. Perceptual quality assessment of smartphone photography. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3677–3686, 2020. 3

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 1

[12] 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 *CVPR*, 2024. 2, 4

[13] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, pages 26565–26577, 2022. 1

[14] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5148–5157, 2021. 3

[15] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 3, 5

[16] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36 (4):1–13, 2017. 2, 3, 4, 7

[17] LAION-AI. Aesthetic predictor, 2024. Accessed: 2024-11-20. 3

[18] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. *Advances in neural information processing systems*, 20, 2007. 2

[19] Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20775–20785, 2024. 3, 5

[20] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010. 2

[21] Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. Amt: All-pairs multi-field transforms for efficient frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[22] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2, 4, 7

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 3

[24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1

[25] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1

[26] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1

[27] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based

generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1

[28] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. 1

[29] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 3

[30] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933. 1

[31] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *ICCV*, pages 9065–9076, 2023. 3, 5

[32] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, pages 20697–20709, 2024. 3, 5

[33] 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. 3

[34] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *ECCV*, pages 399–417. Springer, 2025. 4, 5

[35] 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. 4, 5

[36] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4578–4587, 2021. 5

[37] 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. 5

[38] 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. 3