Reconstruct, Inpaint, Test-Time Finetune: Dynamic Novel-view Synthesis from Monocular Videos

Kaihua Chen[*] **Tarasha Khurana**[*] **Deva Ramanan**
Carnegie Mellon University
`https://cog-nvs.github.io/`

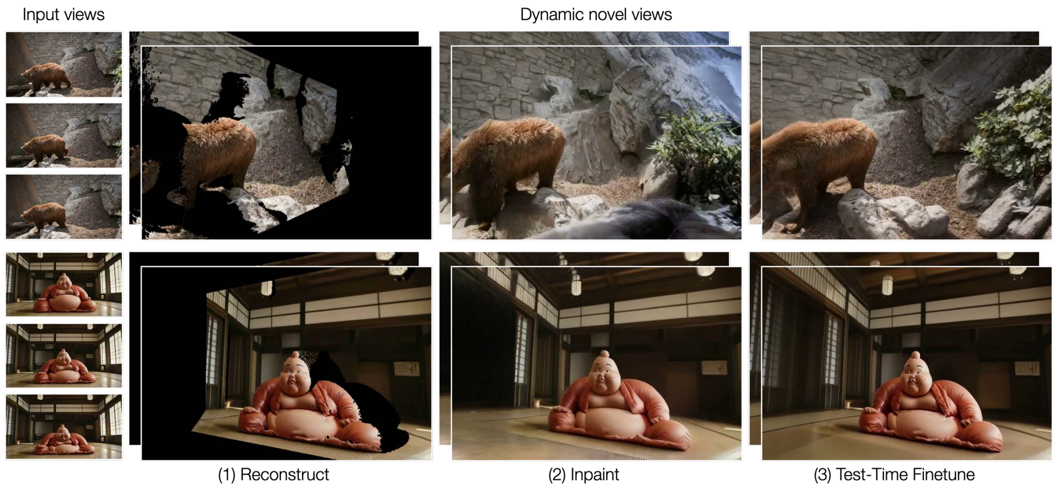

Figure 1: We present CogNVS, a video diffusion model that enables novel-view synthesis of dynamic scenes. Given an in-the-wild monocular video of a dynamic scene, we first reconstruct the scene, render it from the target novel-view and inpaint any unobserved regions. Because CogNVS can be pre-trained via self-supervision, it can also be test-time-finetuned on a given target video, enabling it to zero-shot generalize to novel domains. Our simple pipeline outperforms almost all prior state-of-the-art for dynamic novel-view synthesis. We show outputs from CogNVS from two unseen videos; a real-world video above, and a generated video below.

Abstract

We explore novel-view synthesis for dynamic scenes from monocular videos. Prior approaches rely on costly test-time optimization of 4D representations or do not preserve scene geometry when trained in a feed-forward manner. Our approach is based on three key insights: (1) *covisible* pixels (that are visible in both the input and target views) can be rendered by first reconstructing the dynamic 3D scene and rendering the reconstruction from the novel-views and (2) *hidden* pixels in novel views can be “inpainted" with feed-forward 2D video diffusion models. Notably, our video inpainting diffusion model (CogNVS) can be self-supervised from 2D videos, allowing us to train it on a large corpus of in-the-wild videos. This in turn allows for (3) CogNVS to be applied zero-shot to novel test videos via *test-time finetuning*. We empirically verify that CogNVS outperforms almost all prior art for novel-view synthesis of dynamic scenes from monocular videos.

[*]Equal contribution.

1 Introduction

Rapid advances in static 3D scene representations [50, 31] have paved the way for spacetime understanding of the dynamic world. This has enabled photorealistic content creation and immersive virtual reality applications. In this work, we focus on the problem of novel-view synthesis from casually-captured monocular videos of dynamic scenes.

Why is this hard? Prior work on dynamic view synthesis addresses this task from two extremes. The first class of methods "test-time" optimize a new 4D representation from scratch for every new test video. While this ensures physically-plausible scene geometry, careful choices in modeling scene motion – in the form of an independent deformation field, or learnable temporal offsets – have to be made [74, 41, 81]. More importantly, it can take on the order of hours to optimize and render a novel-view video. An attractive alternative is to train large feed-forward video models *directly* for view synthesis [95, 25]. While inference on such models is dramatically faster (on the order of milliseconds), the resulting renderings often are not as accurate as their test-time optimized counterparts. From a pragmatic perspective, such models need to be trained on mega-scale multi-view training data, which is difficult to obtain for dynamic scenes.

Our method addresses the above challenges by decomposing the problem of dynamic view-synthesis into three distinct stages. First, we lean on the success of non-rigid structure from motion [46, 94, 40] approaches that produce reconstructions of visible scene regions, sometimes known as "2.5D" reconstructions (since occluded regions are not reconstructed). We point out that such reconstructions can be trivially produced for casual mobile videos captured with depth sensors and egomotion [20]. When such reconstructions are rendered from a target novel view, previously-hidden regions will not not be rendered. To "inpaint" these regions, we train a 2D video-inpainter – CogNVS – by fine-tuning a video diffusion model (CogVideoX [89]) to condition on the partially-observable novel-view pixels. Importantly, we allow CogNVS to *also* update the appearance of previously-visible pixels, allowing our pipeline to model view-dependent (dynamic) scene effects.

The **key insight** of our work is that CogNVS can be trained on any 2D video via self-supervision. However, rather than training our inpainter with random 2D masks, we make use of 3D multi-view supervision that better captures 3D scene visiblity, similar to prior art [78]. Specifically, given a 2D training video, we first reconstruct it (with an off-the-shelf method such as MegaSAM) and then render the reconstruction from a random camera trajectory. This rendering is used to identify co-visible pixels from the source video that remain visible in the novel views. This original source video and its co-visible-only masked variant can now form a training pair for 3D-consistent video inpainting. Importantly, because such a training pair does *not* require ground-truth 3D supervision, CogNVS can be trained on diverse in-the-wild 2D videos. We use dynamic scenes from TAO [13], SA-V [59], Youtube-VOS [86], and DAVIS [54]. Equally as important, we use the same paradigm to *test-time finetune* CogNVS on the test video-of-interest. We show that this allows our pipeline to "zero-shot" generalize to test videos that were never seen during training. We argue that our test-time finetuning of 2D diffusion models can be seen as the "best-of-both-worlds", by leveraging large-scale training data (for data-driven robustness) and test-time optimization (for accuracy).

In summary, our contributions are as follows: (1) We decompose dynamic view synthesis into three stages of reconstruction, inpainting and test-time finetuning, (2) we use a large corpus of *only 2D videos* for training CogNVS, and (3) we do extensive *zero-shot* benchmarking on three evaluation datasets against state-of-the-art methods and show improvements on dynamic view synthesis.

2 Related Work

Novel-view synthesis has seen recent advancements with the rise of implicit scene representations like NeRFs [50] and Gaussian primitives [31, 32, 12]. We have seen widespread efforts in scaling these representations to model larger scenes [72, 83, 68], making them faster to fit [36, 22, 15, 31, 51, 6, 14, 8], anti-aliased [42, 2, 3, 26, 4], and extend to representing dynamic scenes [64, 21, 56, 36]. The most popular paradigms have been the adoption of dynamic NeRFs [50, 56, 19, 52] and deformable Gaussian primitives [31, 81, 88, 47] for modeling scene dynamics, apart from using voxel grids [34, 33] or learnable tokenzation [96]. Most approaches need multi-view posed videos

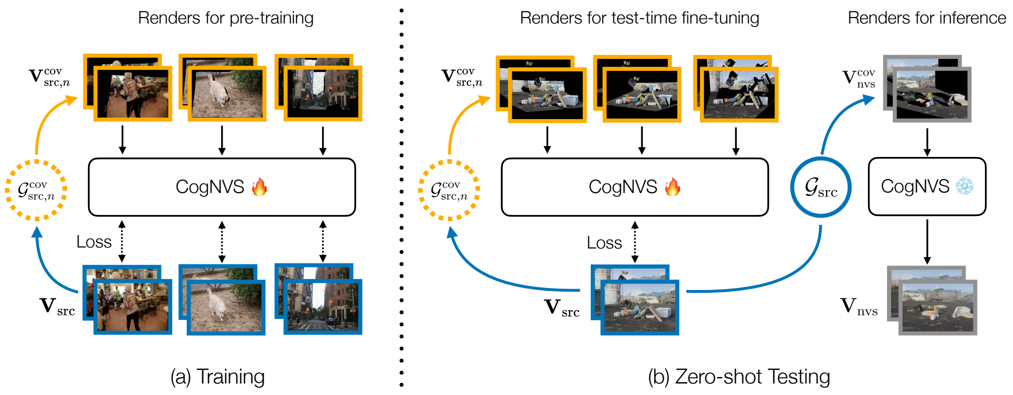
Renders for pre-training
Renders for test-time fine-tuning
Renders for inference
$\mathbf{V}^{\text{cov}}_{\text{src},n}$
$\mathcal{G}^{\text{cov}}_{\text{src},n}$
CogNVS
Loss
$\mathbf{V}_{\text{src}}$
$\mathcal{G}_{\text{src}}$
$\mathbf{V}^{\text{cov}}_{\text{nvs}}$
$\mathbf{V}_{\text{nvs}}$
(a) Training
(b) Zero-shot Testing

Figure 2: **CogNVS overview.** During training (**left**), given a 2D source video (in blue) of a dynamic scene, we first reconstruct the scene using off-the-shelf monocular reconstruction algorithms like MegaSAM [40] to obtain the 3D scene geometry, $\mathcal{G}_{\text{src}}$ and camera odometry, $\mathbf{c}_{\text{src}}$. We then sample a set of arbitrary camera trajectories $\{\mathbf{c}_1, \cdots, \mathbf{c}_N\}$ to simulate plausible occluded geometries, $\{\mathcal{G}^{\text{cov}}_{\text{src},1}, \cdots, \mathcal{G}^{\text{cov}}_{\text{src},N}\}$ which when rendered from original camera trajectory, $\mathbf{c}_{\text{src}}$ produces a mask of source pixels that are co-visible in the sampled trajectory (in orange). The source video and its masked variant produce a self-supervised training pair for learning CogNVS, our video inpainting diffusion model (visualized in Fig. 3). At inference (**right**), we finetune CogNVS on the given input sequence by similarly constructing self-supervised training pairs. The final novel-view is then generated using the finetuned CogNVS in a feed-forward manner.

as input, and only recently monocular view synthesis has gained traction [17, 37, 74, 39]. However, each of the aforementioned approaches have to be test-time optimized separately for every new test video, are slow to optimize and yet fail to recover highly-detailed dynamic scene content [37]. Moreover, there is no focus on predicting the unobservable scene content, which is exacerbated by benchmarking metrics that only evaluate co-visible pixels [20] in training and inference views and therefore encourage benchmarking on novel views that are not too far apart from the training views. Our approach instead reformulates dynamic view synthesis as an inpainting task, which specifically focuses on generating parts of the scene that were occluded from the training views, thereby facilitating extreme novel view synthesis for dynamic scenes. Our large-scale pretraining for feed-forward novel-view inpainting enables data-driven robustness.

Data-driven novel-view synthesis approaches have emerged [73, 82, 61, 92, 93, 45, 87] which train for view synthesis in a feed-forward manner with large-scale data. One class of methods is based on transformer architectures, more often than not trained with multi-view supervision and rendering in the loop [25, 95, 29, 80, 85, 60]. Another class of methods reformulate novel-view synthesis as a conditional generation task and use diffusion-based generative architectures [45, 62] for the same. Initially, the focus was on developing data-driven pipelines for static novel-view synthesis [45, 44, 43, 93, 95, 80, 7, 76], or exploiting data-driven priors [55, 75, 66, 9, 69, 35, 57, 90], using multi-view posed image inputs. However, the focus is now shifting to dynamic view synthesis of casually captured videos in an unconstrained setting [73, 61, 92, 87, 38, 28]. These approaches allow a greater level of hallucination of unseen scene components which instills the capability of view synthesis for camera poses that are far apart from the training views. We fit into this setting. While the data-driven learning provides faster inference times and a broad generalization, it compromises on 3D geometric accuracy and physical-plausibility which introduces unrealistic artifacts in the synthesized outputs (*e.g.*, objects suddenly exist or cease to exist) [92]. In this work, we highlight that test-time finetuning is crucial to preserving the 3D geometry of the scene and reducing implausible artifacts.

Test-time finetuning is a long-standing paradigm to curb distribution shifts in machine learning algorithms and improve their generalization. It's origin lies in early-age algorithms for optical character recognition [5] and text classification [30], where the algorithm adjusts itself *after* observing the test data. A decade back, it popularly resurfaced for super-resolution [63] where learning to super-resolve an image was achieved by downsampling and super-resolving the test image. Domain generalization approaches for vision [67, 11, 27, 10, 18, 97] soon took inspiration from this breakthrough and recently, chain-of-thought prompting [79] and general LLM reasoning [1] in natural language processing adapted this paradigm. The most recent adoption was seen in 4D reconstruction and tracking [16], and we similarly explore this paradigm further in our work.

3 Method

Given a monocular video of a dynamic scene, $\mathbf{V}_{\text{src}} = \{\mathbf{V}^t_{\text{src}}\}_{t=1}^T$, we want to generate a novel view of the observed scene, $\mathbf{V}_{\text{nvs}} = \{\mathbf{V}^t_{\text{nvs}}\}_{t=1}^T$ from a target camera pose. As discussed (c.f. Fig. 2), we achieve this by decomposing the task into three distinct stages – (1) obtain an off-the-shelf reconstruction of the observed scene over time, (2) render the scene from the novel views and inpaint the non-co-visible regions, and (3) curb the train-test distribution shift with test-time finetuning. Note that the first two stages of our pipeline are similar to concurrent work [92, 61] but importantly we find that test-time finetuning is a crucial stage to allow generalization. We now describe each of the stages in detail.

3.1 Dynamic view synthesis as structured inpainting

We use off-the-shelf SLAM frameworks, like MegaSAM [40], to obtain a reconstruction of the given scene. Formally, let the underlying 3D structure of the world as observed by $\mathbf{V}_{\text{src}}$ be represented by, $\mathcal{G}_{\text{src}} = \{\mathbf{X}^t_{\text{src}}\}_{t=1}^T$, where $\mathbf{X}^t_{\text{src}}$ are the evolving 3D primitives (points, Gaussians, etc.) across time, t. Any physical properties of the primitives are omitted from this discussion for simplicity. Let the recovered camera poses from which $\mathbf{V}_{\text{src}}$ was observed be, $\mathbf{c}_{\text{src}} = \{\mathbf{c}^t_{\text{src}}\}_{t=1}^T$, where $\mathbf{c}$ denotes a camera pose and is formulated as, $\mathbf{c} = (\mathbf{R}, \mathbf{t}) \in \text{SE}(3)$ lie group. The source video $\mathbf{V}_{\text{src}}$ can be obtained by using a rendering function $\mathcal{R}$ as,

$$\mathbf{V}_{\text{src}} = \mathcal{R}\big(\mathcal{G}_{\text{src}}, \mathbf{c}_{\text{src}}\big)$$

Learning to inpaint novel views For obtaining $\mathbf{V}_{\text{nvs}}$, we note that a subset of 3D primitives that must be visible from $\mathbf{c}_{\text{src}}$, are already available in the reconstructed scene geometry, $\mathcal{G}_{\text{src}}$. Therefore, a partial observation of the world in the form of *co-visible* pixels [20] from novel views, $\mathbf{c}_{\text{nvs}} = \{c^t_{\text{nvs}}\}_{t=1}^T$, can be rendered as follows,

$$\mathbf{V}^{\text{cov}}_{\text{nvs}} = \mathcal{R}\big(\mathcal{G}_{\text{src}}, \mathbf{c}_{\text{nvs}}\big)$$

At this point, the novel view synthesis is incomplete, and all missing regions have to be generated. To this end, we train a conditional video diffusion model, CogNVS (denoted by ϵ_θ) built on top of a recently proposed transformer-based video diffusion model [89]. CogNVS takes in the partially observed novel view video and generates an inpainted novel-view of the scene. The overall CogNVS pipeline first employs a 3D causal VAE to compress the conditioning $\mathbf{V}^{\text{cov}}_{\text{nvs}}$ and target novel-view $\mathbf{V}_{\text{src}}$ into latent representations $\mathbf{z}_{\text{cond}} = \mathcal{E}(\mathbf{V}^{\text{cov}}_{\text{nvs}})$ and $\mathbf{z}_0 = \mathcal{E}(\mathbf{V}_{\text{src}})$ respectively, enabling efficient training while preserving temporal coherence and photometric fidelity. Here, $\mathcal{E}$ is the VAE encoder. Gaussian noise is then added to the target latent $\mathbf{z}_0$, and the resulting noisy latent is concatenated with the conditional latent $\mathbf{z}_{\text{cond}}$. This joint representation is passed through a self-attention transformer equipped with 3D rotary positional embeddings (3D-RoPE) [65] and adaptive layer normalization, which predicts the added noise. The training objective follows a score matching formulation:

$$\min_\theta \; \mathbb{E}_{\substack{\mathbf{z}_0=\mathcal{E}(\mathbf{V}_{\text{src}}),\ \mathbf{z}_{\text{cond}}=\mathcal{E}(\mathbf{V}^{\text{cov}}_{\text{nvs}}),\\ k\sim\mathcal{U}\{1,\ldots,K\},\ \epsilon\sim\mathcal{N}(0,I)}} \Big\| \, \epsilon_\theta(\mathbf{z}_k, k, \mathbf{z}_{\text{cond}}) \; - \; \epsilon \Big\|_2^2$$

Here, $\mathbf{z}_k = \sqrt{\bar{\alpha_k}}\mathbf{z}_0 + \sqrt{1-\bar{\alpha_k}}\epsilon$ denotes the noisy latent at a uniformly sampled timestep k, where $\bar{\alpha_k}$ is the cumulative signal preserving factor. While CogVideoX was originally designed as an image-to-video diffusion model that zero-pads conditional image patches to match the length of the target video, we adapt it for a video-to-video setting, where the shapes of the conditional and target inputs are inherently aligned and no padding is needed. In practice, CogNVS is trained with datasets of 2D videos which are used to generate self-supervised training pairs. We discuss this below.

3.2 Data generation for self-supervised training

We propose to train CogNVS in a self-supervised manner. This allows us to use a large corpus of 2D videos. For each casually captured monocular video $\mathbf{V}_{\text{src}}$, similar to prior work [23, 48, 92], we

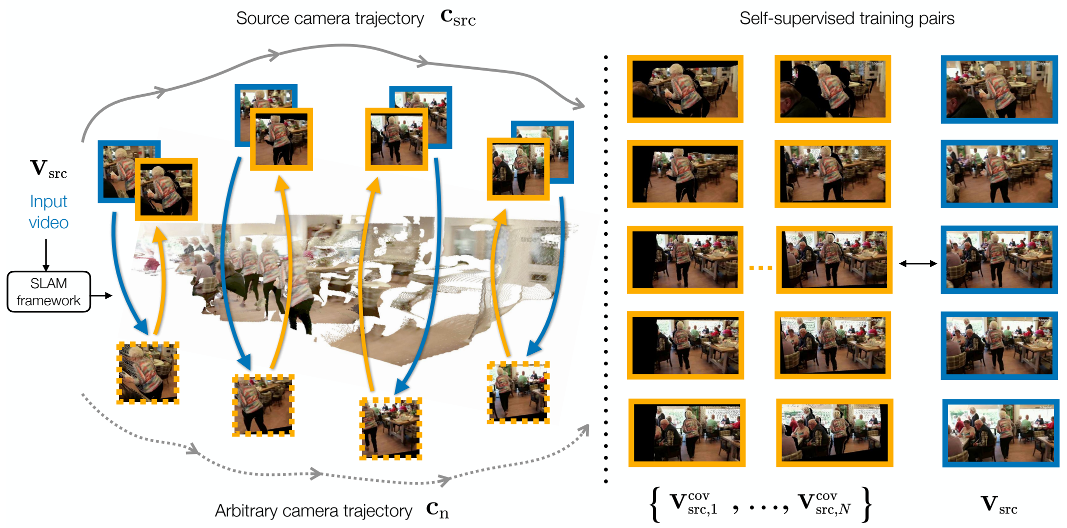
Source camera trajectory $\mathbf{c}_{\text{src}}$
Self-supervised training pairs
$\mathbf{V}_{\text{src}}$
Input video
SLAM framework
Arbitrary camera trajectory $\mathbf{c}_{\text{n}}$
$\{\mathbf{V}^{\text{cov}}_{\text{src},1}, \ldots, \mathbf{V}^{\text{cov}}_{\text{src},N}\}$
$\mathbf{V}_{\text{src}}$

Figure 3: **Self-supervised training data generation.** To curate a large training set for video inpainting, we first reconstruct an input source 2D video (in blue) with an off-the-shelf monocular SLAM system. After reconstruction, we randomly sample N pairs of 'start' and 'end' camera poses around a spherical region, $\mathcal{S}$ of the estimated camera pose in the given 2D video. $\mathcal{S}$ is bounded by a predefined deviation in the spherical coordinate axes, similar to a prior work [93]. We sample a $\text{SE}(3)$ camera trajectory that interpolates the start and end poses while looking at the center of the scene. We render the reconstruction from this novel trajectory (in dotted-orange), and use the rendering to identify co-visible pixels in the original source view (in orange). The source video and its masked variant are used to produce a self-supervised training pair for training CogNVS, our "3D-aware" video inpainting diffusion model.

obtain its 3D reconstruction $\mathcal{G}_{\text{src}}$ and odometry $\mathbf{c}_{\text{src}}$ from off-the-shelf SLAM frameworks [40]. As demonstrated in Fig. 3, we sample N arbitrary camera trajectories in order to create training pairs from 2D videos, described as follows.

We first obtain the "center" of the scene by considering the pixel at the optical center in the first frame of the given video, similar to a prior work [92]. We then construct a bounded region $\mathcal{S}$ in spherical coordinates, around the camera center of c^1_{src}. Within this region, we uniformly sample start and end spherical coordinates of each new camera trajectory, and then again sample two intermediate camera locations between the start and end spherical coordinates to ensure smoothness during interpolation. Camera poses are obtained by converting the spherical coordinates into euclidean space to get translations, and camera rotations are obtained such that the look-at vector always points to the center of the scene. Using the four sampled camera poses, we do bicubic interpolation on the $\text{SE}(3)$ manifold. This results in a set of smooth camera trajectories, $\{\mathbf{c}_{\text{n}}\}_{n=1}^N$ which are then used to construct the training pairs. With N trajectories, we can obtain "partial" novel-view renderings as,

$$\mathbf{V}^{\text{cov}}_n = \mathcal{R}\big(\mathcal{G}_{\text{src}}, \mathbf{c}_n\big)$$

Between $\mathbf{V}^{\text{cov}}_n$ and $\mathbf{V}_{\text{src}}$, only a subset of primitives from $\mathcal{G}_{\text{src}}$ are *co-visible*. Let this subset be denoted by $\mathcal{G}^{\text{cov}}_{\text{src},n}$ for the n^{th} trajectory. Then, partial renderings of the source video are given by,

$$\mathbf{V}^{\text{cov}}_{\text{src},n} = \mathcal{R}\big(\mathcal{G}^{\text{cov}}_{\text{src},n}, \mathbf{c}_{\text{src}}\big) \qquad \text{s.t.} \qquad \mathcal{D} = \{(\mathbf{V}^{\text{cov}}_{\text{src},n}, \mathbf{V}_{\text{src}})\}\forall n \in [1, N]$$

is the set of training pairs created by one monocular video. We repeat this for all 2D videos considered.

3.3 Test-time finetuning for target domain adaptation

At test time, to reduce domain gap arising due to different scene properties (lighting, appearance, motion) we use the source test video $\mathbf{V}_{\text{src}}$ to adjust the priors of CogNVS and create self-supervised finetuning pairs, $\mathcal{D}$ as described above. We therefore adapt the model weights θ on-the-fly with M

gradient steps with η step size as follows,

$$\theta \leftarrow \theta - \eta \nabla_\theta \big\| \epsilon_\theta(\mathbf{z}_k, k, \mathbf{z}^n_{\text{cond}}) - \epsilon \big\|_2^2,$$

where $\mathbf{z}^n_{\text{cond}}$ is the latent of the n^{th} self-supervised training pair input. At the end of finetuning, we obtain the desired novel view $\mathbf{V}_{\text{nvs}}$ from CogNVS by using the partially observed novel-view, $\mathcal{R}(\mathcal{G}_{\text{src}}, \mathbf{c}_{\text{nvs}})$ as the input conditioning, and running a reverse diffusion process.

4 Empirical Analysis

4.1 Experimental setup

Datasets We train CogNVS on four in-the-wild video datasets, SA-V [59], TAO [13], Youtube-VOS [86], and DAVIS [53]. We sample 3000, 3000, 4000 and 100 videos respectively from each of the datasets, giving us a total training video pool of $\approx$ 10,000 videos. For pretraining, we randomly select a new subsequence of 49-frames in every epoch and construct its training pairs. For benchmarking, we follow prior work [37, 73] and use a combination of Kubric-4D, ParallelDomain-4D [73] and Dycheck [20]. These have a held-out test set of 20, 20 and 5 videos each. Note that our evaluation on Kubric-4D, ParallelDomain-4D and Dycheck is zero-shot as the datasets are not seen during training. Since the Kubric-4D and ParallelDomain-4D are synthetic, we use their groundtruth point clouds and odometry for a fair comparison to baselines. For Dycheck, we use MegaSAM for reconstruction and align the estimated point cloud with the groundtruth to solve for scale ambiguity.

Baselines For Kubric-4D, we consider GCD [73] and Gen3C [61], alongside a concurrent work, TrajectoryCrafter [92]. For ParallelDomain-4D, we consider the same baselines except Gen3C, which only evaluates on Kubric-4D, as there is no open-source implementation available yet. For Dycheck, we consider recent work like Shape-of-Motion [74], MoSca [37], CAT4D [82]. Note that we do not benchmark test-time optimization approaches on Kubric-4D and ParallelDomain-4D, because their performance degrades catastrophically on novel views that are far apart from training views. For more quantitative analysis of CAT4D, see appendix.

Metrics For pixel-wise photometric evaluation, we adopt the widely used PSNR, SSIM, and LPIPS family of metrics for evaluating reconstruction quality via novel-view synthesis. We additionally benchmark the generation quality with FID and KID. This is in line with the benchmarking proposed in several diffusion-based view synthesis works [73, 61, 45, 71].

Implementation details During pretraining, we load the official CogVideoX-5B-I2V checkpoint and fully finetune all 42 transformer blocks. We use the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and $\beta_3 = 0.98$, a learning rate of $2 \times 10e - 5$, and a batch size of 8 for 12,000 steps. To fit within 48GB VRAM, we employ DeepSpeed ZeRO-2 [58] to partition model states across 8 A6000 Ada GPUs in a distributed setting. Pretraining completes in approximately 3 days.

During test-time finetuning, we maintain the same optimizer and learning rate but reduce the number of steps to 200 for shorter sequences (e.g., Kubric-4D) and 400 for longer ones (e.g., DyCheck). For all experiments, we use an input resolution of $\mathbb{R}^{49 \times 480 \times 720}$, set the classifier-free guidance scale to 6, and run 50 inference steps. A single novel-view sequence generates in $\sim$5 mins on an A6000 Ada. We provide additional implementation details and evaluation protocols in appendix.

4.2 Comparison to state-of-the-art

Kubric-4D and ParallelDomain-4D We first do zero-shot benchmarking of CogNVS on two synthetic datasets that come with high-fidelity dense depth and accurate camera odometry annotations. For a fair comparison to all baselines, we use the depth and poses to backproject the given scene into a canonical coordinate frame. Given this scene, we generate self-supervised pairs for test-time finetuning. Upon inference (see Tab. 1), we find that CogNVS beats prior work on photometric evaluation with PSNR, SSIM, LPIPS, even when baselines are not evaluated zero-shot (GCD is trained on Kubric-4D and ParallelDomain-4D and Gen3C is trained on Kubric-4D). In Fig. 4, we show the plausible and realistic novel-views predicted by our method on both datasets as compared to the baselines. This is quantitatively demonstrated by better FID and KID scores. A concurrent

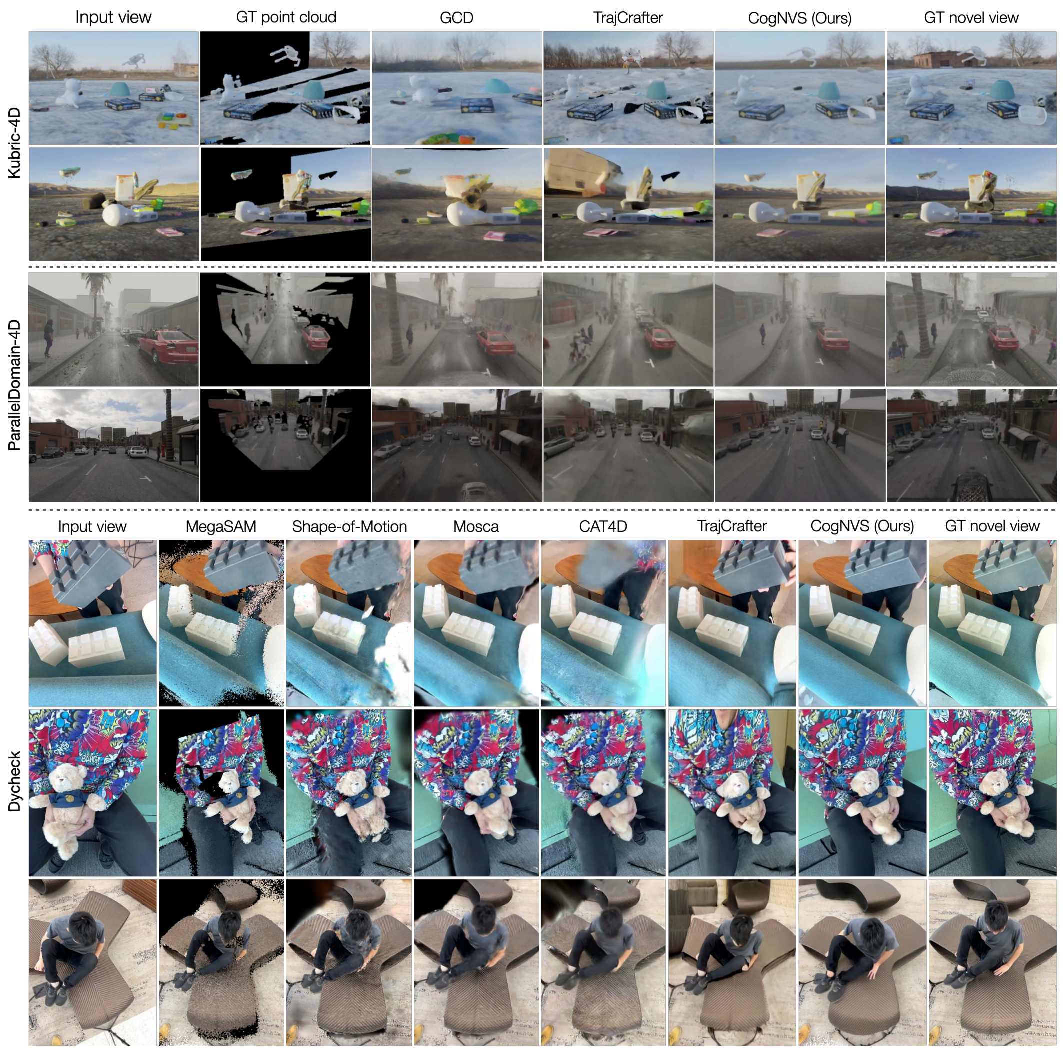

Figure 4: We show a qualitative comparison with state-of-the-art approaches for dynamic novel-view synthesis on Kubric-4D (**top**), ParallelDomain-4D (**middle**) and DyCheck (**bottom**). Note how reconstruction alone, either by groundtruth depth, MegaSAM [40], Shape of Motion [74], or MoSca [37] cannot synthesize a complete novel view. Optimization based approaches like Shape of Motion, and MoSca, blur the dynamic regions when fitting 4D representations. CAT4D [82], whose visuals are taken from its project page due to unavailable code, struggles to generalize. TrajectoryCrafter [92] over-hallucinates the occluded regions and does not preserve geometry. GCD [73] performs well because it was trained on Kubric-4D and ParallelDomain-4D. Our method can instead produce photorealistic and 3D-consistent novel-views for the given scenes in a *zero-shot* manner with test-time finetuning, even starting from point cloud renders that are incomplete and noisy (*e.g.*, from MegaSAM for DyCheck). It is consistently able to synthesize sharp dynamic objects, which the other baselines struggle with. Please see the video in the appendix.

work, TrajectoryCrafter [84] performs competitively. We also evaluate the rendered visible scene structure from groundtruth depth, for establishing a lower bound on dynamic-view synthesis.

DyCheck We evaluate the performance of our method on a real-world dataset of casually captured iPhone videos in Tab. 8. First, note that since CogNVS can be applied on top of reconstructions from any method, we show two variants. Better initial reconstruction (in this case, with MoSca rather than MegaSAM) allows for better dynamic view synthesis. Second, of all approaches, our method produces the most visually plausible novel views, as captured by drastically better FID and KID.

Table 1: Comparison to state-of-the-art for dynamic view synthesis on Kubric-4D and ParallelDomain-4D. We find that our method, that operates zero-shot unlike Gen3C and GCD, achieves state-of-the-art performance across a majority of metrics. † Note that Gen3C only evaluates on Kubric-4D and there is no open-source code that would allow us to benchmark it on ParallelDomain-4D.

Method	Kubric-4D					ParallelDomain-4D				
	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	KID ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	KID ↓
GT	15.12	0.671	0.328	175.01	0.063	18.79	0.499	0.409	197.99	0.129
GCD [73]	18.59	0.555	0.383	121.57	0.020	21.77	0.665	0.400	90.58	0.022
Gen3C† [61]	19.41	0.630	0.290	98.58	n/a	n/a	n/a	n/a	n/a	n/a
TrajCrafter [92]	20.93	0.730	0.257	130.20	0.024	21.46	0.719	0.342	95.38	0.026
CogNVS	22.63	0.760	0.232	102.47	0.008	24.34	0.797	0.302	102.43	0.033

Table 2: Comparison to state-of-the-art for dynamic novel-view synthesis on Dycheck. First, we note that our method can be run on top of any reconstruction approach and the better the reconstruction (*e.g.*, replacing MegaSAM with MoSca), the better the view synthesis. Second, we see that our method can achieve state-of-the-art FID / KID scores because test-time optimization approaches [74, 37, 40] result in blurry dynamic regions and cannot hallucinate new scene content, and completely feed-forward approaches [92] cannot return precise geometry. Our method instead gets the "best of both worlds".

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	KID ↓
MegaSAM [40]	12.16	0.299	0.698	239.57	0.148
Shape-of-Motion [74]	15.30	0.476	0.494	164.29	0.073
MoSca [37]	16.22	0.472	0.586	148.18	0.063
TrajCrafter [92]	12.74	0.337	0.749	140.35	0.059
CogNVS (MegaSAM)	15.19	0.382	0.622	94.48	0.030
CogNVS (MoSca)	16.94	0.449	0.598	92.83	0.031

Third, note how TrajectoryCrafter [92], also based on video diffusion which was the second-best method on Kubric-4D, is unable to handle the distribution shift in Dycheck (shallow field-of-view, close-up videos of moving objects) and fails to generalize. Whereas our method benefits from test-time finetuning and is able to adjust to any new data-distribution at test-time. Other test-time optimization approaches (Shape-of-Motion, MoSca) do better as long as evaluation views are close to training views, because there is only one distribution they need to fit to.

Runtime analysis We conduct a runtime analysis of CogNVS against other state-of-the-art baselines, as shown in Tab. 3, covering both stages of test-time optimization and rendering/inference. Specifically, we measure runtime by evaluating each method on the DyCheck evaluation set, with an average video length of 400 frames. Under our default hardware configuration, CogNVS requires 140 min for 400 steps of test-time fine-tuning on these long videos, followed by an additional 5 minutes for rendering. Note that this duration can be flexibly traded off with computational cost, which has recently been referred to as "inference-time scaling" in diffusion models [49]. For example, as shown in Fig. 8 (left) in the appendix, our method achieves 96% of its final performance with only half the fine-tuning steps.

4.3 Ablation studies

Effect of test-time finetuning We study the effectiveness of the test-time finetuning stage of our method. Row 2 vs. 3 in Tab. 4 show that proposed self-supervised finetuning is crucial for adaptation of CogNVS to a target video's distribution at test-time. Once the self-supervised test-time finetuning stage is completed, our method yields outputs with high fidelity, showcasing improved precision, and more contextually and geometrically consistent 3D appearances, as shown in Fig. 5.

Effect of large-scale pretraining We also study the usefulness of the large-scale pretraining stage with 2D videos from 4 training datasets. In this case, test-time finetuning alone with a self-supervised objective, cannot pull CogNVS out of the local minima it reaches without a good initialization. This is a common failure mode of many test-time optimization approaches that overfit to the training views

Table 3: Runtime analysis of test-time optimization and feed-forward approaches on DyCheck. We report the time taken for optimization / test-time finetuning for all approaches in addition to the final inference / rendering duration. In general, our optimization is faster than some test-time optimization approaches with the additional benefit of being able to inpaint/hallucinate unknown regions in a spatiotemporally consistent manner. Additionally, our inference is on par with other feed-forward methods.

Method	Optimization	Rendering / Inference
MegaSAM [40]	9 min	Real-time
MoSca [37]	66 min	Real-time
Shape of Motion [74]	237 min	Real-time
GCD [73]	-	2 min
TrajCrafter [92]	-	5 min
CogNVS	140 min	5 min

Table 4: We ablate our design choices of large-scale pretraining and test-time finetuning on three randomly chosen sequences from Kubric-4D test set. We find that no pretraining is detrimental to the performance of CogNVS, so much so that the PSNR drops by 5 points, thereby devoiding CogNVS of data-driven robustness. Test-time finetuning is also essential as without the adaptation of CogNVS to the test video, the performance in terms of PSNR drops by ~ 3 points.

Pretrain	Finetune	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	KID ↓
✗	✓	18.62	0.691	0.318	201.29	0.051
✓	✗	20.06	0.662	0.284	185.48	0.038
✓	✓	**23.29**	**0.779**	**0.240**	**158.98**	**0.036**

but default to rendering artifacts such as blurry dynamic regions [37]. We show in Tab. 4 (Row 1 vs. 3) and Fig. 5 that pretraining is essential for data-driven robustness.

Effect of reconstruction quality Although we touch upon how the initial reconstruction affects the quality of dynamic view synthesis, we describe in detail here. We create a pertubed version of the Kubric-4D dataset, by obtaining reconstruction and odometry from MegaSAM and aligning the reconstruction to groundtruth to solve for scale ambiguity. Quantitative results show a ~ 3 points drop on PSNR and a consistently worse performance on all metrics with sub-optimal reconstructions and cameras. This also addresses the gap in the photometric performance (with PSNR, SSIM, LPIPS) of MegaSAM-based CogNVS on DyCheck. For the quantitative and qualitative analysis of this ablation, please see the appendix.

5 Discussion

In this work, we focus on the problem of dynamic novel-view synthesis from monocular videos. Contrary to prior state-of-the-art that approaches this task from two extremes (either test-time optimization for every new video from scratch, or large-scale feed-forward novel view synthesis) – we propose a simple setup that is the “best-of-both-worlds". We reformulate dynamic view synthesis as an inpainting task and lean on the success of reconstruction algorithms like MegaSAM that can estimate the structure and geometry of in-the-wild videos. We first train a video inpainter, CogNVS, on pairs of co-visible novel-view pixels and target novel-views via self-supervision on only 2D videos. At test-time, we propose to finetune CogNVS, again via self-supervision, to adjust to the target video distribution. The proposed setup provides data-driven robustness with the large-scale pretraining of a video inpainting model, and enhances 3D accuracy of the predictions with test-time finetuning.

Limitations CogNVS does not currently take advantage of open-source 3D and 4D video datasets and trains on a relatively small set of 2D videos. While the zero-shot evaluation can achieve better photorealistic performance than prior state-of-the-art even with this unprivileged training data, the model and its geometric inpainting capabilities can be enhanced by adding more training data from all three – 2D, 3D and 4D data sources. Additionally, the performance of CogNVS is dependent

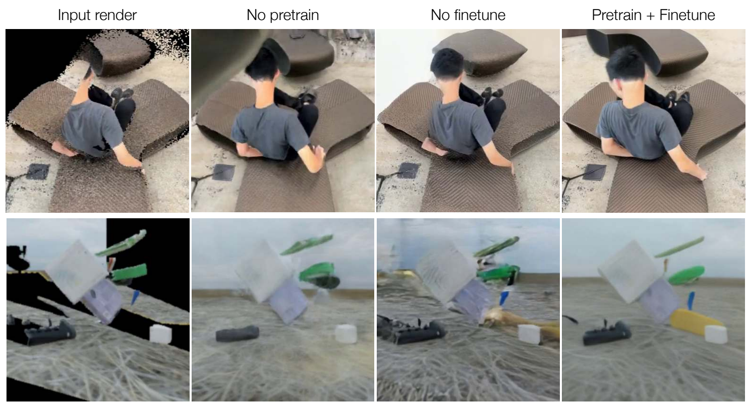

Figure 5: We qualitatively analyze the effect of pretraining and test-time finetuning. We note that without the data-driven robustness and generalization of pretraining (**second column**), CogNVS cannot hallucinate missing regions properly (*e.g.*, inpainted region in first row is still black in top left corner). Finally, without test-time finetuning (**third column**), 3D consistency and adherence to scene lighting and appearance properties cannot be ensured (*e.g.*, overall darker scene in second row, and output off by a few pixels at the bottom and right side of the image in first row, thereby inhibiting geometric consistency).

on the quality of dynamic scene reconstruction obtained from off-the-shelf structure from motion algorithms. When groundtruth structure and odometry is available, such as from ubiquitous depth sensors, CogNVS's performance can be increased. A limitation of the data generation pipeline is that the sampled arbitrary camera trajectories are not able to mimic the diversity of camera trajectories that are encountered in real-life, which is a bottleneck to the performance of CogNVS. A better strategy would be to create a "data-driven" trajectory sampler that samples from a set of real-world trajectories observed in the training set.

Acknowledgements: This work was supported in part by Bosch and by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number 140D0423C0074. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

References

[1] Ekin Akyürek, Mehul Damani, Adam Zweiger, Linlu Qiu, Han Guo, Jyothish Pari, Yoon Kim, and Jacob Andreas. "The Surprising Effectiveness of Test-Time Training for Few-Shot Learning". In: *arXiv preprint arXiv:2411.07279* (2024).

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields". In: *ICCV*. 2021.

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. "Mip-nerf 360: Unbounded anti-aliased neural radiance fields". In: *CVPR*. 2022.

[4] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. "Zip-nerf: Anti-aliased grid-based neural radiance fields". In: *ICCV*. 2023.

[5] Léon Bottou and Vladimir Vapnik. "Local learning algorithms". In: *Neural computation* 4.6 (1992), pp. 888–900.

[6] Ang Cao and Justin Johnson. "Hexplane: A fast representation for dynamic scenes". In: *CVPR*. 2023.

[7] Eric R Chan, Koki Nagano, Matthew A Chan, Alexander W Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. "Generative novel view synthesis with 3d-aware diffusion models". In: *ICCV*. 2023.

[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: *ICCV*. 2021.

[9] Kaihua Chen, Deva Ramanan, and Tarasha Khurana. "Using Diffusion Priors for Video Amodal Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2025.

[10] Liang Chen, Yong Zhang, Yibing Song, Ying Shan, and Lingqiao Liu. "Improved test-time adaptation for domain generalization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 24172–24182.

[11] Liang Chen, Yong Zhang, Yibing Song, Jue Wang, and Lingqiao Liu. "Ost: Improving generalization of deepfake detection via one-shot test-time training". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 24597–24610.

[12] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. "Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images". In: *ECCV*. 2024.

[13] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. "Tao: A large-scale benchmark for tracking any object". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer. 2020, pp. 436–454.

[14] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. "Depth-supervised nerf: Fewer views and faster training for free". In: *CVPR*. 2022.

[15] 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". In: *arXiv:2403.20309* (2024).

[16] Haiwen Feng, Junyi Zhang, Qianqian Wang, Yufei Ye, Pengcheng Yu, Michael J. Black, Trevor Darrell, and Angjoo Kanazawa. "St4RTrack: Simultaneous 4D Reconstruction and Tracking in the World". In: *arXiv preprint arxiv:2504.13152* (2025).

[17] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. "Dynamic neural radiance fields for monocular 4d facial avatar reconstruction". In: *CVPR*. 2021.

[18] Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei Efros. "Test-time training with masked autoencoders". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 29374–29385.

[19] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. "Dynamic View Synthesis from Dynamic Monocular Video". In: *ICCV*. 2021.

[20] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. "Monocular Dynamic View Synthesis: A Reality Check". In: *NeurIPS*. 2022.

[21] Quankai Gao, Qiangeng Xu, Zhe Cao, Ben Mildenhall, Wenchao Ma, Le Chen, Danhang Tang, and Ulrich Neumann. "Gaussianflow: Splatting gaussian dynamics for 4d content creation". In: *arXiv preprint arXiv:2403.12365* (2024).

[22] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. "Fastnerf: High-fidelity neural rendering at 200fps". In: *ICCV*. 2021.

[23] Yuxuan Han, Ruicheng Wang, and Jiaolong Yang. "Single-view view synthesis in the wild with learned adaptive multiplane images". In: *ACM SIGGRAPH*. 2022.

[24] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. "Text2Room: Extracting textured 3D meshes from 2D text-to-image models". In: *ICCV*. 2023, pp. 7909–7920.

[25] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. "LRM: Large Reconstruction Model for Single Image to 3D". In: *ICLR*. 2024.

[26] Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. "Tri-MipRF: Tri-Mip Representation for Efficient Anti-Aliasing Neural Radiance Fields". In: *ICCV*. 2023.

[27] Yusuke Iwasawa and Yutaka Matsuo. "Test-time classifier adjustment module for model-agnostic domain generalization". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 2427–2440.

[28] Hyeonho Jeong, Jinho Chang, Geon Yeong Park, and Jong Chul Ye. “Dreammotion: Space-time self-similarity score distillation for zero-shot video editing”. In: *arXiv e-prints* (2024), arXiv–2403.

[29] Hanwen Jiang, Qixing Huang, and Georgios Pavlakos. “Real3d: Scaling up large reconstruction models with real-world images”. In: *arXiv preprint arXiv:2406.08479* (2024).

[30] Thorsten Joachims et al. “Transductive inference for text classification using support vector machines”. In: *Icml*. Vol. 99. 1999, pp. 200–209.

[31] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. “3d gaussian splatting for real-time radiance field rendering”. In: *ACM TOG* (2023).

[32] Leonid Keselman and Martial Hebert. “Approximate differentiable rendering with algebraic surfaces”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 596–614.

[33] Tarasha Khurana, Peiyun Hu, Achal Dave, Jason Ziglar, David Held, and Deva Ramanan. “Differentiable raycasting for self-supervised occupancy forecasting”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 353–369.

[34] Tarasha Khurana, Peiyun Hu, David Held, and Deva Ramanan. “Point cloud forecasting as a proxy for 4d occupancy forecasting”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 1116–1124.

[35] Tarasha Khurana and Deva Ramanan. “Predicting Long-horizon Futures by Conditioning on Geometry and Time”. In: *arXiv preprint arXiv:2404.11554* (2024).

[36] Yao-Chih Lee, Zhoutong Zhang, Kevin Blackburn-Matzen, Simon Niklaus, Jianming Zhang, Jia-Bin Huang, and Feng Liu. “Fast View Synthesis of Casual Videos with Soup-of-Planes”. In: *ECCV*. 2025.

[37] Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. “MoSca: Dynamic Gaussian Fusion from Casual Videos via 4D Motion Scaffolds”. In: *arXiv preprint arXiv:2405.17421* (2024).

[38] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. “Spacetime gaussian feature splatting for real-time dynamic view synthesis”. In: *CVPR*. 2024.

[39] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. “Neural scene flow fields for space-time view synthesis of dynamic scenes”. In: *CVPR*. 2021.

[40] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. “Megasam: Accurate, fast, and robust structure and motion from casual dynamic videos”. In: *arXiv preprint arXiv:2412.04463* (2024).

[41] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. “DynIBaR: Neural Dynamic Image-Based Rendering”. In: *CVPR*. 2023.

[42] Zhihao Liang, Qi Zhang, Wenbo Hu, Lei Zhu, Ying Feng, and Kui Jia. “Analytic-splatting: Anti-aliased 3d gaussian splatting via analytic integration”. In: *ECCV*. 2024.

[43] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. “One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, pp. 10072–10083.

[44] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. “One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 22226–22246.

[45] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. “Zero-1-to-3: Zero-shot one image to 3d object”. In: *ICCV*. 2023.

[46] Jiahao Lu, Tianyu Huang, Peng Li, Zhiyang Dou, Cheng Lin, Zhiming Cui, Zhen Dong, Sai-Kit Yeung, Wenping Wang, and Yuan Liu. “Align3R: Aligned Monocular Depth Estimation for Dynamic Videos”. In: *arXiv preprint arXiv:2412.03079* (2024).

[47] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. “Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis”. In: *2024 International Conference on 3D Vision (3DV)*. IEEE. 2024, pp. 800–809.

[48] Zhen Lv, Yangqi Long, Congzhentao Huang, Cao Li, Chengfei Lv, Hao Ren, and Dian Zheng. “SpatialDreamer: Self-supervised Stereo Video Synthesis from Monocular Input”. In: *CVPR*. 2025, pp. 811–821.

[49] Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. "Inference-Time Scaling for Diffusion Models beyond Scaling Denoising Steps". In: *arXiv preprint arXiv:2501.09732* (2025).

[50] 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". In: *ECCV*. 2020.

[51] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. "Instant neural graphics primitives with a multiresolution hash encoding". In: *ACM Transactions on Graphics (ToG)* 41.4 (2022), pp. 1–15.

[52] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. "Nerfies: Deformable Neural Radiance Fields". In: *ICCV*. 2021.

[53] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. "A benchmark dataset and evaluation methodology for video object segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 724–732.

[54] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. "The 2017 davis challenge on video object segmentation". In: *arXiv preprint arXiv:1704.00675* (2017).

[55] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. "DreamFusion: Text-to-3D using 2D Diffusion". In: *ICLR*. 2023.

[56] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. "D-nerf: Neural radiance fields for dynamic scenes". In: *CVPR*. 2021.

[57] Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Ben Mildenhall, Nataniel Ruiz, Shiran Zada, Kfir Aberman, Michael Rubenstein, Jonathan Barron, Yuanzhen Li, and Varun Jampani. "DreamBooth3D: Subject-Driven Text-to-3D Generation". In: *ICCV*. 2023.

[58] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. "Zero: Memory Optimizations Toward Training Trillion Parameter Models". In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*. IEEE, 2020, pp. 1–16.

[59] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. "Sam 2: Segment anything in images and videos". In: *arXiv preprint arXiv:2408.00714* (2024).

[60] Jiawei Ren, Cheng Xie, Ashkan Mirzaei, Karsten Kreis, Ziwei Liu, Antonio Torralba, Sanja Fidler, Seung Wook Kim, Huan Ling, et al. "L4gm: Large 4d gaussian reconstruction model". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 56828–56858.

[61] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas MÃ¼ller, Alexander Keller, Sanja Fidler, and Jun Gao. "GEN3C: 3D-Informed World-Consistent Video Generation with Precise Camera Control". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2025.

[62] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. "Zero123++: a single image to consistent multi-view diffusion base model". In: *arXiv preprint arXiv:2310.15110* (2023).

[63] Assaf Shocher, Nadav Cohen, and Michal Irani. ""zero-shot" super-resolution using deep internal learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 3118–3126.

[64] Colton Stearns, Adam Harley, Mikaela Uy, Florian Dubost, Federico Tombari, Gordon Wetzstein, and Leonidas Guibas. "Dynamic gaussian marbles for novel view synthesis of casual monocular videos". In: *SIGGRAPH Asia 2024 Conference Papers*. 2024.

[65] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. "Roformer: Enhanced Transformer with Rotary Position Embedding". In: *Neurocomputing* 568 (2024), p. 127063.

[66] Jingxiang Sun, Bo Zhang, Ruizhi Shao, Lizhen Wang, Wen Liu, Zhenda Xie, and Yebin Liu. "Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior". In: *ICLR*. 2024.

[67] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. "Test-time training with self-supervision for generalization under distribution shifts". In: *International conference on machine learning*. PMLR. 2020, pp. 9229–9248.

[68] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. "Block-nerf: Scalable large scene neural view synthesis". In: *CVPR*. 2022.

[69] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. "Make-It-3D: High-fidelity 3D Creation from A Single Image with Diffusion Prior". In: *ICCV*. 2023.

[70] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. "Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training". In: *Advances in neural information processing systems* 35 (2022), pp. 10078–10093.

[71] Joseph Tung, Gene Chou, Ruojin Cai, Guandao Yang, Kai Zhang, Gordon Wetzstein, Bharath Hariharan, and Noah Snavely. "Megascenes: Scene-level view synthesis at scale". In: *European Conference on Computer Vision*. Springer. 2024, pp. 197–214.

[72] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. "Mega-NeRF: Scalable Construction of Large-Scale NeRFs for Virtual Fly-Throughs". In: *CVPR*. 2022.

[73] Basile Van Hoorick, Rundi Wu, Ege Ozguroglu, Kyle Sargent, Ruoshi Liu, Pavel Tokmakov, Achal Dave, Changxi Zheng, and Carl Vondrick. "Generative camera dolly: Extreme monocular dynamic novel view synthesis". In: *ECCV*. 2024.

[74] Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. "Shape of motion: 4d reconstruction from a single video". In: *arXiv preprint arXiv:2407.13764* (2024).

[75] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. "Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 8406–8441.

[76] Daniel Watson, William Chan, Ricardo Martin Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. "Novel View Synthesis with Diffusion Models". In: *ICLR*. 2023.

[77] Daniel Watson, Saurabh Saxena, Lala Li, Andrea Tagliasacchi, and David J. Fleet. "Controlling space and time with diffusion models". In: *ICLR*. 2025.

[78] Ethan Weber, Aleksander Holynski, Varun Jampani, Saurabh Saxena, Noah Snavely, Abhishek Kar, and Angjoo Kanazawa. "Nerfiller: Completing scenes via generative 3d inpainting". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 20731–20741.

[79] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. "Chain-of-thought prompting elicits reasoning in large language models". In: *Advances in neural information processing systems* 35 (2022), pp. 24824–24837.

[80] Xinyue Wei, Kai Zhang, Sai Bi, Hao Tan, Fujun Luan, Valentin Deschaintre, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. "MeshLRM: Large Reconstruction Model for High-Quality Meshes". In: *arXiv preprint arXiv:2404.12385* (2024).

[81] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. "4d gaussian splatting for real-time dynamic scene rendering". In: *CVPR*. 2024.

[82] Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T Barron, and Aleksander Holynski. "Cat4d: Create anything in 4d with multi-view video diffusion models". In: *arXiv preprint arXiv:2411.18613* (2024).

[83] Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. "BungeeNeRF: Progressive Neural Radiance Field for Extreme Multi-scale Scene Rendering". In: *ECCV*. 2022.

[84] Zeqi Xiao, Wenqi Ouyang, Yifan Zhou, Shuai Yang, Lei Yang, Jianlou Si, and Xingang Pan. "Trajectory Attention for Fine-grained Video Motion Control". In: *ICLR*. 2025.

[85] Desai Xie, Sai Bi, Zhixin Shu, Kai Zhang, Zexiang Xu, Yi Zhou, Sören Pirk, Arie Kaufman, Xin Sun, and Hao Tan. "Lrm-zero: Training large reconstruction models with synthesized data". In: *arXiv preprint arXiv:2406.09371* (2024).

[86] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. “Youtube-vos: Sequence-to-sequence video object segmentation”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 585–601.

[87] Jiawei Yang, Jiahui Huang, Yuxiao Chen, Yan Wang, Boyi Li, Yurong You, Apoorva Sharma, Maximilian Igl, Peter Karkus, Danfei Xu, et al. “STORM: Spatio-Temporal Reconstruction Model for Large-Scale Outdoor Scenes”. In: *arXiv preprint arXiv:2501.00602* (2024).

[88] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. “Real-time Photorealistic Dynamic Scene Representation and Rendering with 4D Gaussian Splatting”. In: *ICLR*. 2024.

[89] 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”. In: *arXiv preprint arXiv:2408.06072* (2024).

[90] Meng You, Zhiyu Zhu, Hui Liu, and Junhui Hou. “NVS-Solver: Video Diffusion Model as Zero-Shot Novel View Synthesizer”. In: *ICLR*. 2025.

[91] Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T. Freeman, Forrester Cole, et al. “WonderJourney: Going from anywhere to everywhere”. In: *CVPR*. 2024, pp. 6658–6667.

[92] Mark YU, Wenbo Hu, Jinbo Xing, and Ying Shan. “TrajectoryCrafter: Redirecting Camera Trajectory for Monocular Videos via Diffusion Models”. In: *arXiv preprint arXiv:2503.05638* (2025).

[93] 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”. In: *arXiv preprint arXiv:2409.02048* (2024).

[94] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. “Monst3r: A simple approach for estimating geometry in the presence of motion”. In: *arXiv preprint arXiv:2410.03825* (2024).

[95] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. “Gs-lrm: Large reconstruction model for 3d gaussian splatting”. In: *European Conference on Computer Vision*. Springer. 2024, pp. 1–19.

[96] Lunjun Zhang, Yuwen Xiong, Ze Yang, Sergio Casas, Rui Hu, and Raquel Urtasun. “Copilot4d: Learning unsupervised world models for autonomous driving via discrete diffusion”. In: *arXiv preprint arXiv:2311.01017* (2023).

[97] Yizhou Zhao, Hengwei Bian, Kaihua Chen, Pengliang Ji, Liao Qu, Shao-yu Lin, Weichen Yu, et al. “Metric from Human: Zero-shot Monocular Metric Depth Estimation via Test-time Adaptation”. In: *NeurIPS*. 2024.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: All claims are verified by the experiments in Sec. 4.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Limitations is in Sec. 5 of the main paper.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

Justification: We don't have any theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Most details are in the main paper in Sec. 3 and 4, some have been included in the supplement.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We include a zip file of the code in supplement with some starter instructions. More in-depth details about running all code will be publicly released on GitHub after acceptance.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: All details are provided exhaustively in Sec. 4.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [No]

Justification: Computing error bars is computationally expensive for video diffusion models which is what we are building upon.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: See Sec. 4.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: It does.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [NA]

    Justification: This does not apply to our work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: Our work only generates novel views of the provided input video.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Citations are in the main paper, licenses are included in the supplement.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: Our asset is the code and model, which we have already said is included in supplement with documentation.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: We do not do any crowdsourcing or research with human subjects.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: We did not work with any human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [NA]

    Justification: We do not use an LLM for this purpose.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix

In this appendix, we extend our discussion of dynamic view synthesis in casual monocular videos. First, we discuss the intricacies in the training and evaluation protocols adopted. This is followed by an in-depth ablation study on various design decisions in the proposed pipeline. Finally, we show more qualitative results, both on the considered benchmarks and on in-the-wild examples.

A Implementation Details

Training pair details To generate self-supervised training pairs, we randomly perturb the source camera trajectory to create diverse camera paths. In the spherical coordinate system, we sample random elevations from [-15°, 15°], azimuths from [-30°, 30°], and radius deviations from [-0.15, 0.15], followed by bicubic interpolation. This procedure enables flexible generation of training pairs across arbitrary camera trajectories. For pretraining, we use $N = 2$ camera views per training videos. For test-time finetuning, we set $N = 5$ for DyCheck and $N = 9$ for both Kubric-4D and ParallelDomain-4D, due to their wider novel-view gaps. When a video sequence exceeds CogVideoX's default input length of 49 frames, we randomly sample a 49-frame subsequence in each epoch. On DyCheck, we additionally apply a noise injection strategy to simulate real-world degradation on training pairs, as analyzed in Section B.

Evaluation protocol For Kubric-4D and ParallelDomain-4D, we follow the official GCD evaluation protocol, using the 20-sequence sub–test set and an input resolution of 576 (width) × 384 (height). On DyCheck, we follow the official Shape-of-Motion and Mosca evaluation protocols, and we consistently report evaluation metrics at an image resolution of 360 (width) × 480 (height) on the five standard test sequences: apple, block, paper-windmill, spin, and teddy. We render dynamic Gaussian representations from Mosca and Shape-of-Motion with a black background for fair comparison. Both methods optimize camera poses using the ground-truth novel views to improve photometric metrics; we retain this step to stay consistent with their original implementation. CAT4D, although diffusion-based, fits a 4D-GS representation (with minor extensions) after synthesizing multi-view videos. When evaluating CogNVS on MegaSAM renders, we append a static background extracted from the full input video to better capture long-term context. The effectiveness of background stacking is validated in Section B. Also note that Shape-of-Motion and MoSca optimize for evaluation camera poses during evaluation using ground-truth novel view videos. Whether CAT4D adopts this step is unknown. We do not do this camera pose optimization at test-time. Since the DyCheck evaluation sequences are more than 49-frames in length, we isolate the static scene regions and stack them in 3D across time. This accumulated background is then rendered onto each frame which helps, to a large extent, "pre-inpaint" the static background regions using fused information from multiple 49-frame length sequences.

Table 5: Effect of reconstruction quality on Kubric-4D. We quantitatively evaluate CogNVS's performance with the use of two different reconstructions for Kubric-4D. Groundtruth depth gives an upperbound on view synthesis performance by CogNVS. Our first observation, perhaps unsurprisingly, is that the quality of MegaSAM reconstruction is subpar to that of the groundtruth. This difference is quality is also translated to the novel-view synthesis task with CogNVS, where CogNVS used with groundtruth depth does 3 and 45 points better at PSNR and FID respectively as compared to CogNVS used on top of MegaSAM.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	KID ↓
MegaSAM	12.73	0.299	0.644	280.62	0.164
GT	15.12	0.671	0.328	175.01	0.063
CogNVS (MegaSAM)	19.62	0.621	0.313	147.83	0.033
CogNVS (GT)	**22.63**	**0.760**	**0.232**	**102.47**	**0.008**

B Ablation Study

Effect of reconstruction quality In Tab. 5, we show the effect of using different sources of reconstructions on the entire Kubric-4D evaluation set. Specifically, we compare the structure and

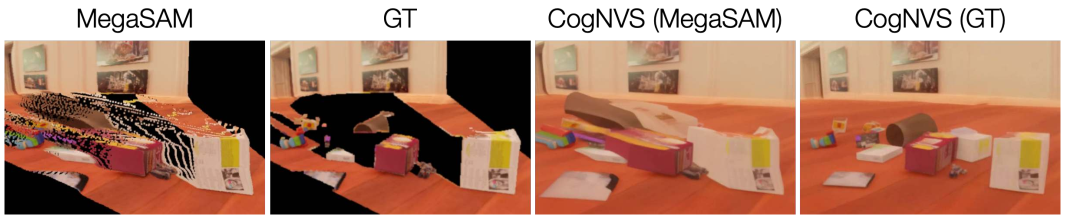

Figure 6: We show the effect of using different qualities of reconstruction. Note that the groundtruth depth of the scene is perfect because it is derived synthetically. This re-rendered depth results in more realistic object placements in the scene as compared to the predictions using the depth from MegaSAM. This is because the MegaSAM depth is noisy at the object edges and therefore results in smeared objects in the novel view predictions.

Table 6: Effect of masking strategy on Kubric-4D. We study the effect of building CogNVS as an inpainting model using other masking strategies, specifically, random and tube masking [70]. We find that random masking is the least optimal as it does not mimic the test-time scenario, tube masking does better, but our structured masking strategy is the best for the proposed structured inpainting task.

Mask	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	KID ↓
Random	20.62	0.755	0.310	187.79	0.059
Tube	21.75	0.778	**0.236**	173.55	0.041
Ours	**23.29**	**0.779**	0.240	**158.98**	**0.036**

odometry from MegaSAM [40] and the synthetic depth groundtruth from Kubric-4D [73]. We find that both quantitatively and qualitatively (c.f. Fig. 6), our pipeline benefits more from better reconstructions. This is because the quality of reconstruction directly affects the input to CogNVS, and if the input point cloud is noisy (*e.g.*, smearing at the object borders), the prediction of the novel view also becomes inaccurate.

Ablation on masking strategy Since CogNVS is an inpainting model, we ablate different masking strategies to train CogNVS on three sequences from Kubric-4D, instead of the proposed structured masking. In Tab. 6 and Fig. 7, we use random and tube masking from VideoMAE [70] and apply them with a 50% masking ratio on the input video sequences divided into 16×16 patches. We find that random masking is the least optimal as it does not resemble the structured inpainting task at test-time. Tube masking is more amenable to the test-time inpainting pattern, which reflects as better photometric and generative metrics. Of all, our structured masking obtained by rendering scene reconstructions into the novel views performs the best.

Ablation on test-time finetuning epochs Following the same data setup as above, we assess how the length of test-time finetuning affects the final prediction from CogNVS. In Fig. 8 (left), as expected we observe that the performance improvement in the first few epochs is high (both in terms of PSNR and FID going from 0 to 50 epochs) and saturates as the number of epochs are increased further (up to 200).

Top-K evaluation Following the same data setup as above, we compute probabilistic PSNR and FID metrics for CogNVS's performance on Kubric-4D in the form of Top-k metrics (where the best of k number is reported) in Fig. 8 (right). As we sample multiple modes from CogNVS's learnt distribution, the Top-k metrics for PSNR and FID become better and start to saturate near k=8.

Ablation on background stacking and noise addition We conduct an ablation on the MegaSAM reconstructions of DyCheck for the effect of static background stacking described in the previous section. In Tab. 7 and Fig. 9, we see that stacking the background on DyCheck provides a large in photometric performance. Secondly, we propose to add noise to dynamic object depths during training, especially for out-of-distribution data. This is essential as our creation of self-supervised training pairs only masks out certain pixels from the source video which leaves no room for CogNVS to be able to see real-world noise. To simulate real noise, at say object edges, we estimate the

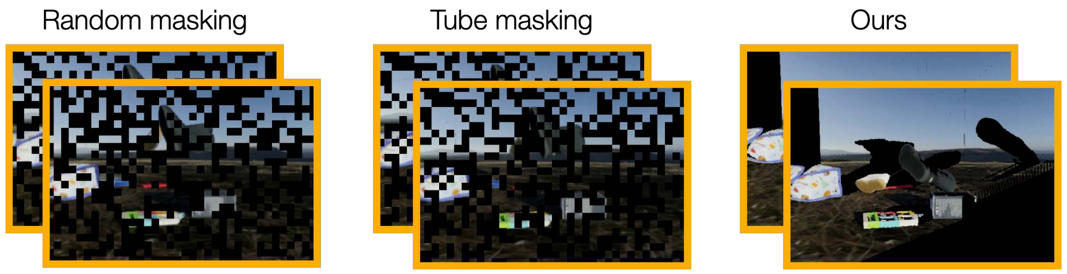

Figure 7: We illustrate the different masking strategies considered, as proposed by a prior work [70]. For random masking (**left**), the masked out patches are different in each frame of the input video. For tube masking (**center**), a random set of patches is masked but this set is constant across multiple frames of the video. For our structured masking (**right**), we derive the mask by rendering visible scene reconstruction from the novel views.

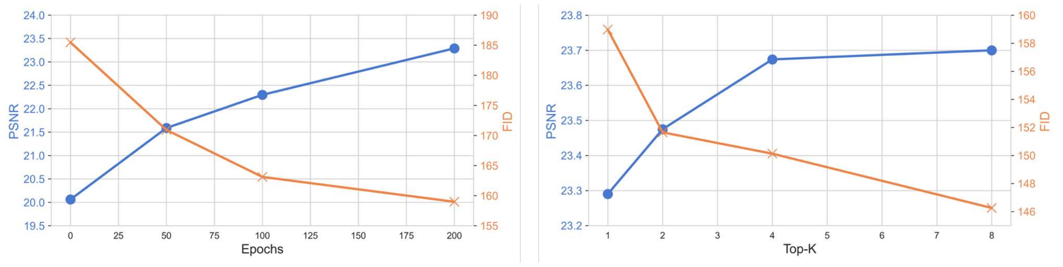

Figure 8: We conduct ablations on the number of epochs used for test-time finetuning (**left**) and number of samples drawn from CogNVS for a probabilistic evaluation (right). Both experiments suggest similar trends; performance improves with an increase in the number of finetuning epochs and increase in the number of samples drawn from our diffusion model. Performance saturates once a threshold is reached.

noise between (pseudo) groundtruth depth (coming from iPhone LiDARs or a state-of-the-art depth estimator, say, MoGe) and the predicted depth (coming from a SLAM framework like MegaSAM). This estimated noise for the source pixels, is added to the visible scene reconstruction but in the ray direction of the pixels visible in the arbitrary cameras. This results in noisy visuals that make CogNVS training more robust, especially to out-of-distribution cases. In Tab. 7 and Fig. 9, we demonstrate the improvements in performance by training CogNVS to inpaint in the presence of distracting noise artifacts.

Table 7: We quantitatively evaluate the effect of static background stacking and noise addition on DyCheck. Note that background stacking helps DyCheck because the video sequences are longer than 49-frames that CogNVS can handle. This gives us 3 points performance boost in PSNR. Adding real-world noise to dynamic objects helps make CogNVS robust to noise and therefore it reduces artifacts like smeared object edges, reflected in a much lower FID metric.

Background stacking	Noise addition	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	KID ↓
✗	✗	11.55	0.304	0.848	197.93	0.204
✓	✗	14.27	0.352	**0.737**	180.67	0.171
✓	✓	**14.30**	**0.354**	0.740	**156.83**	**0.141**

Evaluation with masked metrics on DyCheck In addition to the metrics reported in the main paper, we also report masked photometric errors as proposed by a prior work [20]. While this metric only evaluates the visible scene content and how any view-dependent changes were handled during novel-view synthesis, it does not encourage the generation of unseen scene regions. On this metric, CogNVS performs competitively as compared to baselines.

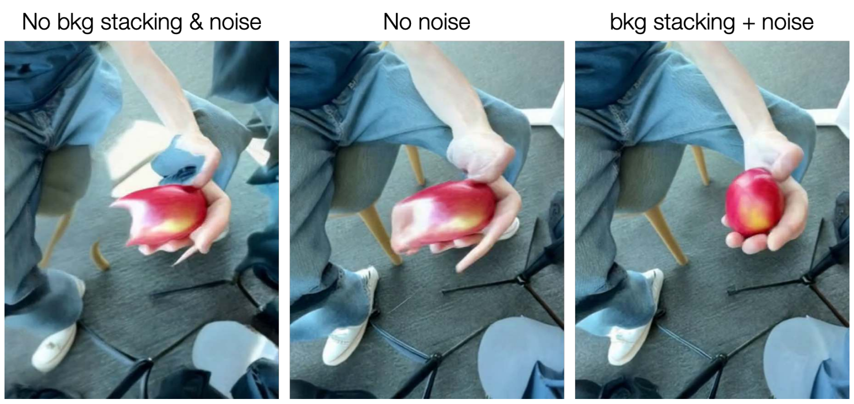

Figure 9: We visualize the 'apple' evaluation sequence from DyCheck for analysis of the effect of background stacking over time and noise addition strategy during training to simulate realistic in-the-wild scenarios. First (column 1 vs. 2), we see that for longer videos, stacking the static background from the entire input video helps accumulate multi-view cues about the static background. Second (column 2 vs. 3), we see that due to the noise addition strategy during training, CogNVS is more robust to real-world noise patterns like smearing across object (in this case, apple) edges.

Table 8: We report masked perceptual quality metrics as proposed by prior work [20]. This metric only evaluates the visible regions of the scene and so does not encourage generation of unseen scene components. Note that our method performs competitively as compared to the baselines which only focus on modeling the visible scene content.

Method	mPSNR ↑	mSSIM ↑	mLPIPS ↓
MegaSAM [40]	14.60	0.517	0.609
Shape-of-Motion [74]	16.47	0.639	0.409
MoSca [37]	17.82	0.635	0.507
CAT4D [82]	17.39	0.607	0.341
TrajCrafter [92]	13.60	0.518	0.663
CogNVS (MegaSAM)	15.35	0.549	0.557
CogNVS (MoSca)	17.33	0.607	0.530

Maximum novel view synthesis angle As demonstrated in Kubric 4D evaluation and several in-the-wild examples, CogNVS can plausibly generate novel views with up to 90 degrees of variation (up, down, left, or right) in general with a single forward pass. Moreover, our model can produce even more extreme viewpoints by progressively generating new views conditioned on past generations – a fairly straightforward and common strategy adopted by many generative models [24, 91, 77]. We illustrate such a progressively generated panoramic view in Fig. 10. Additionally, we want to point out that our method is not able to generate table top views of object-centric scenes (inward facing 360 degree view) beyond 90 degrees of camera deviation, likely because such training data was not seen by the model during pretraining.

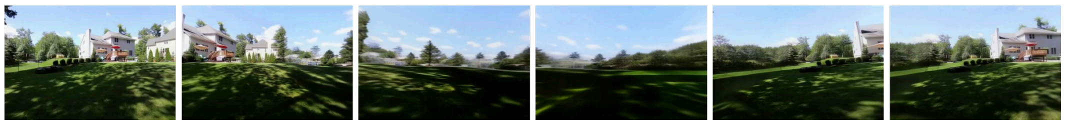

Figure 10: We create a panoramic view of the outdoor scene through progressive generation. Specifically, we divide the 360° camera trajectory into four 90° segments of 49 frames each, reconstructing the previously generated views before reprojecting and inpainting the next 90° novel view. This qualitative analysis suggests that CogNVS can still generate 3D plausible and temporally coherent novel-views even in such extreme cases.

C Qualitative comparison on evaluation datasets

Please see our webpage for videos in addition to the qualitative visuals below.

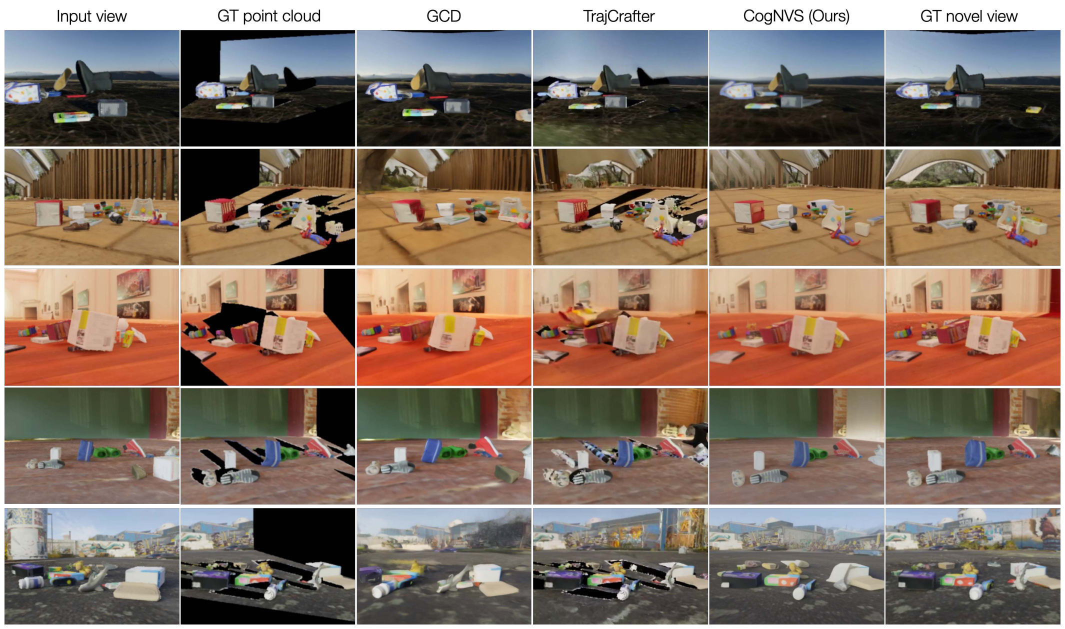

Figure 11: We show supplementary qualitative comparison on Kubric-4D. Note that TrajectoryCrafter is able to generate a reasonable background for the unseen scene regions, but is not able to inpaint the shadows / masks created by foreground objects. GCD is trained on Kubric-4D so performs reasonably well but struggles to preserve the precise geometry. CogNVS achieves better performance as compared to baselines and is the closest is geometric consistency to the groundtruth novel view.

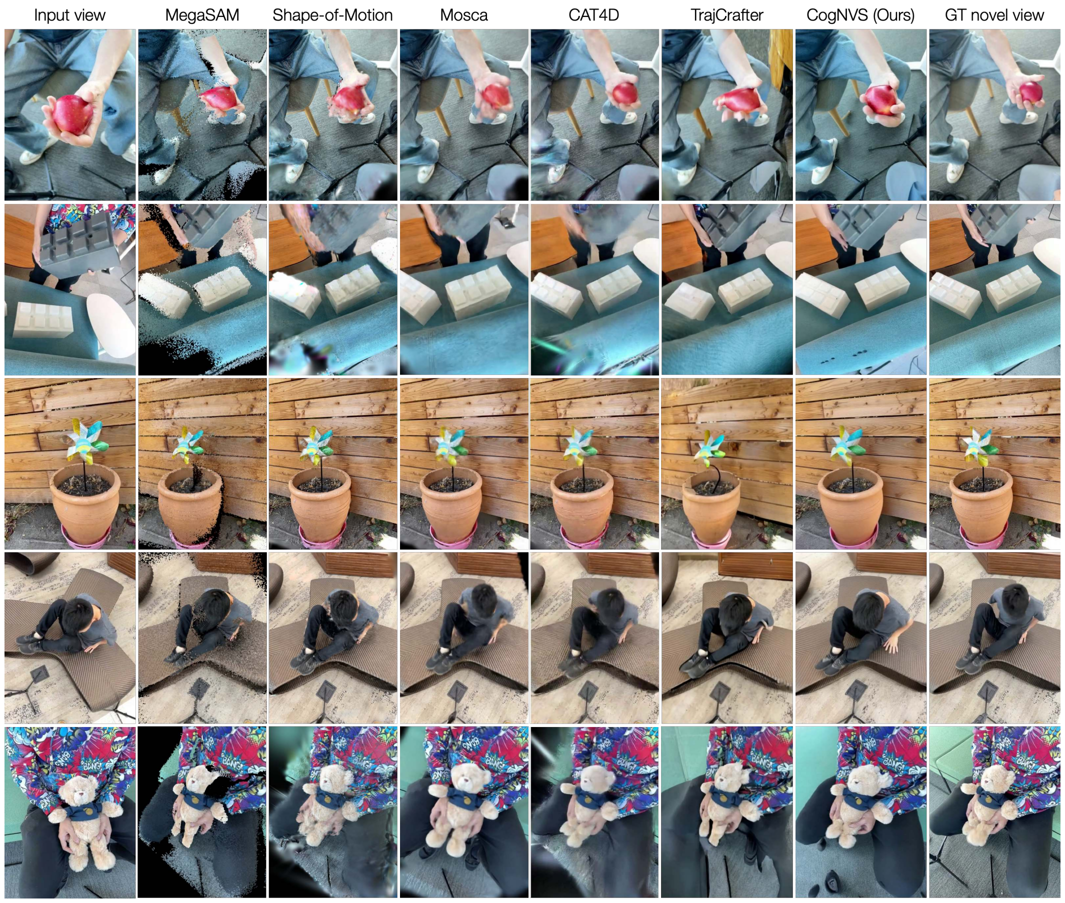

Figure 12: We show supplementary qualitative comparison on DyCheck with CogNVS which surpasses the performance of all prior state-of-the-art. Note that baselines either do not hallucinate the unseen regions in the novel-view (Shape-of-Motion, MegaSAM), show blurry dynamic regions (MoSca, CAT4D), or are not able to preserve the underlying geometry of the scene (TrajectoryCrafter).

D Qualitative results on in-the-wild examples

Please see our webpage for videos in addition to the qualitative visuals below.

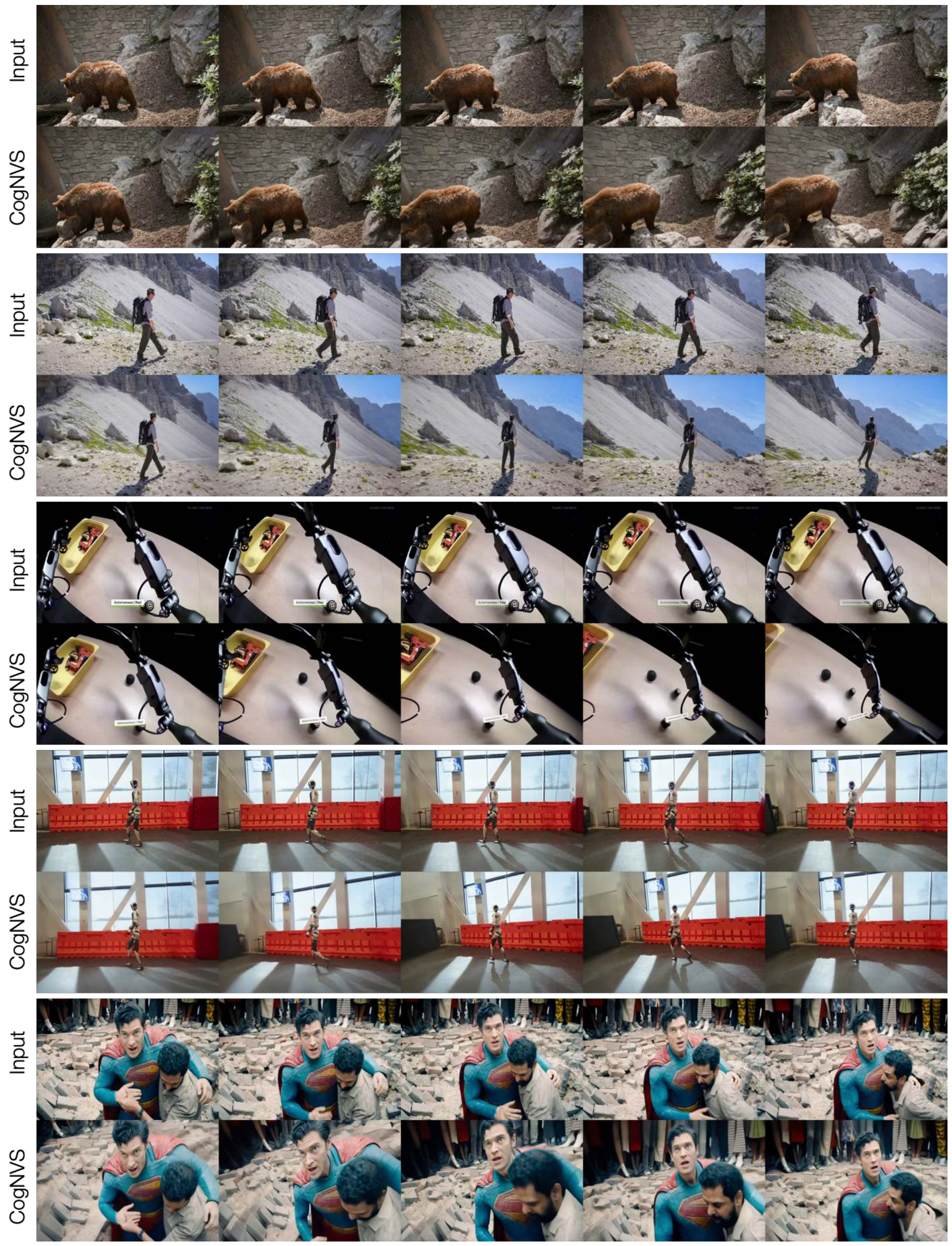

Figure 13: Qualitative results on in-the-wild examples. Part 1 of 2.

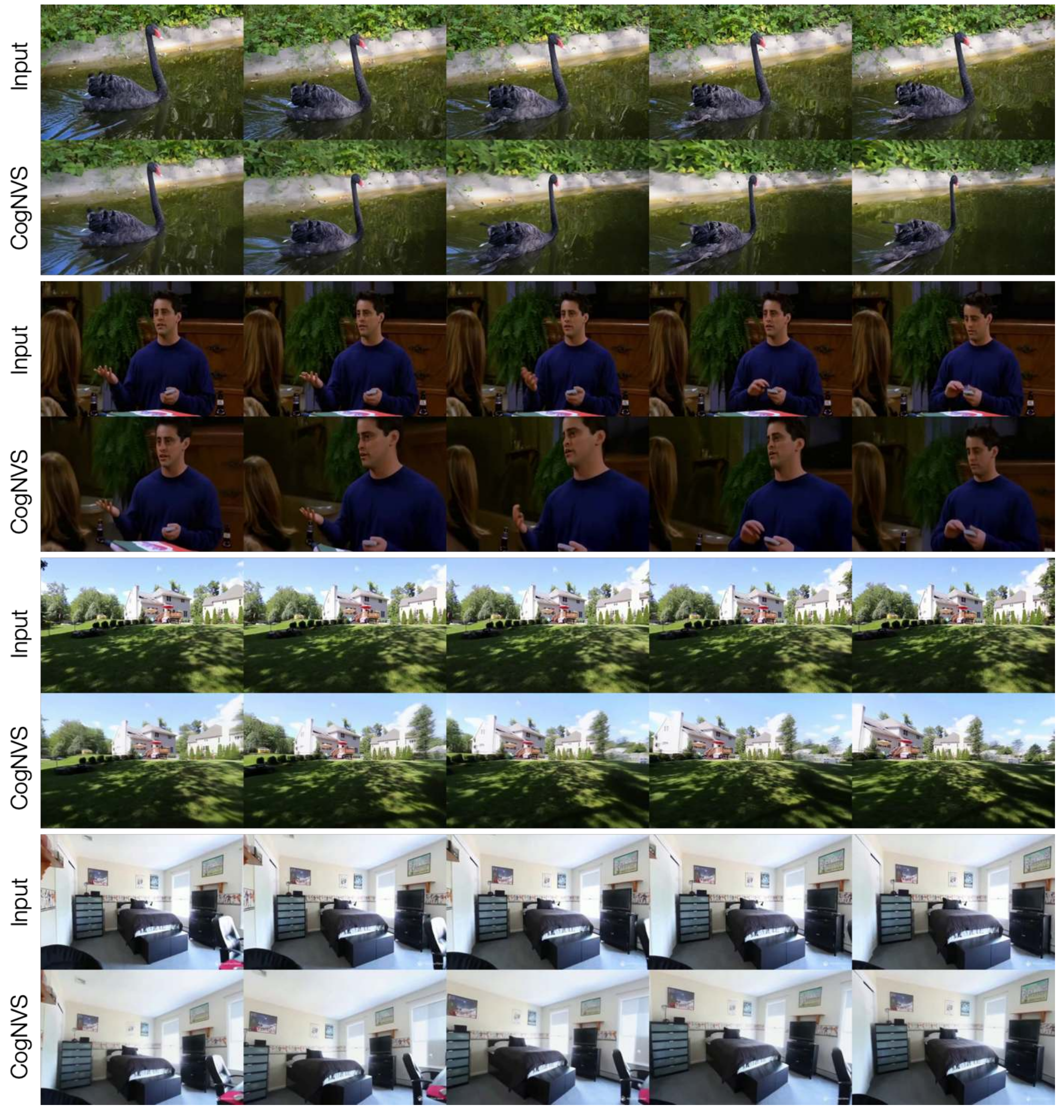

Figure 14: Qualitative results on in-the-wild examples (static scenes included in last two rows). Part 2 of 2.

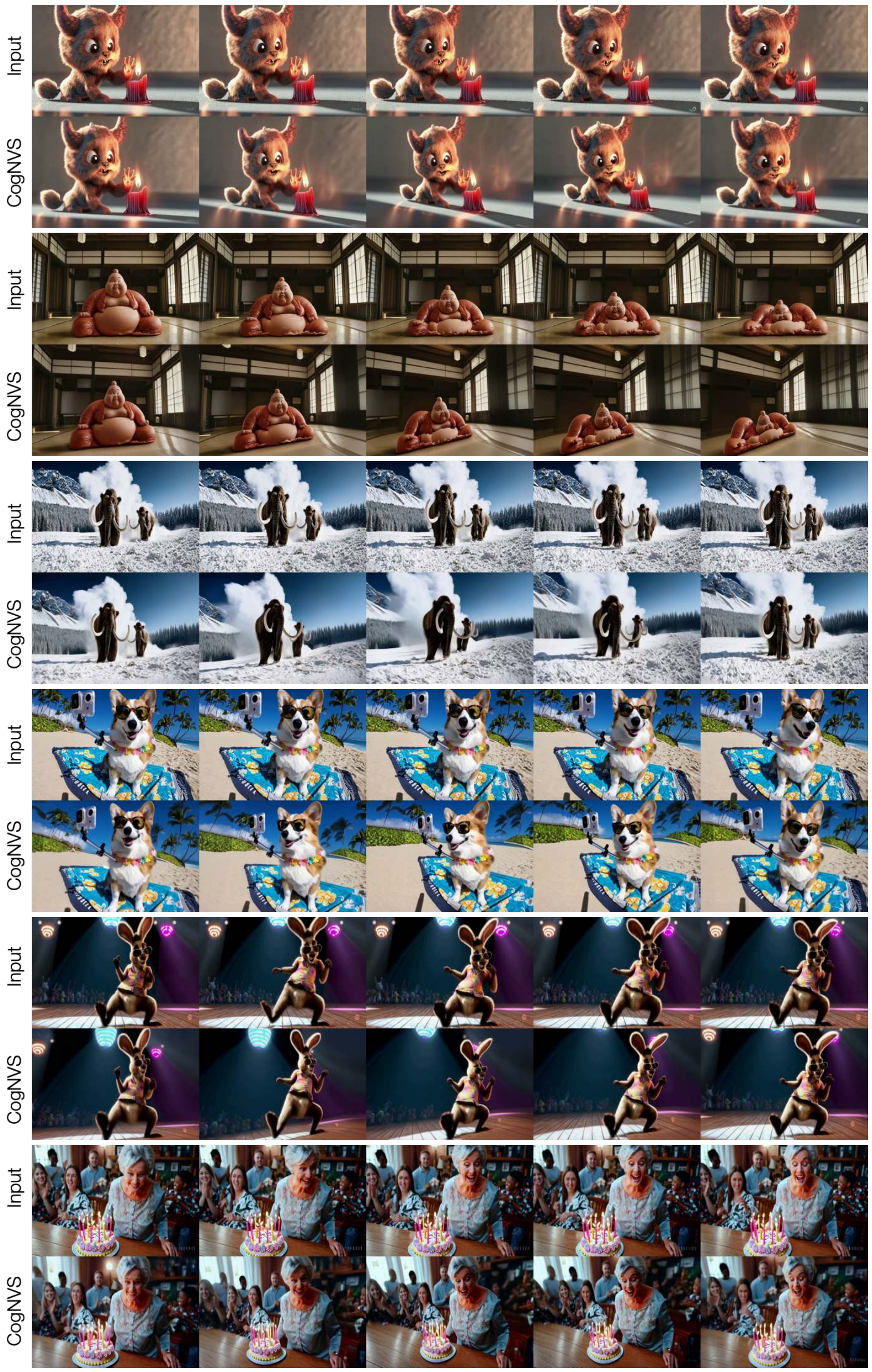

Figure 15: Qualitative results on synthetic videos from SORA.