InverseCrafter: Efficient Video ReCapture as a Latent Domain Inverse Problem

Yeobin Hong*[1], Suhyeon Lee*[1], Hyungjin Chung†[2,3], and Jong Chul Ye†[1]

[1] Graduate School of AI, KAIST, South Korea
[2] EverEx, South Korea
[3] Korea University, South Korea
∗ Equal contribution, † Co-correspondence

Fig. 1: Representative video on camera control *("zoom in", "arc left", "arc right").* Inpainting with editing *("goldfish" to "turtle") in the right bottom box.*

Abstract. Recent approaches in controllable novel view video generation often rely on fine-tuning pre-trained Video Diffusion Models (VDMs). This dominant paradigm is computationally expensive and frequently suffers from catastrophic forgetting of the model's original generative priors. To address this challenge, here we propose **InverseCrafter**, a VDM training-free framework that reformulates novel view video generation as an inpainting-based inverse problem in the latent space, eliminating the need for any annotated 4D training data. The core of our method is to establish operator equivalence by employing a lightweight latent mask encoder to define a latent-domain masking operation via a continuous, multi-channel representation. This principled representation faithfully models the forward process in the latent domain, enabling efficient, backpropagation-free solvers while bypassing the costly bottleneck of repeated VAE operations. InverseCrafter achieves high-fidelity, spatio-temporally coherent novel view synthesis with near-zero additional inference overhead and excels at general-purpose video inpainting and editing by fully preserving the pre-trained VDM's generative capabilities.

Keywords: Inverse Problem · Novel View Synthesis · Video Inpainting

1 Introduction

Generating controllable novel view videos is a long-standing goal in computer vision, with applications in AR/VR, filmmaking, and digital twins. Recent Video

Diffusion Models (VDMs) [6,8,19,27,55] have achieved remarkable progress in generating photorealistic and diverse videos. However, their control mechanism is typically limited to coarse inputs such as text descriptions or a conditioning first frame, offering minimal control over camera motion or scene geometry. This limitation prevents their direct application to novel view synthesis, which requires consistent rendering under user-specified camera trajectories.

One branch of research that has emerged to address the challenge of novel view synthesis is the framing of the problem as target-view recapture given an input video: given a input video, synthesize novel views over time while following a target camera trajectory (*e.g.*, camera control). For example, a line of work fine-tunes VDMs to condition on prospective cameras [2,54,59], which requires large-scale 4D datasets and still generalizes poorly to unseen camera trajectories. Another branch [16,44,60,66] reframes the task as 3D-aware inpainting problem and solve it via VDM fine-tuing. Specifically, these methods leverage 3D reconstruction priors (*e.g.*, depth estimation or point tracking) to create large-scale, pseudo-annotated datasets, and then fine-tune the VDM to inpaint the occluded regions. Finetuning, however, often creates a strong dependency on the specific 3D reconstruction model used during training. Moreover, finetuning often leads to overfitting and catastrophic forgetting, degrading the pretrained model's original text-to-video generation capability and limiting its usefulness as a general generative prior. Furthermore, some methods require further augmenting the VDM architecture (*e.g.*, adding new attention blocks [60,66]), substantially increasing inference latency. Although, per-video optimization approaches [22] avoid large-scale retraining, they overfit to individual scenes and fail to generalize.

A promising training-free alternative is to formulate novel view video generation as an inpainting-based inverse problem at inference time, where the warped video provides partial measurements and a pretrained VDM inpaints the unknown pixels at the inference time. However, adapting diffusion inverse solvers from the image domain to video inpainting is challenging because modern diffusion models operate in a compressed spatio-temporal latent space. First, naively performing pixel-space inpainting (Fig. 2(a)) [23,46,49] requires per-timestep decoding, incurring significant computational overhead and instability due to the nonlinear decoder [43].

To mitigate this issue, recent video inpainting methods – including training-based methods [22,66] and training-free inverse solvers [39,65] (Fig. 2(b)), approximate the pixel-space inpainting problem in the latent space by naively resizing the pixel-space mask to the latent domain. This removes the need for VAE decoding and results in a linearized inverse problem in the full latent space. Although computationally efficient in practice, this approach assumes that spatio-temporal structures and relationships are faithfully preserved in the compressed latent representation – an assumption that often fails, resulting in degraded quality and visible artifacts. Furthermore, despite the linearized formulation, existing video latent inverse solvers do not fully exploit general-purpose linear solvers [39]; instead, they rely on architecture-specific algorithms tailored to particular VDMs [64] or even resort to computationally expensive VDM backpropagation [65].

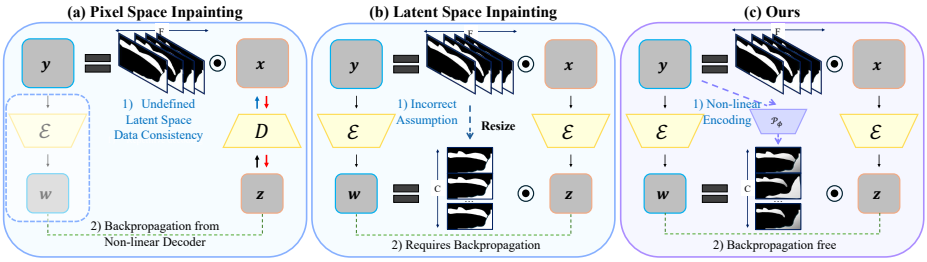

Fig. 2: (a) Pixel-space inpainting [23, 46, 49] requires per-timestep VAE decoding, increasing computational cost. (b) Existing latent space inpainting methods [22, 65, 66] downsamples the pixel space mask via spatio-temporal interpolation. This process results in a single-channel, binary mask that is uniformly broadcast across all C latent channels, ignoring their distinct feature representations and leading to information loss. These methods typically rely on computationally heavy backpropagation. (c) InverseCrafter computes a continuous, C-channel latent mask, enabling efficient latent-space guidance.

To address this gap, we present **InverseCrafter**, a framework that formulates controllable novel view video generation as a latent-domain continuous inpainting-based inverse problem at inference time under a pretrained VDM prior. InverseCrafter requires no VDM fine-tuning and no additional 4D datasets, and it incurs negligible inference overhead, yet it enables controllable novel view video generation from a single input video under arbitrary camera trajectories.

Specifically, consider the pixel-space forward process $\boldsymbol{y} = \boldsymbol{x} \odot \boldsymbol{m}$. Here, $\boldsymbol{y}$ and $\boldsymbol{x}$ denote the measurement and the unknown novel view video, respectively, after the camera view change, and $\boldsymbol{m}$ denotes the mask indicating the unconvered regions induced by the view change. The key idea of InverseCrafter is to establish *operator equivalence* between the pixel-space measurement process and its latent-space counterpart, as highlighted in [43]. More specifically, rather than of naively resizing the pixel-space mask, we learn a latent-space equivalent masking operator that faithfully reproduces the pixel-space forward process $\boldsymbol{y} = \boldsymbol{x} \odot \boldsymbol{m}$ after VAE encoding. To this end, we train a lightweight *latent mask encoder* $\mathcal{P}_\phi$ to predict a continuous C-channel latent mask $\boldsymbol{h}$. This learned masking operator enables the latent operation $\boldsymbol{w} = \boldsymbol{z} \odot \boldsymbol{h}$ to be operator-equivalent to the original pixel-space masking process. This formulation enables efficient latent-space inverse solvers, allowing guidance to be performed entirely in the VDM latent space without per-step VAE decoding or costly backpropagation, significantly reducing computation while maintaining high synthesis quality through more accurate operator modeling.

Our main contributions are summarized as follows:

- We reformulate controllable video generation as an inverse problem at inference time, leveraging pretrained VDM priors without task-specific fine-tuning or additional 4D datasets.
- We introduce a mask encoder to establish operator equivalence via a continuous mask in the latent space, which enables leveraging efficient, backpropagation-free inverse problem solvers without any decoding into pixel space.

– Our framework achieves high-fidelity, spatio-temporally coherent synthesis with near-zero additional inference overhead over standard diffusion sampling.

2 Preliminaries

2.1 Video Inpainting for Generation

Given a partially observed target-view video obtained through geometric warping, the occluded regions must be filled in a temporally coherent manner. When latent diffusion models (LDMs) are used as priors, this completion is performed in the compressed latent space [22, 56, 65, 66]. For instance, NVS-Solver [65] employ diffusion Posterior Sampling (DPS) in the latent space at inference time. Zero4D [39] formulates the 4D video generation as a video interpolation problem in the latent domain by employing mask-based interpolation between the generated output and the measurement.

A critical limitation of these approaches is their reliance on models like Stable Video Diffusion (SVD) [6], whose VAE performs spatial compression only. Both project the degradation operator (pixel space mask) into the latent space via *simple average pooling* over the spatial dimensions. The challenge of correctly formulating the measurement and masking operation in the latent space is a common, unaddressed issue. This problem of binary resizing is significantly exacerbated when it comes to modern Video Diffusion Models (VDMs). Since modern VDMs utilize 3D VAEs for spatio-temporal compression, mapping a 2D pixel space mask to a 3D latent space is ill-defined.

2.2 Diffusion Models for Inverse Problems

Consider the following forward model:

$$\boldsymbol{y} = \mathcal{A}(\boldsymbol{x}) + \boldsymbol{n}, \tag{1}$$

where $\mathcal{A}$ denotes the forward operator and $\boldsymbol{n}$ is additive noise. The goal of the inverse problem is to recover the unknown signal $\boldsymbol{x}$ from the corrupted measurement $\boldsymbol{y}$. Although diffusion models are trained to sample from the prior distribution $p(\boldsymbol{x})$, diffusion-based inverse solvers modify the reverse diffusion dynamics at inference time to approximate posterior sampling, *i.e.*, $\boldsymbol{x} \sim p(\boldsymbol{x} \mid \boldsymbol{y})$ [13]. Most zero-shot approaches guide the reverse diffusion process to reduce the following data consistency term $\|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{x}}_{0|t})\|^2$, where $\hat{\boldsymbol{x}}_{0|t} = \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$ denotes the denoised estimate at timestep t.

Unfortunately, applying such solvers to LDMs—which dominate modern generative modeling—requires additional considerations. Existing latent-space solvers [15, 23, 46, 51] typically decode the intermediate latent variable $\hat{\boldsymbol{z}}_{0|t} = \mathbb{E}[\boldsymbol{z}_0|\boldsymbol{z}_t]$ into the pixel space by decoding with the VAE decoder $\mathcal{D}$, and reduce the measurement consistency term in the decoded pixel space, $\|\boldsymbol{y} - \mathcal{A}(\mathcal{D}(\hat{\boldsymbol{z}}_{0|t}))\|^2$, to update the latent variable $\boldsymbol{z}$. However, enforcing measurement consistency

through the nonlinear decoder $\mathcal{D}$ introduces approximation errors and substantial computational overhead, often leading to over-smoothed or blurry reconstructions.

Recently, SILO [43] addressed this limitation by training a latent forward operator H_θ that approximates the pixel-space forward operator $\mathcal{A}$ through an operator-equivalent mapping between the pixel and latent domains. Specifically, this formulation enables measurement consistency to be evaluated directly in the latent space as $\|\mathcal{E}(\boldsymbol{y}) - H_\theta(\hat{\boldsymbol{z}}_{0|t})\|^2$, where $\mathcal{E}$ refers to the encoder of VAE, thereby eliminating repeated decoding to the pixel domain. Nevertheless, since H_θ is nonlinear, it cannot leverage efficient linear inverse solvers, and backpropagation through the neural network may reduce numerical stability. Moreover, training H_θ requires paired samples $(\boldsymbol{y}, \boldsymbol{x})$, which becomes impractical in our setting where large-scale 4D datasets are scarce.

3 InverseCrafter

3.1 Problem Formulation

To solve the inverse problem of novel view synthesis, we must formulate the measurement $\boldsymbol{y}$ based on 3D geometric constraints. This requires projection of the source video into a target camera view. First, we establish a 3D representation of the scene. We process the source video $\boldsymbol{x}_{1:F}^{src}$ frame-by-frame using a pre-trained monocular depth estimation network [20] to acquire a per-frame estimated depth map $\hat{\mathbf{D}}_{1:F}$. Next, for each frame i, we unproject the 2D pixel coordinates (from the image $\boldsymbol{x}_i^{src}$) into a 3D point cloud $\mathbf{P}_i$ using the estimated depth $\hat{\mathbf{D}}_i$ and an intrinsic camera matrix $\mathbf{K}$. This back-projection function is denoted $\mathit{\Pi}^{-1}$:

$$\mathbf{P}_i = \mathit{\Pi}^{-1}(\boldsymbol{x}_i^{src}, \hat{\mathbf{D}}_i, \mathbf{K}). \tag{2}$$

Once the scene is represented in 3D, we can render it from a new perspective. The 3D point cloud $\mathbf{P}_i$ is transformed into the target camera's coordinate system using a relative transformation matrix $\mathbf{T}_i$. These transformed points are then projected back into a 2D image $\boldsymbol{y}_i$ using the forward projection function $\mathit{\Pi}$:

$$\boldsymbol{y}_i = \mathit{\Pi}(\mathbf{T}_i \cdot \mathbf{P}_i, \mathbf{K}). \tag{3}$$

The warped video $\boldsymbol{y} := \boldsymbol{y}_{1:F}$ represents the pixel-space *measurement*, which contains occlusions. Our goal is then to recover the target-view video $\boldsymbol{x} := \boldsymbol{x}_{1:F}$ from occluded measurement obtained through 3D warping:

$$\boldsymbol{y} = \boldsymbol{m} \odot \boldsymbol{x}, \quad \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{m} \in \mathbb{R}^N \tag{4}$$

where $\odot$ denotes the element-wise product and $N = FHW$ is the pixel space resolution, and $\boldsymbol{m}$ refers to a mask that represents the visiblity induced by the 3D warping process. This defines the pixel-space *forward operator*.

Recent VDMs [55, 63] introduce 3D VAEs that perform spatio-temporal compression, creating the need to reformulate the pixel-domain forward model in Eq. (4) in the latent domain to preserve operator equivalence between the pixel

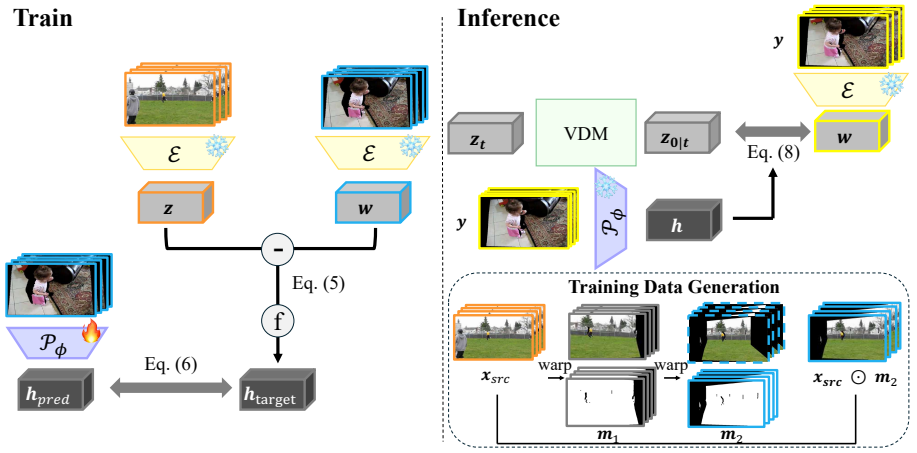

Fig. 3: Overview of InverseCrafter. (a) $\mathcal{P}_\phi$ is trained to project the pixel space degradation operator to the latent domain. **(b)** During inference, $\boldsymbol{z}_t$ is optimized at each t to enforce data consistency Eq. (8), the latent mask derived from the $\mathcal{P}_\phi$.

and latent domains, enabling efficient inverse problem solving in the latent space. Image inpainting inverse solvers for LDMs [24,25] typically construct latent masks by projecting pixel-space masks into the latent space via nearest-neighbor downsampling. This approximation is an ad-hoc adaptation for the LDM, originating from pixel-space diffusion inverse solvers. However, naive downsampling only loosely approximates the true degradation operator. The additional temporal compression in video models further exacerbates this mismatch, motivating a more principled treatment beyond these image-based approaches.

Nevertheless, recent video inpainting methods that operate directly in the latent domain [22,39,60,66] largely adopt direct spatio-temporal projection (*i.e.*, spatio-temporal interpolation) as a naive extension of image-based approaches. This practice has several limitations. First, it implicitly assumes that spatial structures in pixel space are preserved under the non-linear VAE encoder, which is generally false. Second, it treats the latent mask as uniform across channels, despite the fact that latent channels encode highly heterogeneous feature representations, as illustrated in Fig. 2(c) (visualizations in Appendix E.6). Third, some methods [22,66] adopt an overly conservative temporal rule: a latent region is marked as "known" only if its corresponding pixel region is visible in all frames within its temporal group. This strategy becomes a major bottleneck for videos with fast motion. If an object appears in only a subset of frames due to rapid movement or partial occlusion, the entire corresponding latent region is marked as "unknown", which forces unnecessary inpainting and increases reliance on the generative model, as shown in Fig. 9(a).

3.2 Generating Continuous Mask

To address the mask projection inconsistencies outlined previously, we propose learning a lightweight neural network, dubbed the mask encoder $\mathcal{P}_\phi(\cdot) : \mathbb{R}^N \rightarrow$

$\mathbb{R}^{C\times M}$, where C denotes the number of latent channels and $M = fhw$ represents the latent-space resolution. Then, our goal is to predict a latent mask that parameterizes a latent-domain masking operator approximating the operator-equivalent to the pixel-space masking. Specifically, the network takes as input the pixel measurement $\boldsymbol{y}$ and outputs a continuous mask that can be applied *linearly* to the latent $\boldsymbol{z}$, with the same Hadamard product operation as in Eq. (4).

Our training target is derived from the observation that autoencoders are effective at reconstructing degraded data [43]. We define this ground-truth latent mask, $\boldsymbol{h} \in [0,1]^{C\times M}$, as the normalized difference between the latent encoding of the clean video, $\mathcal{E}(\boldsymbol{x})$, and the latent encoding of the masked video, $\mathcal{E}(\boldsymbol{m}\odot\boldsymbol{x})$:

$$\boldsymbol{h} = 1 - \mathrm{f}\left(\mathcal{E}(\boldsymbol{x}) - \mathcal{E}(\boldsymbol{m}\odot\boldsymbol{x})\right), \tag{5}$$

where f stands for normalization using an activation function, scaling the latent difference to a $[0,1]$ range.

However, a primary challenge is obtaining training pairs, as for generating warped videos (Eq. (3)), the corresponding "ground truth" video $\boldsymbol{x}$ is not available unlike in typical inverse problem settings. To resolve this, we employ a double-reprojection strategy from [66] to synthesize the correctly masked counterpart $\boldsymbol{m}\odot\boldsymbol{x}$ from the clean video $\boldsymbol{x}$, which provides the necessary pair to compute $\boldsymbol{h}$ (See Fig. 3-Right).

The mask encoder is then trained to predict this ground-truth latent mask directly from the pixel space measurement with a combination of the L1 loss and the Structural Similarity Index Measure (SSIM) [57] loss:

$$\min_{\phi}\left(\left\|\mathcal{P}_\phi(\boldsymbol{y}) - \boldsymbol{h}\right\|_1 + \lambda\left(1 - \mathrm{SSIM}(\mathcal{P}_\phi(\boldsymbol{y}), \boldsymbol{h})\right)\right) \tag{6}$$

Unlike naive mask resizing, which generates a single-channel binary mask that is uniformly broadcast across all C latent channels (ignoring their distinct feature representations), our mask encoder outputs a continuous, C-channel mask. This high-fidelity representation better captures the mask's influence across the latent features, leading to improved VAE reconstruction and superior measurement consistency in final generated outputs, as detailed in Tab. 8.

3.3 DDS Solver for Latent Inpainting with Continuous Masks

To align with the latent diffusion model, we encode the pixel space measurement into the latent space using the VAE encoder $\mathcal{E}$. Then, under our noiseless linear setting of Eq. (4), the inverse problem in the pixel domain transforms to an inverse problem in the latent space, i.e.

$$\underbrace{\boldsymbol{w}}_{\mathcal{E}(\boldsymbol{y})} = \underbrace{\boldsymbol{h}}_{\mathcal{P}_\phi(\boldsymbol{y})} \odot \underbrace{\boldsymbol{z}}_{\mathcal{E}(\boldsymbol{x})}, \quad \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{h} \in \mathbb{R}^{C\times M}. \tag{7}$$

Notice that the design of our mask encoder $\mathcal{P}_\phi$ is different from that of [43]. The operator encoder used in [43] acts *non-linearly* on the encoded latent $\boldsymbol{z}$ to produce

Algorithm 1 InverseCrafter

Require: Source video $\boldsymbol{x}^{src}$, pixel space measurement $\boldsymbol{y}$, pixel space mask $\boldsymbol{m}$, I2V flow-based model $\boldsymbol{v}^{\theta}$, VAE encoder & Decoder $\mathcal{E}/\mathcal{D}$, hyper-parameter Γ, CG iteration K

$\boldsymbol{h} \leftarrow \mathcal{P}_{\phi}(\boldsymbol{y})$
Initialization $\boldsymbol{z}_T \sim \mathcal{N}(0, \mathbf{I})$
$\boldsymbol{w} \leftarrow \mathcal{E}(\boldsymbol{y})$
$\mathbf{H} \leftarrow \text{diag}(\boldsymbol{h})$
for $t : 1 \rightarrow 0$ **do**
  ▷ Tweedie denoising
  $\hat{\boldsymbol{z}}_{0|t} = \boldsymbol{z}_t - \boldsymbol{v}^{\theta}(\boldsymbol{z}_t)t$
  $\hat{\boldsymbol{z}}_{1|t} = \boldsymbol{z}_t + \boldsymbol{v}^{\theta}(\boldsymbol{z}_t)(1-t)$
  if $t \in \Gamma$ **then**
    ▷ Data consistency
    $\hat{\mathbf{z}}(\mathbf{y}) \leftarrow \text{CG}(\mathbf{I} + \gamma\mathbf{H}^{\top}\mathbf{H}, \hat{\boldsymbol{z}}_{0|t} + \gamma\mathbf{H}^{\top}\boldsymbol{w}, \hat{\mathbf{z}}_{0|t}, K)$
    $\boldsymbol{z}_{t-1} \leftarrow \text{ODESolve}(\hat{\boldsymbol{z}}(\boldsymbol{y}), \hat{\boldsymbol{z}}_{1|t})$
  else
    $\boldsymbol{z}_{t-1} \leftarrow \text{ODESolve}(\hat{\boldsymbol{z}}_{0|t}, \hat{\boldsymbol{z}}_{1|t})$
  end if
end for
$\boldsymbol{x} \leftarrow \mathcal{D}(\boldsymbol{z}_0)$

Algorithm 2 Conjugate Gradient (CG)

Require: $\mathbf{A}, \mathbf{y}, \mathbf{x}_0, M$

$\mathbf{r}_0 \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}_0$
$\mathbf{p}_0 \leftarrow \mathbf{b}_0$
for $i = 0 : K-1$ **do do**
  $\alpha_k \leftarrow \frac{\mathbf{r}_k^{\top}\mathbf{r}}{\mathbf{p}_k^{\top}\mathbf{A}\mathbf{p}_k}$
  $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k\mathbf{p}_k$
  $\mathbf{r}_{k+1} \leftarrow \mathbf{r}_k - \alpha_k\mathbf{A}\mathbf{p}_k$
  $\beta_k \leftarrow \frac{\mathbf{r}_{k+1}^{\top}\mathbf{r}}{\mathbf{r}_k^{\top}\mathbf{r}_k}$
  $\mathbf{p}_{k+1} \leftarrow \mathbf{b}_{k+1} + \beta_k\mathbf{p}_k$
end for
return $\mathbf{x}_K$

the latent measurement $\boldsymbol{w}$, whereas our mask encoder produces a continuous linear mask $\boldsymbol{h}$, which can be applied linearly with a Hadamard product. Thanks to such a design choice, we are free from selecting *any* diffusion model-based inverse problem solvers, even the ones that are specialized for solving linear inverse problems in the pixel space (See appendix Sec. E.1). Among them, we propose to leverage DDS [14]. Namely, the data consistency step of DDS solves the following proximal optimization problem :

$$\min_{\boldsymbol{z}} \frac{\gamma}{2}\|\boldsymbol{w} - \boldsymbol{h} \odot \boldsymbol{z}\|_2^2 + \frac{1}{2}\|\boldsymbol{z} - \hat{\boldsymbol{z}}_{0|t}\|_2^2, \tag{8}$$

which is followed by the ODE solver. Here, $\hat{\boldsymbol{z}}_{0|t} := \mathbb{E}[\boldsymbol{z}_0|\boldsymbol{z}_t]$ is the estimated posterior mean with the flow model $\boldsymbol{v}^{\theta}$, and γ is a hyper-parameter that controls the proximal regularization. Following [14], we solve Eq. (8) using conjugate gradient (CG) iterations (Alg. 2). Specifically, let $\mathbf{H} := \text{diag}(\boldsymbol{h})$. Then, Eq. (8) can be solved with $\text{CG}(\mathbf{I} + \gamma\mathbf{H}^{\top}\mathbf{H}, \hat{\boldsymbol{z}}_{0|t} + \gamma\mathbf{H}^{\top}\boldsymbol{w}, \hat{\boldsymbol{z}}_{0|t}, K)$ with K iterations. The overall algorithm of InverseCrafter is shown in Alg. 1.

4 Experimental Results

4.1 Evaluation Details

Task We evaluate our method on two major video inpainting tasks: video camera control and text-guided video inpainting. Our primary focus is the video camera

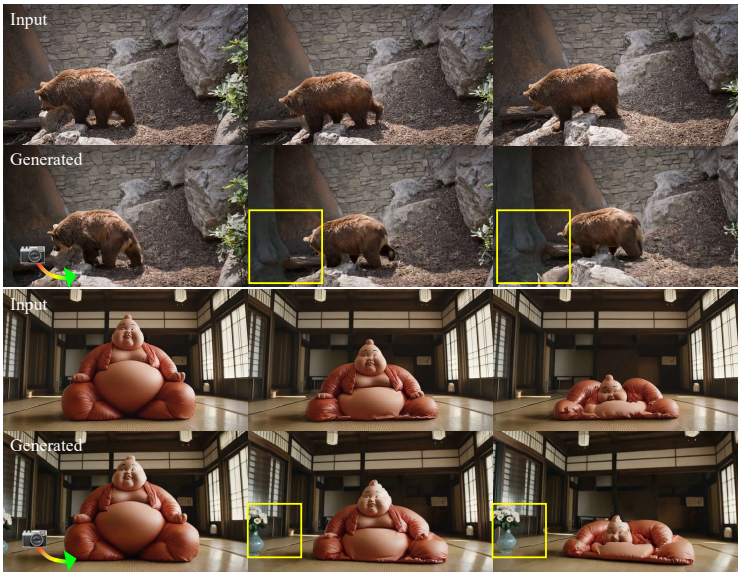

Fig. 4: Video camera control results with novel content generation. (Up) "+a grey tree." (Bottom) "+a flower vase."

control task, for which we use 1,000 videos and corresponding captions from the UltraVideo dataset [62]. We sample one of the six target camera trajectories for each video, where the target trajectories are selected following the work in [22]. For text-guided video inpainting, we apply our proposed video inpainting method to replace objects in the video, guided by target text prompts and source object masks. This evaluation is conducted on 20 videos from the DAVIS [41] dataset, along with all available foreground masks. For each video, we generate five target editing prompts using GPT-5-mini [37], while the corresponding source captions used for prompt construction are obtained using BLIP-2 [32].

Evaluation For video camera control task, we assess performance using a comprehensive set of metrics: measurement consistency - PSNR, LPIPS, SSIM, source consistency - DINO distance [10], Fréchet Video Distance (FVD), and general video quality - subject consistency, temporal flickering, motion smoothness, and overall consistency from VBench [21]. The measurement consistency metric was compared by the warped video, frame by frame. DINO and FVD were calculated by reference to the source video datasets. For video inpainting tasks, we additionally evaluate the generated videos with the editing text-alignment metrics: VLM (see appendix B.2 for details), CLIP Score [42], and Pick Score [26].

4.2 Implementation Details

Training We train our model on the VidSTG [67] dataset. For each of the 7,750 videos in the dataset, we generate 6 double-reprojection datasets from the

Table 1: Quantitative evaluation on the camera control task, showing source video consistency and generation quality together with runtime efficiency.

Method	Runtime	Measurement Consistency			Source Consistency		VBench
	sec ↓	PSNR ↑	LPIPS ↓	SSIM ↑	DINO ↓	FVD ↓	Total Score ↑
ReCamMaster [2]	926	-	-	-	0.0504	136.28	0.8967
GCD [54]	**44**	-	-	-	0.0691	250.85	0.8865
TrajAttention [60]	167	19.44	0.1405	0.6909	0.0562	233.40	0.8756
TrajCrafter [66]	134	<u>28.37</u>	<u>0.0573</u>	**0.8942**	<u>0.0376</u>	120.03	0.8954
NVS-Solver [65]	696	26.68	0.0816	0.8314	0.0393	**96.90**	<u>0.8955</u>
CogNVS [11]	164	15.41	0.4339	0.4016	0.1108	2175.93	0.8193
Ours	<u>71</u>	**29.35**	**0.0485**	<u>0.8827</u>	**0.0359**	<u>99.73</u>	**0.8977**

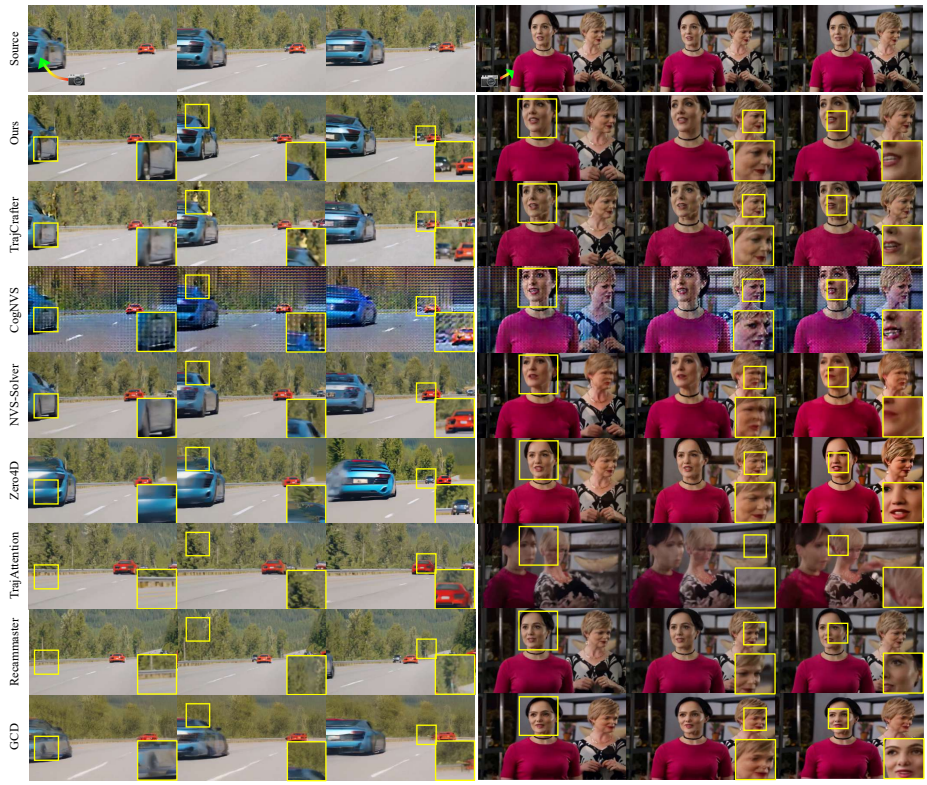

Fig. 5: Qualitative comparison of video camera control. Camera trajectories are *("arc left" "zoom in")*. Our method demonstrates a clear advantage in source consistency and semantically aligned generation. Insets provide magnified views of the regions marked by yellow boxes.

selected camera trajectories, resulting in a total of 46,500 training samples. We utilize DepthCrafter [20] for monocular depth estimation. Our model architecture is based on the Wan2.1 VAE, which we modify to create a lightweight model

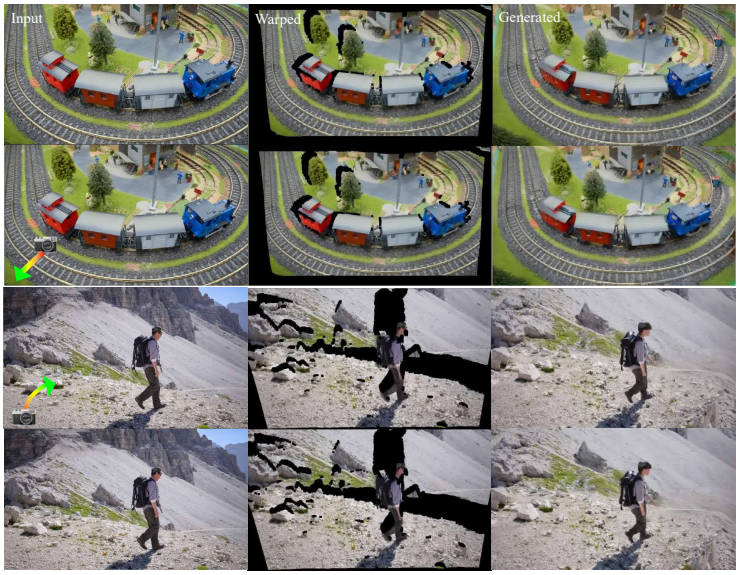

Fig. 6: Visualization of video camera control process as an inpainting inverse problem on a real video example. (*"Zoom out", "Translate Up".*)

by reducing the channel dimension from 96 to 16, resulting in 1.5M parameters. The model is trained at a fixed resolution of 240×416. We employ the AdamW optimizer [36] with a learning rate of 1×10^{-4} and a weight decay of 3×10^{-2}. Training is conducted with a total batch size of 16, distributed as 4 samples per GPU across 4 GPUs, completed in 1 day.

Inference We use Wan2.1-Fun-V1.1-1.3B-InP [55] at a resolution of 480×832 for the main experiments. For encoding masked videos during training or inference, as in [22], we infill the masked region to minimize the train-inference following gap [48]. We use [20] for all training-free methods and TrajectoryCrafter for fair comparison to generate a depth map for each frame.

Compute Resources All experiments are conducted using NVIDIA RTX A6000 GPUs (48GB VRAM). A detailed runtime analysis is provided in Tab. 1.

4.3 Results

Video Camera Control Tab. 1 reports results on the video camera control task compared to major baselines. Despite requiring neither 4D training data nor VDM fine-tuning, and adding virtually no overhead beyond standard diffusion inference, our method achieves strong performance that outperforms or remains competitive with existing approaches, while providing the best performance when runtime is taken into account (Fig. 1). Furthermore, it delivers consistently strong results across all evaluation metrics, indicating reliable behavior in practical video generation settings. Notably, methods that depend on large-scale VDM training

Table 2: Quantitative evaluation on the text-guided video inpainting task, showing source video consistency and generation quality.

Method	Source Consistency	Text Alignment			
	FVD ↓	Overall Consistency ↑	VLM ↑	CLIP Score ↑	PickScore ↑
TrajCrafter [66]	1520.08	0.2191	0.6140	23.88	20.24
NVS-Solver [65]	1648.63	0.2240	<u>0.6400</u>	24.94	20.52
Zero4D [39]	2524.49	0.2257	0.6260	**25.83**	**20.79**
VideoPainter [5]	<u>1432.77</u>	<u>0.2287</u>	0.6290	<u>25.26</u>	20.52
Ours (training-free)	**1422.72**	**0.2289**	**0.6510**	24.98	<u>20.56</u>

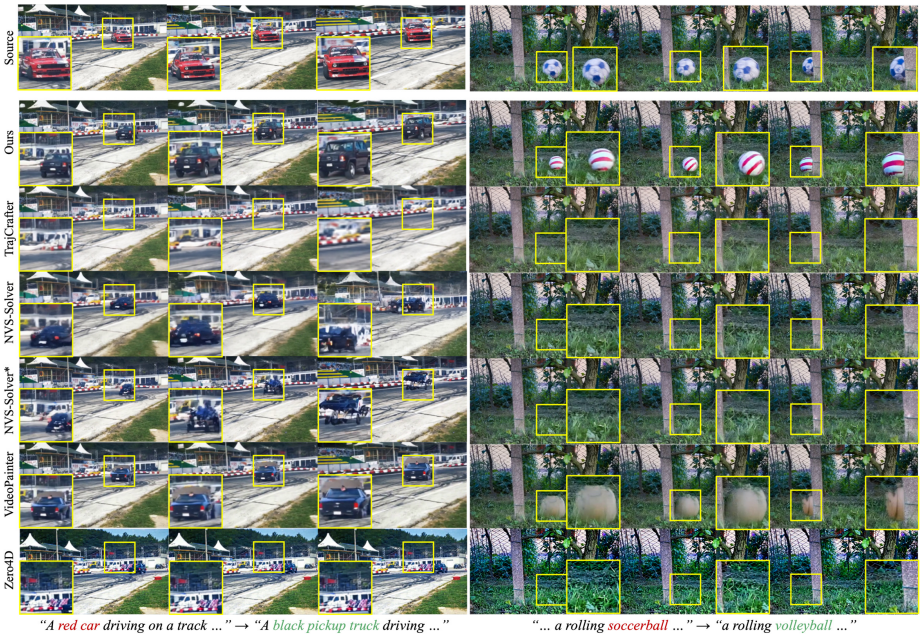

Fig. 7: Qualitative comparison of video inpainting with editing. InverseCrafter achieves better target text alignment compared to the baselines. NVS-Solver indicates NVS-Solver (post), NVS-Solver* indicates NVS-Solver (dgs).

or heavy gradient-based optimization incur high computational cost, whereas our approach attains competitive or superior performance while remaining lightweight.

The qualitative comparisons in Fig. 5 further show that baseline training-based or backpropagation methods (TrajectoryCrafter, NVS-Solver (post), Trajectory-Attention, ReCamMaster) generate artifacts (*e.g.* in the trees) or fail to faithfully preserve the source video's content (*e.g.* background car and women's faces). Backpropagation-free methods (NVS-Solver (dgs), Zero4D) fail to capture the video's motion (*e.g.* background car), generate boundary artifacts, or result in highly saturated videos. On the other hand, our method maintains higher fidelity

Table 3: Quantitative results on VAE reconstruction of different masking strategies.

Method	Measurement Consistency		
	PSNR ↑	LPIPS ↓	SSIM ↑
Binary mask resize	27.76	0.0559	**0.8794**
Elementwise division	26.59	0.0688	0.8458
Ours (target)	**28.10**	**0.0526**	0.8764
Ours ($\mathcal{P}_\phi$)	27.97	0.0515	0.8772

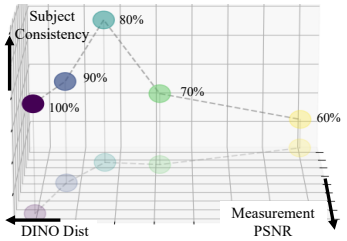

Fig. 8: Quantitative comparison on Conjugate Gradient Step Hyperparameter α.

to the source while preserving visual coherence, demonstrating the practical benefit and reliability of our efficient formulation.

To illustrate the generation process on a real video, Fig. 6 shows how video camera control is performed by casting the task as an inpainting inverse problem. Fig. 4 presents text-conditioned novel content generation results, where the pre-trained VDM prior ensures strong semantic alignment and contextually consistent synthesis under complex geometric manipulation.

Video Inpainting with Editing The video inpainting with editing task provides a valid ground-truth latent mask $\boldsymbol{h}$ at inference time[4], without requiring training of $\mathcal{P}_\phi$. We provide first-frame conditioning input for training-free methods (NVS-Solver, Zero4D) and VideoPainter, as first-frame conditioning is essential for I2V models. The conditioning frame is obtained using an off-the-shelf image inpainting model [31].

The proposed method outperforms the baselines in source distribution and video text alignment metrics in Tab. 2. While Zero4D [39] outperforms image based text alignment metrics, this comes from the fact that output video of Zero4D is nearly static as shown in Fig. 7. In the second example, InverseCrafter uniquely generates the volleyball in the masked region, while also matching the measurement, whereas all baselines fail to generate regarding the target text. In both examples, NVS-Solver (post) deviates a lot from the measurement, NVS-Solver (dgs) and VideoPainter generates boundary artifacts and inferior results. This implies that our method can be used as a general inpainting inverse solver, integrating text descriptions by fully utilizing the pre-trained model's generative prior.

4.4 Ablation Studies

Tab. 3 compares our continuous latent mask formulation with several alternative mask projections on a subset of the DAVIS [41] dataset. Conventional binary mask resizes $\times 8$ nearest-neighbor spatially and logical AND downsampling temporally.

[4] Motivated by this property, a similar training-free projection strategy can also be applied to the video camera control task; see Sec. D for details.

Fig. 9: **(a)** VAE reconstruction of binary downsampled mask and ours. **(b)** Video camera control results of Pixel Space DDS and ours. **(c)** Ablation results for α.

Binary mask resizing results in a single-channel, conservative mask. Another option is elementwise division, dividing the masked latents by the clean latents. However, this is numerically unstable when the denominator contains small values. Overall, our method achieves stronger measurement consistency than these alternatives, indicating that a channel-wise continuous mask better captures how pixel-space occlusions propagate through the VAE encoder.

Additionally, Fig. 9(b) compares our method with conventional pixel space DDS as in prior methods [28, 29], which requires additional VAE encoding and decoding at every timestep, a costly bottleneck that our method entirely avoids. Furthermore, as highlighted by the insets, pixel space Conjugate Gradient introduces severe color and textural inconsistencies. In contrast, our proposed method enables seamless integration with measurement without these artifacts.

Fig. 8 and Fig. 9(c) illustrate the quantitative and qualitative effect of the conjugate gradient hyperparameter Γ on inpainting performance on UltraVideo [62] dataset, revealing the trade-off between measurement / source consistency and generation fidelity. We set Γ as $\Gamma = \{t | t \geq 1 - \alpha\}$. When using a larger α (*i.e.*, applying the measurement consistency step earlier in the diffusion process), the result exhibits stronger consistency with the source measurement and improved content faithfulness. In contrast, a smaller α delays the measurement consistency

update to later diffusion steps, emphasizing the model's generative prior and producing more visually refined outputs. This demonstrates that α effectively controls the balance between the diffusion prior and measurement constraints during posterior sampling, where we observe that setting $\alpha = 0.8$ consistently achieves a favorable balance between semantic alignment and structure preservation.

5 Conclusion

In this paper, we presented a novel and efficient framework for novel-view video generation by casting the task as an inverse problem fully defined in the latent space. The core of our method is to establish operator equivalence between the pixel-space measurement process and its latent-space counterpart via a lightweight latent mask encoder. $\mathcal{P}_\phi$ projects pixel-space masks into continuous, multi-channel latent representations. Our experiments demonstrate that InverseCrafter achieves superior measurement consistency while preserving comparable generation quality in the camera control task. We further show its effectiveness as a general-purpose video inpainting solver for editing tasks. Overall, our framework offers a versatile and lightweight solution for controllable video synthesis, eliminating the need for computationally expensive VDM fine-tuning and introducing negligible inference-time overhead.

Limitations and Future Work. Although our solver is efficient, its speed is limited by the multi-step diffusion sampling process. Our method inherits the bias of the pre-trained VDM. The initial warping stage is a major source of artifacts due to its dependence on monocular depth estimates. We expect that performance will further scale with improved depth estimation. The mask encoder is currently tailored to a specific VAE architecture, which may limit applicability to other latent space. For larger viewpoint changes with more occlusions, the VAE-derived latent-mask target may become less reliable because the masked video contains limited valid visual context. Moreover, the current target mask is constructed by normalizing the latent difference; future work may improve this with more principled latent mask target construction.

Acknowledgements

This work was supported by the National Research Foundation of Korea under Grant RS-2024-00336454, Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2022-II220984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation), the Advanced GPU Utilization Support Program funded by the Government of the Republic of Korea (Ministry of Science and ICT) (02-26-01-0404), and the AI Computing Infrastructure Enhancement (GPU Rental Support) User Support Program funded by the Ministry of Science and ICT (MSIT), Republic of Korea (RQT-25-120217).

References

1. Bahmani, S., Skorokhodov, I., Siarohin, A., Menapace, W., Qian, G., Vasilkovsky, M., Lee, H.Y., Wang, C., Zou, J., Tagliasacchi, A., et al.: Vd3d: Taming large video diffusion transformers for 3d camera control. arXiv preprint arXiv:2407.12781 (2024)
2. Bai, J., Xia, M., Fu, X., Wang, X., Mu, L., Cao, J., Liu, Z., Hu, H., Bai, X., Wan, P., Zhang, D.: Recammaster: Camera-controlled generative rendering from a single video (2025), https://arxiv.org/abs/2503.11647
3. Bai, J., Xia, M., Wang, X., Yuan, Z., Fu, X., Liu, Z., Hu, H., Wan, P., Zhang, D.: Syncammaster: Synchronizing multi-camera video generation from diverse viewpoints. arXiv preprint arXiv:2412.07760 (2024)
4. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
5. Bian, Y., Zhang, Z., Ju, X., Cao, M., Xie, L., Shan, Y., Xu, Q.: Videopainter: Any-length video inpainting and editing with plug-and-play context control. In: Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers. pp. 1–12 (2025)
6. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)
7. Blau, Y., Michaeli, T.: The perception-distortion tradeoff. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6228–6237 (2018)
8. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators (2024), https://openai.com/research/video-generation-models-as-world-simulators
9. Caelles, S., Pont-Tuset, J., Perazzi, F., Montes, A., Maninis, K.K., Van Gool, L.: The 2019 davis challenge on vos: Unsupervised multi-object segmentation. arXiv:1905.00737 (2019)
10. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
11. Chen, K., Khurana, T., Ramanan, D.: Reconstruct, inpaint, test-time finetune: Dynamic novel-view synthesis from monocular videos (2026), https://arxiv.org/abs/2507.12646
12. Chung, H., Kim, J., Mccann, M.T., Klasky, M.L., Ye, J.C.: Diffusion posterior sampling for general noisy inverse problems. In: International Conference on Learning Representations (2023), https://openreview.net/forum?id=OnD9zGAGT0k
13. Chung, H., Kim, J., Ye, J.C.: Diffusion models for inverse problems. arXiv preprint arXiv:2508.01975 (2025)
14. Chung, H., Lee, S., Ye, J.C.: Decomposed diffusion sampler for accelerating large-scale inverse problems. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=DsEhqQtfAG
15. Chung, H., Ye, J.C., Milanfar, P., Delbracio, M.: Prompt-tuning latent diffusion models for inverse problems. In: International Conference on Machine Learning. pp. 8941–8967. PMLR (2024)

16. Gu, Z., Yan, R., Lu, J., Li, P., Dou, Z., Si, C., Dong, Z., Liu, Q., Lin, C., Liu, Z., Wang, W., Liu, Y.: Diffusion as shader: 3d-aware video diffusion for versatile video generation control (2025), https://arxiv.org/abs/2501.03847
17. He, H., Xu, Y., Guo, Y., Wetzstein, G., Dai, B., Li, H., Yang, C.: Cameractrl: Enabling camera control for text-to-video generation. arXiv preprint arXiv:2404.02101 (2024)
18. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems **33**, 6840–6851 (2020)
19. Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. Advances in Neural Information Processing Systems **35**, 8633–8646 (2022)
20. Hu, W., Gao, X., Li, X., Zhao, S., Cun, X., Zhang, Y., Quan, L., Shan, Y.: Depthcrafter: Generating consistent long depth sequences for open-world videos (2024), https://arxiv.org/abs/2409.02095
21. Huang, Z., Zhang, F., Xu, X., He, Y., Yu, J., Dong, Z., Ma, Q., Chanpaisit, N., Si, C., Jiang, Y., et al.: Vbench++: Comprehensive and versatile benchmark suite for video generative models. arXiv preprint arXiv:2411.13503 (2024)
22. Jeong, H., Lee, S., Ye, J.C.: Reangle-a-video: 4d video generation as video-to-video translation (2025), https://arxiv.org/abs/2503.09151
23. Kim, J., Kim, B.S., Ye, J.C.: Flowdps: Flow-driven posterior sampling for inverse problems. arXiv preprint arXiv:2503.08136 (2025)
24. Kim, J., Park, G.Y., Chung, H., Ye, J.C.: Regularization by texts for latent diffusion inverse solvers. arXiv preprint arXiv:2311.15658 (2023)
25. Kim, J., Park, G.Y., Ye, J.C.: Dreamsampler: Unifying diffusion sampling and score distillation for image manipulation. arXiv preprint arXiv:2403.11415 (2024)
26. Kirstain, Y., Polyak, A., Singer, U., Matiana, S., Penna, J., Levy, O.: Pick-a-pic: An open dataset of user preferences for text-to-image generation (2023), https://arxiv.org/abs/2305.01569
27. Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al.: Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024)
28. Kwon, T., Ye, J.C.: Solving video inverse problems using image diffusion models (2025), https://arxiv.org/abs/2409.02574
29. Kwon, T., Ye, J.C.: Vision-xl: High definition video inverse problem solver using latent image diffusion models (2025), https://arxiv.org/abs/2412.00156
30. Labs, B.F.: Flux (2024), uRL https://blackforestlabs.ai/
31. Labs, B.F.: Flux.1 fill [dev]. https://huggingface.co/black-forest-labs/FLUX.1-Fill-dev (2025), model release. License: flux-1-dev-non-commercial-license
32. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: International conference on machine learning. pp. 19730–19742. PMLR (2023)
33. Li, T., Zheng, G., Jiang, R., Wu, T., Lu, Y., Lin, Y., Li, X., et al.: Realcam-i2v: Real-world image-to-video generation with interactive complex camera control. arXiv preprint arXiv:2502.10059 (2025)
34. Lipman, Y., Chen, R.T.Q., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=PqvMRDCJT9t
35. Liu, X., Gong, C., qiang liu: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=XVjTT1nw5z

36. Loshchilov, I.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
37. OpenAI: Openai platform: gpt-5-mini. https://platform.openai.com/docs/models/gpt-5-mini (2025), accessed on 2025-09-14
38. Papamakarios, G., Nalisnick, E., Rezende, D.J., Mohamed, S., Lakshminarayanan, B.: Normalizing flows for probabilistic modeling and inference. Journal of Machine Learning Research **22**(57), 1–64 (2021)
39. Park, J., Kwon, T., Ye, J.C.: Zero4d: Training-free 4d video generation from single video using off-the-shelf video diffusion (2025), https://arxiv.org/abs/2503.22622
40. Patel, M., Wen, S., Metaxas, D.N., Yang, Y.: Steering rectified flow models in the vector field for controlled image generation (2024), https://arxiv.org/abs/2412.00100
41. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675 (2017)
42. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
43. Raphaeli, R., Man, S., Elad, M.: Silo: Solving inverse problems with latent operators. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10570–10580 (2025)
44. Ren, X., Shen, T., Huang, J., Ling, H., Lu, Y., Nimier-David, M., Müller, T., Keller, A., Fidler, S., Gao, J.: Gen3c: 3d-informed world-consistent video generation with precise camera control (2025), https://arxiv.org/abs/2503.03751
45. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10684–10695 (2022)
46. Rout, L., Raoof, N., Daras, G., Caramanis, C., Dimakis, A., Shakkottai, S.: Solving linear inverse problems provably via posterior sampling with latent diffusion models. Advances in Neural Information Processing Systems **36** (2024)
47. Rout, L., Raoof, N., Daras, G., Caramanis, C., Dimakis, A.G., Shakkottai, S.: Solving linear inverse problems provably via posterior sampling with latent diffusion models (2023), https://arxiv.org/abs/2307.00619
48. Saxena, S., Herrmann, C., Hur, J., Kar, A., Norouzi, M., Sun, D., Fleet, D.J.: The surprising effectiveness of diffusion models for optical flow and monocular depth estimation (2023), https://arxiv.org/abs/2306.01923
49. Song, B., Kwon, S.M., Zhang, Z., Hu, X., Qu, Q., Shen, L.: Solving inverse problems with latent diffusion models via hard data consistency. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=j8hdRqOUhN
50. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: 9th International Conference on Learning Representations, ICLR (2021)
51. Spagnoletti, A., Prost, J., Almansa, A., Papadakis, N., Pereyra, M.: Latino-pro: Latent consistency inverse solver with prompt optimization. arXiv preprint arXiv:2503.12615 (2025)
52. Sun, W., Chen, S., Liu, F., Chen, Z., Duan, Y., Zhang, J., Wang, Y.: Dimensionx: Create any 3d and 4d scenes from a single image with controllable video diffusion. arXiv preprint arXiv:2411.04928 (2024)

53. Unterthiner, T., Van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Towards accurate generative models of video: A new metric & challenges. arXiv preprint arXiv:1812.01717 (2018)
54. Van Hoorick, B., Wu, R., Ozguroglu, E., Sargent, K., Liu, R., Tokmakov, P., Dave, A., Zheng, C., Vondrick, C.: Generative camera dolly: Extreme monocular dynamic novel view synthesis. In: European Conference on Computer Vision. pp. 313–331. Springer (2024)
55. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Feng, M., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z.F., Liu, Z.: Wan: Open and advanced large-scale video generative models (2025), https://arxiv.org/abs/2503.20314
56. Wang, Y., Yu, J., Zhang, J.: Zero-shot image restoration using denoising diffusion null-space model. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=mRieQgMtNTQ
57. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
58. Watson, D., Saxena, S., Li, L., Tagliasacchi, A., Fleet, D.J.: Controlling space and time with diffusion models. arXiv preprint arXiv:2407.07860 (2024)
59. Wu, R., Gao, R., Poole, B., Trevithick, A., Zheng, C., Barron, J.T., Holynski, A.: Cat4d: Create anything in 4d with multi-view video diffusion models. arXiv preprint arXiv:2411.18613 (2024)
60. Xiao, Z., Ouyang, W., Zhou, Y., Yang, S., Yang, L., Si, J., Pan, X.: Trajectory attention for fine-grained video motion control. arXiv preprint arXiv:2411.19324 (2024)
61. Xu, D., Nie, W., Liu, C., Liu, S., Kautz, J., Wang, Z., Vahdat, A.: Camco: Camera-controllable 3d-consistent image-to-video generation. arXiv preprint arXiv:2406.02509 (2024)
62. Xue, Z., Zhang, J., Hu, T., He, H., Chen, Y., Cai, Y., Wang, Y., Wang, C., Liu, Y., Li, X., et al.: Ultravideo: High-quality uhd video dataset with comprehensive captions. arXiv preprint arXiv:2506.13691 (2025)
63. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072 (2024)
64. Yesiltepe, H., Yanardag, P.: Dynamic view synthesis as an inverse problem (2025), https://arxiv.org/abs/2506.08004
65. You, M., Zhu, Z., Liu, H., Hou, J.: Nvs-solver: Video diffusion model as zero-shot novel view synthesizer. arXiv preprint arXiv:2405.15364 (2024)
66. YU, M., Hu, W., Xing, J., Shan, Y.: Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models (2025), https://arxiv.org/abs/2503.05638
67. Zhang, Z., Zhao, Z., Zhao, Y., Wang, Q., Liu, H., Gao, L.: Where does it exist: Spatio-temporal video grounding for multi-form sentences (2020), https://arxiv.org/abs/2001.06891