Computers & Graphics 138 (2026) 104645

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on SMI2026

PSF-4D: A progressive sampling framework for view-consistent 4D editing ☆

Hasan Iqbal [a,*,1], Nazmul Karim [b,1], Azib Farooq [b], Umar Khalid [c], Chen Chen [c], Zichun Zhang [a], Jing Hua [a]

[a] Department of Computer Science, Wayne State University, USA
[b] University of Central Florida, USA
[c] Center for Research in Computer Vision, University of Central Florida, USA

ARTICLE INFO

Keywords:
Diffusion Models
4D editing
4D Gaussian Splatting

ABSTRACT

Instruction-guided generative models, especially those using text-to-image (T2I) and text-to-video (T2V) diffusion frameworks, have advanced the field of content editing in recent years. To extend these capabilities to 4D scene, we introduce a progressive sampling framework for 4D editing (PSF-4D) that ensures temporal and multi-view consistency by intuitively controlling the noise initialization during forward diffusion. For temporal coherence, we design a correlated Gaussian noise structure that links frames over time, allowing each frame to depend meaningfully on prior frames. Additionally, to ensure spatial consistency across views, we implement a cross-view noise model, which uses shared and independent noise components to balance commonalities and distinct details among different views. To further enhance spatial coherence, PSF-4D incorporates view-consistent iterative refinement, embedding view-aware information into the denoising process to ensure aligned edits across frames and views. Our approach enables high-quality 4D editing without relying on external models, addressing key challenges in previous methods. Through extensive evaluation on multiple benchmarks and multiple editing aspects (e.g., style transfer, multi-attribute editing, object removal, local editing, etc.), we show the effectiveness of our proposed method. Experimental results demonstrate that our proposed method outperforms state-of-the-art 4D editing methods in diverse benchmarks.

1. Introduction

The field of 4D scene reconstruction has witnessed significant advancements with the development of dynamic neural 3D representations [1,2], including K-Planes [3], HexPlanes [4], and dynamic 3D Gaussian fields [5,6]. These methods have substantially improved our ability to capture and model temporally evolving scenes with high fidelity. Conceptually, a 4D scene can be viewed as a pseudo-3D representation [7], where each viewpoint corresponds to a video rather than a static image. Consequently, adapting a text-to-image (T2I) model for 4D scene editing necessitates extending it into a text-to-video (T2V) framework.

However, ensuring temporal and multiview consistency in edits remains a key challenge. Variations in modifications across different viewpoints and time frames can introduce significant inconsistencies, complicating interactive 4D scene editing. Recent approaches, such as Control4D [8] and Instruct-4D-to-4D [7], have made strides toward addressing these challenges, but they rely heavily on auxiliary models beyond diffusion-based architectures. For example, Instruct-4D-to-4D employs a pre-trained optical flow model [9] to enforce consistency, while Control4D integrates a GAN-based refinement module. These dependencies introduce inherent limitations: GAN training can be unstable, optical flow models may struggle in complex or unseen scenarios, and Instruct-4D-to-4D's anchor-aware attention mechanism can lead to inconsistencies depending on anchor selection. In this work, we aim to overcome these limitations by leveraging the internal forward and reverse sampling processes of diffusion models, ensuring a more principled and end-to-end diffusion-based approach for 4D scene editing.

Building on these insights, we present **PSF-4D**, a novel 4D editing framework that introduces *progressive noise sampling* and *iterative refinement* to enhance generation quality. Prior works [10–14] have demonstrated that careful noise control and multiview geometry information can significantly improve diffusion-based synthesis. Inspired by this, we propose a targeted manipulation of noise initialization

☆ This article is part of a Special issue entitled: 'CAG_SS_SMI2026' published in Computers & Graphics.

* Corresponding author.
E-mail addresses: hasan.iqbal.cs@wayne.edu (H. Iqbal), nazmul.karim170@gmail.com (N. Karim), azibfarooq10@gmail.com (A. Farooq), umar.khalid@ucf.edu (U. Khalid), chen.chen@crcv.ucf.edu (C. Chen), zichunzhong@wayne.edu (Z. Zhang), jinghua@wayne.edu (J. Hua).

[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.cag.2026.104645
Received 6 May 2026; Accepted 29 May 2026
Available online 26 June 2026
0097-8493/© 2026 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

during the forward diffusion phase, coupled with view-consistent noisy latent refinement in the reverse diffusion phase. To ensure temporal coherence, we leverage the autoregressive nature of temporal data by explicitly modeling relationships across the sequence.

However, robust 4D editing demands not only temporal consistency but also view consistency across perspectives. To address this, we introduce a *cross-view noise model* within the Text-to-Video (T2V) framework, enhancing spatial alignment across views. CNM builds upon principles of 3D multiview geometry, enforcing spatial coherence by decomposing noise into two complementary components: a *shared component* that captures cross-view similarity and an *independent component* that preserves view-specific variations. While noise initialization plays a key role in maintaining coherence, sometimes it alone is insufficient to enforce consistency across edits. To this end, we develop a *view-consistent iterative refinement* mechanism that directly integrates view-aware editing signals into the denoising stages of the diffusion model. This strategy enforces consistent modifications across perspectives while retaining necessary view-dependent details, ensuring both temporal and spatial coherence in the final 4D output. Our key contributions are summarized below:

- We propose a novel 4D scene editing framework that exploits different aspects of the diffusion process to ensure temporal and multi-view coherence during editing. After a careful analysis of the forward and reverse diffusion process (*Section* 3), we identify that correlation in noise plays a crucial role in enabling superior 4D scene editing.
- By intuitively controlling noise correlation in the forward diffusion process, we establish coherence across noisy video frames captured from different views (*Section* 5). The proposed noise correlation translates into significantly reduced artifacts in the edited 4D model.
- Establishing noise correlation may not produce the desired editing, especially during complex scene editing. Therefore, a refinement strategy focusing solely on improving view consistency is introduced to further refine the edited 4D model (*Section* 5.2.2).

Through comprehensive evaluations across various benchmarks and diverse editing tasks, we demonstrate the superior results of PSF-4D (*Sections* 6.1 *and* 6.2).

2. Related work

Diffusion-based video editing. Diffusion-based generative models have excelled in text-guided image editing [15–21], but adapting them for video editing presents unique challenges, especially in preserving temporal coherence. A common approach is to transform Text-to-Image (T2I) models into Text-to-Video (T2V) models. For instance, Tune-A-Video [22] adds temporal self-attention layers for one-shot fine-tuning, while Make-A-Video [23] and MagicVideo [24] incorporate spatio-temporal attention (ST-Attn) to handle temporal aspects. Other recent methods focus on localizing edits within the video, such as Video-P2P [25], which uses decoupled-guidance attention to ensure semantic consistency, and Pix2Video [26], which propagates anchor frame edits.

4D scene editing. Earlier 4D scene editing methods [27,28] remain limited in advanced, real-time editing capabilities. Recent advancements, such as Control4D [8] and Instruct-4D-to-4D (I4D-to-4D) [7], have made progress in improving consistency for 4D scene editing, yet these methods rely significantly on supplementary models beyond diffusion-based approaches. For instance, I4D-to-4D integrates a pre-trained optical flow model [9] to maintain temporal alignment across frames, while Control4D uses a GAN-based framework to manage dynamic adjustments and edits. However, GAN architectures are known for their instability, which can complicate training and reduce reliability during edits, while optical flow models may yield unreliable results in novel or complex scenes where accurate flow guidance is essential. Consequently, the added complexity of these external models often restricts the overall performance gains, highlighting the need for more robust and adaptable solutions in 4D editing. Our proposed framework solely focuses on controlling the diffusion process rather than relying on external models.

Noise correlation in diffusion models. Recent work has demonstrated that using structured blue noise can improve the perceptual quality of 2D images [29], while others improve temporal consistency in sequential data by introducing an autoregressive structure directly into the noise distribution [13,30]. In contrast, other methods address 3D consistency by explicitly aligning noise predictions across multiple views using geometric correspondences [14]. We take motivation from [30] to design our autoregressive noise framework. However, while [30] applies autoregressive noise correlations to single-view video diffusion for temporal consistency, PSF-4D operates across multi-view 4D scene data, where both temporal and cross-view spatial coherence must be simultaneously enforced. Therefore, our work is the first to propose a unified 4D correlated noise structure that ensures both spatial and temporal consistency in dynamic 3D scenes.

Table 1
Cosine Similarity Scores between latents from different sources. We first get the VAE latents for $K \times F$ 2D image frames of the same 4D scene. We also randomly take the same number of 2D frames from different scenes. We then calculate the similarity score (a) between latents from the same scene, **b)** between latents from different scenes, **c)** between noisy latents from the same scene with i.i.d. noise, **d)** between noisy latents from different scenes with i.i.d. noise, **e)** between noisy latents from different scenes with the proposed ANM and CNM noise models (Section 5.1, 5.2.1) It can be seen that i.i.d. noise destroys the correlation between frames of a scene. As a result, the model treats the noisy latents from the same scene and the latents from different scenes similarly. Our proposed method addresses this issue by preserving the correlation between latents through a noise prior. We use $\gamma = 0.65$ (Eq. (3)) and $\lambda = 0.7$ (Eq. (4)) for calculating these scores.

Latents	Similarity score
(a) Same Scene	0.9641 ±0.0121
(b) Different Scene	0.0651 ±0.2779
(c) Same Scene + i.i.d. Noise	0.0984 ±0.2492
(d) Different Scene + i.i.d. Noise	0.0067 ±0.2748
(e) Same Scene + ANM & CNM Noise	0.4216 ±0.1328

3. Noise correlation in diffusion process

In this section, we provide a more in-depth explanation of why noise correlation matters in 4D editing. In Table 1, we present the cosine similarity scores between latents from the same and different scenes. The latent representations of frames drawn from the same 4D scene exhibited very high cosine similarity, reflecting their shared underlying structure, whereas latents originating from different scenes displayed negligible correspondence, as expected for uncorrelated content. When independent, identically distributed Gaussian noise (i.i.d. noise) was applied, the intra-scene similarity collapsed to a level indistinguishable from that of inter-scene pairs, demonstrating that unstructured noise obliterates the temporal and multi-view coherence inherent to a single 4D scene. In contrast, employing our proposed progressive noise model recovered a substantial fraction of the original latent correlation, confirming that imposing a structured noise prior effectively preserves meaningful frame-to-frame relationships. The t-SNE representation in Fig. 1 also shows how our proposed noise model (indicated by *"correlated"*) preserve the correlation among the noise maps or latents. On the contrary, the correlation gets destroyed when we add i.i.d. noise to the frames of the same scene. shown by the scattered noise maps. Figs. 2 and 3 shows the impact of preserving the correlation during forward diffusion porcess. We can achive both temporal and view consistency in the edited output following our proposed progressive noise sampling method. Next, we explain how we achieve consistency through progressive noise sampling.

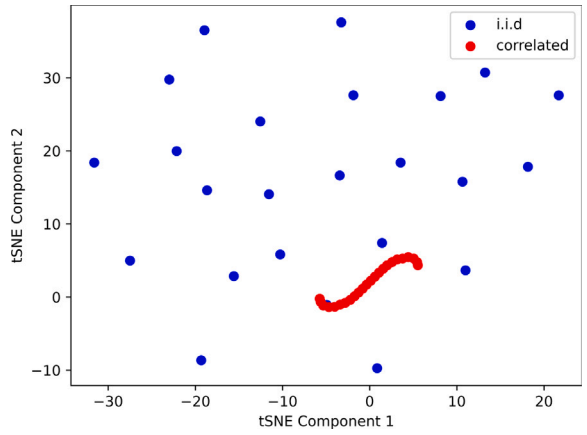


Fig. 1. tSNE visualization of noise maps of a 4D scene. To get the noise maps, we add i.i.d. noise to the input latents of the same 4D scene following the diffusion ODE process of a trained T2I model. Each blue dot represents the embeddings of K noise maps corresponding to K input views of a 4D scene. Due to the addition of i.i.d. noise, the inherent correlation between frames gets ignored by the T2V model while editing the multi-view videos. However, our progressive noise sampling framework (ANM and CNM noise models in 5.1, 5.2.1) preserves the correlation well (red dots), which gives us an edge during the reverse diffusion process.



Fig. 2. Impact of temporally consistent ANM noise. Note that we obtain such consistency only after fine-tuning IP2P following Section 5.3.

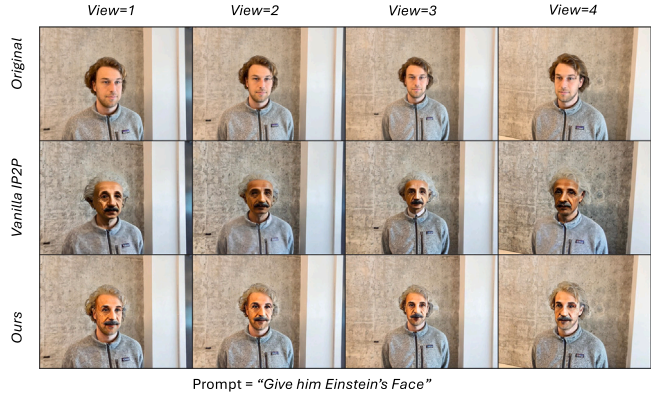


Fig. 3. Impact of view consistent CNM noise. When we preserve the noise correlation among different views, the model produce consistent edited frames. Note that we obtain such consistency only after fine-tuning IP2P following Section 5.3.

4. Preliminaries

Gaussian splatting (GS). 3D Gaussian splitting (3DGS) [31] is a technique used in computer graphics and vision to represent complex 3D scenes by positioning Gaussian functions in space to approximate surfaces and volumes. In this method, each point in a 3D scene is represented by a Gaussian ellipsoid, defined by its center position $\mathbf{p} \in \mathbb{R}^3$, covariance $\mathbf{\Sigma}$, and intensity I. The Gaussian at each point $\mathbf{p}$ is described by the function:

$$G(\mathbf{x}) = I \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{p})^\top \mathbf{\Sigma}^{-1}(\mathbf{x}-\mathbf{p})\right), \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^3$ represents a query point in space. By projecting these Gaussian onto a 2D image plane, a continuous, smooth approximation of the 3D scene is created, allowing for efficient rendering and visualization with realistic depth and detail. However, our work focuses on 4D Gaussian Splatting (4DGS) [6,32] model where 3DGS can be extended by incorporating per-frame dense tracking and novel view synthesis for dynamic scenes, leveraging a lightweight deformation field to represent Gaussian motions and shape variations across frames.

Text-guided 4D editing. Following Instruct4D-to-4D [7], we consider a 4D scene as a pseudo-3D representation where each pseudo-view consists of a sequence of multiple frames in a video format. This essentially means a 4D scene can be interpreted as a series of multi-view videos captured from different views. Given a text instruction C_T and a set of extrinsic camera parameters $\kappa \in \mathbb{R}^{K\times 16}$, our goal is to edit these multi-view videos from K different view angles, $I \in \mathbb{R}^{K\times F\times H\times W\times C}$. Here, we can edit each view using a T2V model and then train a 4DGS model [6] on the edited views. For the T2V model, we take a pre-trained InstructPix2Pix (IP2P) [21] image editing model and follow Tune-A-Video (TAV) [22] to inflate the 2D convolutions to 3D convolutions. Similarly to IP2P, *training* video diffusion models also consist of a diffusion process paired with a denoising process, both operating within the latent space of an autoencoder, $\mathcal{E}$. During forward diffusion, *i.i.d* noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is added to the latent $z = \mathcal{E}(I)$ to produce a noisy latent z_t, with noise level set by a random timestep $t \in T$. For reverse diffusion, we consider the DDIM sampling [33] process where we start from a latent z_T with maximum noise. The T2V model (with parameters θ) is trained to predict the clean latent for the next timestep $\tilde{z}_{t-1}$ as $\tilde{\epsilon}_t = \tilde{\epsilon}_\theta(z_t, t, \kappa, I, C_T)$. Here, the model is trained to approximate the noise added during the forward diffusion. The update rule for each timestep t is defined as:

$$\tilde{z}_{t-1} = \sqrt{\alpha_{t-1}}\left(\frac{z_t - \sqrt{1-\alpha_t}\,\tilde{\epsilon}_t}{\sqrt{\alpha_t}}\right) + \sqrt{1-\alpha_{t-1}}\,\tilde{\epsilon}_t, \tag{2}$$

where α_t and α_{t-1} control the noise level at each step. Please see *Supplementary* for more details on 4DGS and the diffusion process.

5. Methodology

In Fig. 4, we present our proposed framework, PSF-4D, where we achieve text-driven 4D editing using a 2D image diffusion model. Since a 4D scene consists of multi-view video data, we begin with the adaptation of a T2I model to a T2V model that can perform multi-view video editing. We aim to fine-tune the T2V model on the multi-view videos to capture the motion dynamics and 3D geometry information that are fundamental in 4D scene editing. However, there are several challenges associated with vanilla fine-tuning due to the lack of 4D-scene-specific awareness of the T2V model. *temporal consistency* and *multi-view consistency*. In general, i.i.d. noise is used during forward diffusion, which does not account for the correlation between video frames across different views. To overcome these challenges, we propose a *progressive noise sampling* strategy that consists of two noise initialization models: *auto-regressive noise model (ANM)* to enforce temporal consistency and *cross-view noise model (CNM)* to enforce multi-view consistency. We

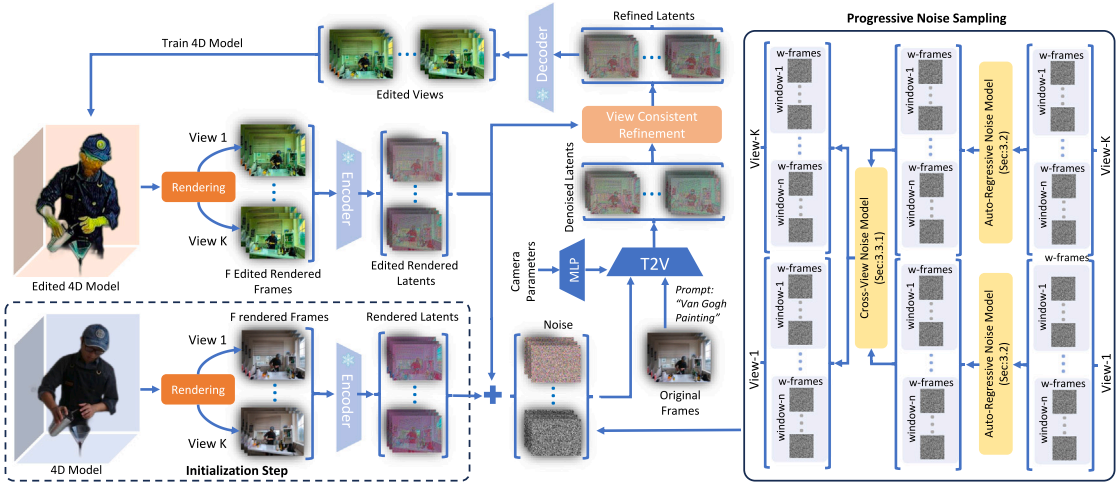


Fig. 4. PSF-4D framework is designed for text-guided 4D editing. **Right.** We introduce a progressive noise sampling method for noise initialization, consisting of two key stages: **(i)** an *autoregressive noise model* to ensure temporal consistency and **(ii)** *cross-view noise control* to maintain spatial coherence. **Left.** This technique is incorporated into the diffusion process of the text-to-video (T2V) editing model, enabling the generation of 4D scenes with spatio-temporal coherence across multiple views. We further refine the edited 4D scene by enforcing a *view-consistent refinement* strategy. Note that we consider this refinement process only after constructing the initial edited 4D model, i.e. $l >= 1$ (Section 5.2). After the initialization step ($l = 0$), we do not consider the original rendered latents anymore; only edited rendered latents have a role in next stages ($l >= 1$).

take additional measures for preserving multi-view geometry information during editing: **(i)** a *view-consistent refinement (VCR)* technique that iteratively refines the edits obtained from the T2V model; **(ii)** *view-aware positional encoding (VPE)* to distinguish between different views.

5.1. Temporal consistency

In case of *i.i.d*, noise across frames is drawn independently from a Gaussian distribution $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\mathbf{I}$ represents the identity covariance matrix. Since this independent frame-wise noise model does not consider cross-frame correlations, the generated video may contain inconsistent or jittery frame transitions. Therefore, we replace this independent noise assumption with a correlated noise sequence generated by an autoregressive (AR) model. To this end, we take a window-based approach where we have n number of windows with each having w frames, i.e. $F = nw$. Let the noise tensor $\epsilon = (\epsilon^1, \epsilon^2, \dots, \epsilon^n)^\top$ represent the noise values across n windows. We define the AR(1) model as follows:

$$\epsilon^i = \gamma \epsilon^{i-1} + \sqrt{1-\gamma^2}\,\boldsymbol{\eta}_i, \quad \boldsymbol{\eta}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{3}$$

where $\gamma \in (0, 1)$ is a parameter controlling the degree of temporal correlation between consecutive windows, and $\boldsymbol{\eta}_i$ represents independent Gaussian noise at each window.

Algorithm 1 summarizes the proposed framework. The inputs include the original 4DGS model, target text prompt, T2V denoising model, VAE encoder/decoder, camera parameters, refinement steps, noise hyperparameters, and refinement weights.

5.2. Multi-view consistency

After performing temporally consistent editing in all views, we can train a 4D model on the given edited views. However, simply regenerating these edits again and again still produces inconsistent results due to the issue of multiview consistency. Hence, we propose to enforce multi-view consistency in the final editing through: *Cross-view Noise Model (CNM)*, *View Consistent Refinement (VCR)*, and *View-Aware Position Embedding (VPE)*.

5.2.1. Cross-view noise model

Although the auto-regressive noise model is better suited for temporal consistency, spatial coherence across different perspectives is more important in multi-view generation. Therefore, we consider a slightly different noise model in this case. Considering K views of a 4D scene, where each view is a video of n windows. Before considering the cross-view noise model, we first apply the auto-regressive noise model to all K views $\boldsymbol{\epsilon} = \{\epsilon^i_k\}$ where $i \in [1, n]$ and $k \in [1, K]$. Here, ϵ^i_k represents the noise value for ith window of the kth view.

For better understanding, we present a window-by-window noise model as the T2V framework processes one window of a specific view at a time. For ith window of all K views, let $\hat{\epsilon}^i = (\hat{\epsilon}^i_1, \hat{\epsilon}^i_2, \dots, \hat{\epsilon}^i_K)^\top$ denote the tensor comprising noise components for individual views. Here, $\hat{\epsilon}^i_k$ corresponds to the kth element in the noise tensor $\hat{\epsilon}^i$. We introduce a shared noise component $\hat{\epsilon}^i_{\text{shared}}$ that is constant across all views, establishing a baseline level of similarity among the generated views. This is crucial in multi-view generation, where maintaining coherence across different perspectives of the same scene is essential for realistic rendering. On the other hand, another component $\hat{\epsilon}^i_{k,\text{ind}}$ is also considered, which provides the individual noise for each view. Adding $\hat{\epsilon}^i_{k,\text{ind}}$ allows for controlled variation across views, capturing slight differences that would naturally occur when observing a 3D object or scene from multiple angles. This helps avoid rigid or overly uniform results that can occur when only shared noise is applied. The final noise for each view $\hat{\epsilon}^k_i$ is then constructed as a linear combination of these two components.

$$\begin{aligned} &\hat{\epsilon}^i_{\text{shared}} \sim \mathcal{N}(\mathbf{0}, \lambda\mathbf{I}), \quad \hat{\epsilon}^i_{k,\text{ind}} = \sqrt{1-\lambda}\,\epsilon^i_k, \\ &\hat{\epsilon}^i_k = \hat{\epsilon}^i_{\text{shared}} + \hat{\epsilon}^i_{k,\text{ind}} \end{aligned} \tag{4}$$

Here, $\lambda \in (0, 1)$ controls the balance between shared and individual noise contributions. This design introduces cross-view correlation through $\hat{\epsilon}^i_{\text{shared}}$, while $\hat{\epsilon}^i_{k,\text{ind}}$ retains unique noise characteristics for each view, creating both shared and individual noise features to enhance coherence and diversity in multi-view generation. Finally, we have the noise values for all windows and all views, $\hat{\epsilon} = \{\hat{\epsilon}^i_k\}$ where $i \in [1, n]$ and $k \in [1, K]$.

Fig. 4 illustrates the process of generating the initial set of edited views. Starting from the unedited 4D model, we render multiple views, which are passed through the VAE encoder to obtain their corresponding unedited latent representations. Using $\hat{\epsilon}$, we introduce noise to these latents, preparing them for processing through the Text-to-Video (T2V) model, conditioned on both text prompts and the original view information. After denoising, the latents are decoded through the VAE

Algorithm 1: PSF-4D: Progressive Sampling Framework for 4D Editing

Input: $M_{\text{orig}}, C_T, \epsilon_\theta, \mathcal{E}, \mathcal{D}, \kappa, L, \gamma, \lambda, \{\omega_l\}_{l=0}^{L-1}$
Output: M_{final}
Function `NoiseModel`(γ, λ, K, n):
  `// ANM: temporal noise`
  for $k = 1$ **to** K **do**
    Initialize $\epsilon_k^1 \sim \mathcal{N}(0, I)$
    for $i = 2$ **to** n **do**
      $\eta_i \sim \mathcal{N}(0, I)$
      $\epsilon_k^i \leftarrow \gamma \epsilon_k^{i-1} + \sqrt{1-\gamma^2}\eta_i$
    end
  end
  `// CNM: cross-view noise`
  for $i = 1$ **to** n **do**
    $\hat{\epsilon}_{\text{shared}}^i \sim \mathcal{N}(0, \lambda I)$
    for $k = 1$ **to** K **do**
      $\hat{\epsilon}_{k,\text{ind}}^i \leftarrow \sqrt{1-\lambda}\epsilon_k^i$
      $\hat{\epsilon}_k^i \leftarrow \hat{\epsilon}_{\text{shared}}^i + \hat{\epsilon}_{k,\text{ind}}^i$
    end
  end
  return $\{\hat{\epsilon}_k^i\}$
`// Initial 4D editing`
Render $\{I_{\text{orig}}^k\}_{k=1}^K$ from M_{orig}
Encode $\{z_{\text{orig}}^k\}_{k=1}^K \leftarrow \mathcal{E}(\{I_{\text{orig}}^k\})$
$\{\hat{\epsilon}^k\} \leftarrow$ `NoiseModel`(γ, λ, K, n)
Apply forward diffusion to obtain $\{z_T^k\}$ from $\{z_{\text{orig}}^k\}$ and $\{\hat{\epsilon}^k\}$
Denoise $\{z_T^k\}$ with ϵ_θ and prompt C_T to obtain $\{\tilde{z}_0^k\}_{k=1}^K$
Decode $\{\tilde{I}_0^k\}_{k=1}^K \leftarrow \mathcal{D}(\{\tilde{z}_0^k\})$
Train initial 4DGS model M_0 on $\{\tilde{I}_0^k\}$
`// View-consistent refinement`
for $l = 0$ **to** $L-1$ **do**
  Render $\{\tilde{I}_l^k\}_{k=1}^K$ from M_l
  Encode $\{\tilde{z}_l^k\}_{k=1}^K \leftarrow \mathcal{E}(\{\tilde{I}_l^k\})$
  $\{\hat{\epsilon}^k\} \leftarrow$ `NoiseModel`(γ, λ, K, n)
  Apply forward diffusion to obtain $\{z_T^k\}$ from $\{\tilde{z}_l^k\}$
  Denoise with ϵ_θ to obtain $\{\tilde{z}_{l+1}^k\}_{k=1}^K$
  $\hat{z}_{l+1}^k \leftarrow \omega_l \tilde{z}_{l+1}^k + (1-\omega_l)\tilde{z}_l^k, \ \forall k$
  Decode $\{\tilde{I}_{l+1}^k\}_{k=1}^K \leftarrow \mathcal{D}(\{\hat{z}_{l+1}^k\})$
  Train M_{l+1} on $\{\tilde{I}_{l+1}^k\}$
end
return M_L

decoder, resulting in the initial edited views $\tilde{I}_0 = \{\tilde{I}_0^k\}, k \in [1, K]$. By training on this initial set $\tilde{I}_0$, we obtain the initial edited 4D model.

5.2.2. View consistent refinement

Let us denote $\tilde{I}_l^k$ as kth renderings of the edited 4D model at the lth iteration of the refining process. We then obtain $\tilde{z}_l^k$ as the latent equivalent of $\tilde{I}_l^k$. After adding noise to the $\tilde{z}_l^k$, we use the T2V model with conditioning for denoising. Following Eq. (2), the denoised latent $\tilde{z}_{l+1}^k$ can be estimated after T number of DDIM sampling steps. If we decode $\tilde{z}_{l+1}^k$ to $\tilde{I}_{l+1}^k$, we should have a higher quality view generation with smooth motion as compared to $\tilde{I}_l^k$. However, the same cannot be said for spatial consistency among views as the diffusion process of T2V struggles with view-consistent generation (even with the utilization of CNM). Therefore, we explicitly inject view information into the T2V editing pipeline. To this end, the rectified $\hat{z}_{l+1}^k$ are computed by

$$\hat{z}_{l+1}^k = \omega_l \tilde{z}_{l+1}^k + (1-\omega_l)\tilde{z}_l^k, \tag{5}$$

Here, ω_l is a predefined weight to balance between the denoising results $\tilde{z}_{l+1}^k$ and the rendered multi-view consistent $\tilde{z}_l^k$. The parameter ω_l determines how much multi-view consistency is imposed on the denoising process. Training a 4D model with more focus on $\tilde{z}_l^k$ (low ω_l) forces multi-view consistency but may oversmooth some regions. On the other hand, directly utilizing the denoising directions from $\tilde{z}_{l+1}^k$ ($\omega_l = 1$) produces videos with more details but less multi-view consistency. At the beginning of the refinement stage, we focus more on the fidelity of the generated views while emphasizing more on the multi-view consistency at later iterations. Therefore, we start with a high value of ω_l and slowly decrease it as the refinement progresses. We repeat the refinement process for L steps.

5.2.3. View-aware position encoding

In our work, we fine-tune a T2V model with multi-view video data. To distinguish between different views while fine-tuning, view-aware position encoding is necessary, which can be derived from the absolute camera parameters. To this end, we encode the camera parameters κ by employing a 2-layer MLP with parameters Φ and add the resulting camera embeddings to *time embeddings* as residuals [11]. Doing so provides additional view awareness to the T2V model and reduces spatial artifacts. We summarize our proposed method in Algorithm 1.

5.3. Fine-tuning objective

For scene-specific adaptation of the T2V model, we minimize the following multi-view diffusion loss,

$$\mathcal{L}(\theta, \Phi) = \mathrm{E}_{I, C_T, \kappa, t, \hat{\epsilon}}\left[\|\hat{\epsilon} - \tilde{\epsilon}_\theta(z_t, t, \kappa, I, C_T)\|_2^2.\right] \tag{6}$$

In our work, we tune the model for around 3000 iterations on each scene before utilizing it for text-guided editing.

6. Experiments

Our implementation is built on the PyTorch framework and tested on a single RTX A6000 GPU, utilizing the 4D Gaussian Splatting framework [6] for constructing 4D scenes. The model initialization phase involves 12,000 training iterations, 10,000 iterations in the coarse phase to optimize the static 3D Gaussian, and 2000 iterations in the fine phase to refine the 4D Gaussian. For example, in the HyperNeRF dataset [34], we use a rendering resolution of 960 × 540, achieving a rendering speed of 34 FPS. The editing phase adds 8000 more iterations. Considering the fine-tuning of the T2V model, PSF-4D has a total training time of approximately 2 h. In addition, we incorporate SAM [35] to achieve local and precise editing.

6.0.1. Datasets

We evaluate our method on 4D scenes captured using both single hand-held cameras and multi-camera setups. These include: (I) *Monocular* scenes from the DyCheck [37] and HyperNeRF [36] datasets, featuring object-centric scenes with single moving cameras, and (II) *Multi-camera* scenes from DyNeRF/N3DV [38], consisting of indoor environments with human motions and multiple views.

6.0.2. Baselines

We compare our approach with the baseline I4D-to-4D [7] and an extended version of the 3D editing framework, IN2N [39], which we adapted into a 4D variant (IN2N+4D) by iteratively editing each frame and incorporating it back into the dataset. Although Control4D [8] is a similar 4D editing work, we could not report the comparisons here as no implementation was available. For I4D-to-4D, we follow the implementation here[2] where the NeRF-based 4D editing was performed which usually takes a higher number of iterations (around 100K). We follow the implementations here[3] for InstructNeRF2NeRF and this codebase[4] for Hexplane. We iterative update all the frames of a scene sequentially, without employing any spatiotemporal attention of T2V.

[2] https://github.com/Friedrich-M/Instruct-4D-to-4D/
[3] https://github.com/ayaanzhaque/instruct-nerf2nerf
[4] https://github.com/Caoang327/HexPlane



Fig. 5. Qualitative 4D Editing Results. Examples of 4D editing tasks performed using PSF-4D. Each row represents a specific editing scenario, demonstrating the versatility and precision of PSF-4D across a variety of tasks, including style transfer, object transformation, and attribute modification. From transforming a scene into different artistic styles to specific object edits, PSF-4D maintains consistency and coherence across frames in dynamic 4D scenes.

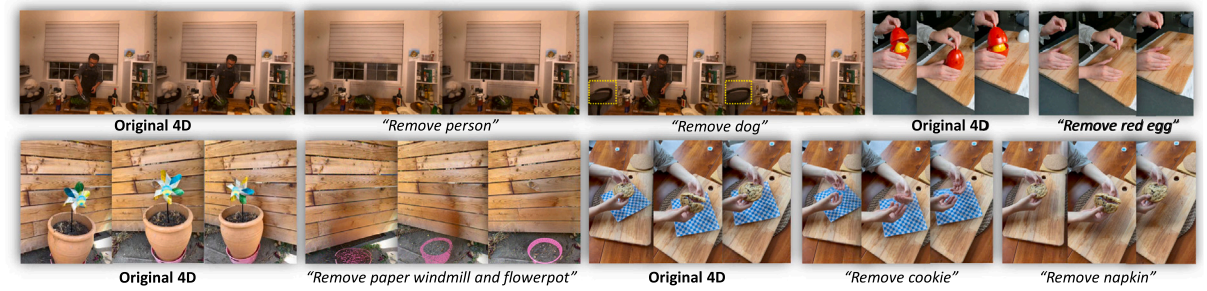


Fig. 6. Object Removal in 4D Scenes. Examples of object removal across various scenes from different datasets, including DyNeRF, DyCheck, and HyperNeRF. Each row illustrates the original 4D scene followed by frames with specific objects removed, as per the editing prompt.



Fig. 7. Comparison of 4D Editing Results. Examples of 4D scene editing using our approach compared to Instruct 4D-to-4D (I4Dto4D) and the original 4D scene across various scenes in the DyNeRF dataset. Our approach (bottom row) achieves more precise and consistent adherence to the editing prompts across all frames, producing visually coherent and superior editing effects.

Table 2
Per-Scene results of HyperNeRF's vrig datasets [36] by different models.

Method	3D Printer		Chicken		Broom		Banana	
	PSNR↑	MS-SSIM↑	PSNR↑	MS-SSIM↑	PSNR↑	MS-SSIM↑	PSNR↑	MS-SSIM↑
IN2N+HexPlane	14.16	0.627	16.84	0.654	16.30	0.647	16.96	0.670
I4D-to-4D	17.93	0.713	20.62	0.742	20.17	0.739	19.88	0.738
PSF-4D (Ours)	20.16	0.753	23.83	0.815	19.62	0.748	22.31	0.808

Table 3
Quantitative Comparison across 100 dynamic scene edits.

Method	FVD ↓	FID ↓	CLIP-S ↑	PSNR ↑	SSIM ↑
IN2N + HexPlane	382.6	68.75	0.2985	17.28	0.652
I4D-to-4D	294.1	37.58	0.3045	19.74	0.697
PSF-4D (Ours)	**215.3**	**25.39**	**0.3292**	**21.85**	**0.728**

Table 4
Ablation Study on different components of PSF-4D. We consider the DyNeRF dataset with 8 different prompts for this experiment.

Method	FVD ↓	FID ↓	CLIP-S ↑	PSNR ↑	SSIM↑	LPIPS↓
IN2N + HexPlane	379.2	64.27	0.2971	16.71	0.649	0.451
I4D-to-4D	281.5	34.52	0.3068	19.92	0.706	0.419
PSF-4D *w/o VCR*	292.8	35.06	0.2924	17.84	0.662	0.427
PSF-4D *w/o CNM*	253.7	27.17	0.3088	19.96	0.698	0.379
PSF-4D *w/o ANM*	229.1	26.12	0.3104	21.15	0.711	0.365
PSF-4D *w/o VPE*	218.0	24.05	0.3196	21.93	0.716	0.352
PSF-4D	**210.4**	**22.58**	**0.3241**	**22.17**	**0.726**	**0.348**

6.0.3. Evaluation metrics

For quantitative evaluation, we utilize the Fréchet Video Distance (FVD) [40] and Fréchet Inception Distance (FID) [41] metrics to assess the visual similarity between the edited dataset and generated images. Additionally, we calculate the CLIP cosine similarity (CLIP-S) [42] to measure alignment between generated images and textual descriptions, thereby providing a robust evaluation of both visual fidelity and semantic relevance in the edits. In addition to FID and CLIP-S, the evaluation of visual quality is conducted using LPIPS [43], Structural Similarity Index (SSIM) [44], Multi-scale Structural Similarity Index (MS-SSIM) [45], and PSNR metrics. However, due to the reconstruction of dynamic scenes from a single moving camera, slight deviations in geometry or appearance may occur, introducing ambiguities. These discrepancies often cause standard quantitative metrics to fail in capturing perceptual quality accurately. For instance, PSNR is highly sensitive to minor shifts, often favoring blurred outputs over sharp ones, while MS-SSIM may overlook artifacts that are noticeable to human observers. Another metric we employ is LPIPS, which measures perceptual distance to the original (pre-edit) scene. In an editing context, a more prompt-faithful edit necessarily deviates more from the original, which can paradoxically increase LPIPS even when edit quality improves.

6.1. Qualitative evaluation

Our PSF-4D framework demonstrates robust 4D editing capabilities across four key tasks:

6.1.1. Multi-attribute editing

In multi-attribute editing, PSF-4D effectively manages multiple modifications on a single subject. For example, in Fig. 5, the prompt "Give him blue trousers and a brown leather bag" requires simultaneous color and object modifications. PSF-4D successfully applies both edits consistently across frames, demonstrating its ability to handle compound changes without compromising coherence.

6.1.2. Style transfer

PSF-4D excels in style transfer tasks, as seen in Fig. 5 with prompts such as 'Make it a Van Gogh painting" and "Turn him into silver statue". PSF-4D accurately applies the specified artistic styles across the entire scene, maintaining consistency in both spatial and temporal dimensions. The stylistic transformations are visually coherent, reflecting the superior control of PSF-4D in the aesthetic changes across the scene. Table 3 demonstrate the effectiveness of our proposed method.

6.1.3. Object removal

Fig. 6 highlights PSF-4D's capability in object removal tasks. Prompts such as "Remove person", "Remove red egg", and "Remove napkin" illustrate how PSF-4D seamlessly removes targeted objects from complex scenes while preserving background integrity and spatial consistency. This precise control in object manipulation highlights PSF-4D's advanced scene understanding in dynamic 4D environments. We use the inpainting version of the Stable Diffusion-2.1 model here.

6.1.4. Local editing

Fig. 7 demonstrates the local editing capabilities of our proposed approach. In addition to SAM, we modify the "target editing prompt", C_T with GPT-4 [46] and LlaVA [47]. For example, if we feed the "Coffee Martini" video from Dy-NeRF dataset to LLaVA, the description looks like this, C_D: *"The video features a person standing behind a white countertop in what appears to be a home bar setup. The individual is wearing a dark apron and a cap, suggesting they are preparing a drink. On the countertop, there is an array of items including a juicer with orange juice, a cocktail shaker, a glass filled with ice, and several bottles of alcohol. The background shows a window with blinds partially drawn, allowing natural light into the room. The person is seen pouring liquid from the cocktail shaker into the glass, indicating the process of making a cocktail".* Consider the target prompt to be, C_T: "Turn him into an Anime Hero". Now we feed this description and target caption to GPT-4, we get the final editing prompt, C_S: *"Transform the person into an anime hero with spiky hair, a vibrant outfit, and a glowing aura, keeping the home bar setup and cocktail-making intact".* C_S has been used to edit the final 4D model.

Qualitative results in different editing tasks emphasize PSF-4D's adaptability and precision in 4D scene editing, consistently outperforming baseline methods by producing high-quality, contextually aligned modifications. Note that SAM is used only for local/segmentation-guided editing (Section 6.1.4), GPT-4 and LLaVA are used only for prompt enhancement in local editing experiments, and Stable Diffusion 2.1 inpainting is used only for object removal (Section 6.1.3). These are optional task-specific modules, not part of the core PSF-4D consistency framework evaluated in Tables 3 and 4. The ablation in Table 4 isolates the core components (ANM, CNM, VCR, VPE) without these external modules.

6.2. Quantitative evaluation

The quantitative results in PSF-4D framework compared to existing methods on 100 dynamic scene edits. We evaluate across 100 dynamic scene edits, comprising around 14 scenes and 8 prompts per scene across the DyNeRF, HyperNeRF, and DyCheck datasets. Specifically, 48 edits are from DyNeRF (6 scenes × 8 prompts), 32 from HyperNeRF (4 scenes × 8 prompts), and 20 from DyCheck. We evaluate the methods using Frechet Inception Distance (FID) to measure the quality of generated images and CLIP Similarity to assess alignment with the textual prompts. Our PSF-4D approach achieves the lowest FID score of 25.39, indicating superior visual quality and coherence in the edited outputs compared to IN2N+HexPlane [4] and I4D-to-4D [7], which have FID scores of 68.75 and 37.58, respectively. Furthermore, PSF-4D achieves the highest CLIP similarity score of 0.3292, reflecting better alignment with the intended editing prompts than the other methods. These results highlight PSF-4D's ability to produce contextually accurate and visually realistic edits, outperforming baseline methods in both perceptual quality and prompt adherence. To validate our results, a Wilcoxon signed-rank test was performed on the per-scene DyNeRF data, confirming that PSF-4D significantly outperforms the baseline on PSNR, SSIM, CLIP-S, FVD, FID metrics ($p < 0.05$).

In Table 5, we present the performance of PSF-4D on each dataset. Due to the proposed noise model design and iterative refinement, PSF-4D achieves superior results in all datasets, showing the impact of our proposed design. These improvements indicate that controlled noise in the diffusion model enforces generation with greater temporal and view

Table 5
Quantitative results on all datasets. For HyperNeRF, rendering resolutions are set to 960 × 540 and 1352 × 1014 for DyNeRF. On the other hand, we use a resolution of 480p for Dy-Check datasets. In total, 8 different prompts for each dataset is used in this experiment.

Dataset	HyperNeRF [36]			DyNeRF [38]			DyCheck [37]		
Model	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑↑	SSIM↑	LPIPS↓
IN2N+HexPlane	16.04	0.651	0.343	16.71	0.649	0.374	11.80	0.469	0.431
I4D-to-4D	19.65	0.736	0.371	19.92	0.706	0.419	14.10	0.516	0.367
PSF-4D (Ours)	**21.48**	**0.781**	**0.352**	**22.17**	**0.726**	**0.417**	**15.28**	**0.531**	**0.332**

Table 6
Per-Scene results of DyNeRF's [38] datasets.

Method	Cut Beef		Cook Spinach		Sear Steak	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
IN2N+HexPlane	19.42	0.674	18.94	0.668	18.02	0.661
I4D-to-4D	21.40	0.727	20.85	0.728	20.51	0.716
Ours	23.47	0.743	23.16	0.732	22.83	0.729
Method	**Flame Steak**		**Flame Salmon**		**Coffee Martini**	
	PSNR↑	**SSIM↑**	**PSNR↑**	**SSIM↑**	**PSNR↑**	**SSIM↑**
IN2N+HexPlane	17.31	0.656	16.25	0.649	15.56	0.640
I4D-to-4D	19.68	0.704	18.92	0.696	18.16	0.683
Ours	22.31	0.724	21.10	0.719	20.15	0.709

Table 7
Ablation Study on number of refinement steps, L. Although the longer refining phase produces slightly better results, the computational burden for performing these extra refining steps. Considering both performance and computation, we choose $L = 2$.

Refinement Steps, L	FID ↓	CLIP Similarity ↑	PSNR↑	SSIM↑	LPIPS↓
1 ($\omega_1 = 0.9$)	23.37	0.3194	21.98	0.717	0.465
2 ($\omega_3 = 0.6$)	22.58	0.3241	22.17	0.726	0.348
3 ($\omega_5 = 0.5$)	**22.21**	**0.3275**	**22.48**	**0.731**	**0.342**

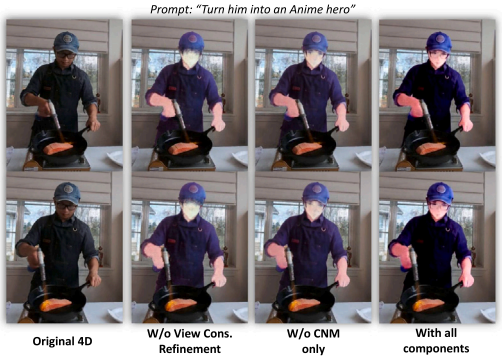


Fig. 8. Ablation on different components of our proposed method. We show the impact of CNM and view consistent refinement in obtaining the desired editing effect. PSF-4D w/o view consistent refinement (VCR) indicates we edit the model only once ($L = 0$), resulting in poor editing. Without CNM, the initial edited 4D model's quality drops significantly. However, quality improvement can be observed with iterative VCR. These results suggested that VCR plays the most important role in achieving the SOTA 4D editing performance.

consistency. Tables 6–2 shows per-scene statistics for Dy-NeRF and HyperNeRF datasets. It can be observed that some scenes are relatively more complex than others. For the HyperNeRF dataset, we consider the MS-SSIM metric as suggested in the original work (See Table 5).

7. Ablation study

We study the impact of different hyperparameters on the performance of PSF-4D. Fig. 8 and Table 4 show the impact of different components on the overall performance of PSF-4D. It can be observed that all of these components play important roles in obtaining our desired results. For choosing the values for different hyperparameters, we also conduct further ablation. For instance, we choose $\lambda = 0.7$ and $\gamma = 0.65$ to obtain the best editing performance. On the other hand, we start with $\omega_1 = 0.9$ and decrease it to $\omega_L = 0.6$ at the end of the refinement stage. The detailed breakdown of them is as follows.

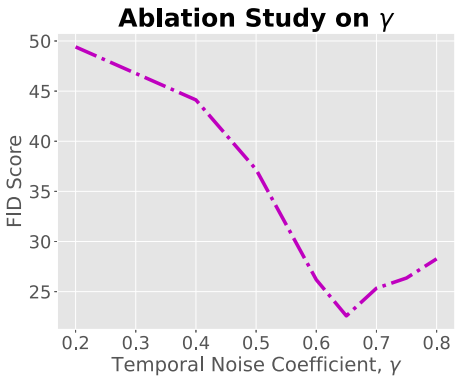


Fig. 9. Ablation Study on temporal noise coefficient, γ in ANM noise model. The value of γ determines how much weight T2V model puts on the previous window of frames while generating the current window of frames. The optimal value is $\gamma = 0.65$. Choosing a higher value puts a tighter constraint on the generation which eventually leads to suboptimal performance. We calculate the fidelity over all generated frames of DyNeRF datasets. A total of 8 distinct prompts are utilized for generating edits.

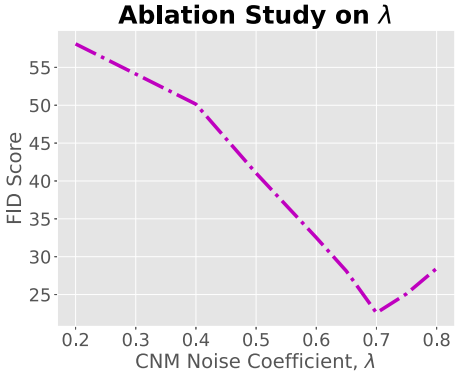


Fig. 10. Ablation Study on cross-view noise coefficient, λ in CNM noise model. A lower value of λ results in less consistent editing across different views due to reduced correlation between intermediate latents in T2V. Conversely, selecting a value greater than 0.7 makes it challenging to capture nuances in view-specific changes. This experiment is conducted using the DyNeRF dataset.

7.1. Ablation on γ

Fig. 9 shows the FID scores for different values of γ. As shown in the main paper, γ determines the nature of noise initialization in the T2V

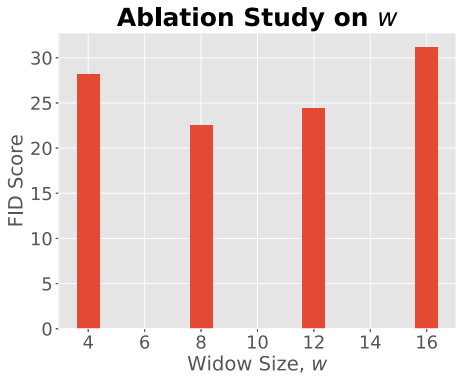


Fig. 11. Ablation Study on the video window size, w. Here, we use $w = 8$ number of frames in each window. Given a fixed number of frames in a dataset, temporal consistency at later frames is hard to achieve if w is small (the number of windows n is low). Due to the current limitations of T2V models, intra-window consistency deteriorates once we increase the window size. We consider Dy-NeRF dataset for this.

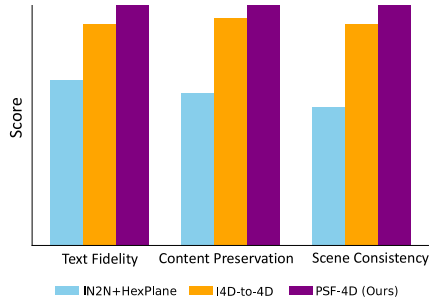


Fig. 12. User Study Evaluation. Results of a user study comparing text fidelity, content preservation, and scene consistency.

model and helps us obtain better temporal consistency. For all these ablations, we consider the Dy-NeRF dataset.

7.2. Ablation on λ

In Fig. 10, we present the ablation on λ. If we choose a smaller value, the model struggles to preserve the commonalities between views.

7.3. Ablation on window size, w

Fig. 11 illustrates the performance variation when we change the window size of a video.

7.4. Ablation on refinement steps, L

Table 7 shows the impact of the view-consistent refinement process over time. If we refine the edited 4D model for longer, the computation burden will be significant, as it takes around 15 min for each refinement step. We also linearly reduce the value of ω_l as the refinement progresses.

7.5. User study

For the user study shown in Fig. 12, we surveyed a random sample of 30 participants between the ages of 21 and 40. Participants were asked to rate the generated edits based on three key aspects: text fidelity, content preservation, and scene consistency. Scores were then averaged for each method, with PSF-4D consistently achieving the highest ratings across all categories, reflecting strong user preference and perceived quality.

8. Discussion and limitations

Although PSF-4D follows the Tune-a-Video framework and uses a Stable Diffusion image editing model, it is not tied to a specific editor or T2V backbone. This modular design allows it to be integrated with different off-the-shelf models for flexible 4D editing. At the same time, PSF-4D still has limitations: progressive noise modeling improves temporal and view consistency, but may be less effective in highly dynamic or complex scenes where noise-based conditioning cannot fully capture fine spatio-temporal details. View-consistent refinement helps address this issue, but excessive refinement may over-smooth details, especially for large structural edits such as full-body transformations, reducing realism in textured regions. Careful selection of ω_l and L can mitigate this effect, while future adaptive noise control could adjust γ and λ based on scene complexity.

Additional limitations could be the computational burden for per-scene editing. Potential acceleration paths include (i) replacing the full fine-tuning stage with few-step distillation methods such as Latent Consistency Models (LCMs) or diffusion distillation; (ii) adopting faster 4D-GS initialization; and (iii) reducing refinement steps with adaptive stopping criteria based on view-consistency metrics. Another area of concern can be the edits requiring large structural changes (e.g., full-body transformation like 'Turn him into monkey') can produce artifacts because the correlated noise model alone cannot guarantee semantic faithfulness under radical appearance change. Note that these artifacts may partly arise from multi-view inconsistency in the T2V model when edits deviate substantially from the original scene geometry.

9. Conclusion

We developed a novel 4D editing framework based on a text-to-image generation model and focusing on the fundamental aspects of the diffusion process. PSF-4D offers an effective framework for achieving consistent and high-quality 4D video editing. By combining progressive noise sampling with iterative refinement, PSF-4D addresses key challenges in maintaining both temporal and view coherence across frames and perspectives. Leveraging autoregressive noise initialization and a cross-view noise model, the framework captures temporal dependencies and spatial alignment, while view-consistent iterative refinement ensures precise and stable edits. PSF-4D demonstrates a robust approach for complex 4D editing tasks, laying the groundwork for future improvements in efficiency and adaptability for dynamic scene editing across various applications. Diverse editing capabilities in multiple benchmarks have demonstrated the merit of our proposed framework.

CRediT authorship contribution statement

Hasan Iqbal: Conceptualization, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Nazmul Karim:** Writing – original draft, Writing – review & editing. **Azib Farooq:** Writing – original draft, Writing – review & editing. **Umar Khalid:** Writing – review & editing. **Zichun Zhang:** Writing – review & editing. **Jing Hua:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by National Science Foundation (OAC-1910469 and OAC-2311245).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cag.2026.104645.

Data availability

Data will be made available on request.

References

[1] Fridovich-Keil S, Meanti G, Warburg FR, Recht B, Kanazawa A. K-Planes: Explicit radiance fields in space, time, and appearance. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, p. 12479–88.

[2] Liu JW, Cao YP, Mao W, Zhang W, Zhang DJ, Keppo J, Shan Y, Qie X, Shou MZ. Devrf: Fast deformable voxel radiance fields for dynamic scenes. Adv Neural Inf Process Syst 2022;35:36762–75.

[3] Fridovich-Keil S, Meanti G, Warburg FR, Recht B, Kanazawa A. K-planes: Explicit radiance fields in space, time, and appearance. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023, p. 12479–88.

[4] Cao A, Johnson J. Hexplane: A fast representation for dynamic scenes. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023, p. 130–41.

[5] Luiten J, Kopanas G, Leibe B, Ramanan D. Dynamic 3d Gaussians: Tracking by persistent dynamic view synthesis. In: 2024 international conference on 3D vision. IEEE; 2024, p. 800–9.

[6] Wu G, Yi T, Fang J, Xie L, Zhang X, Wei W, Liu W, Tian Q, Wang X. 4D gaussian splatting for real-time dynamic scene rendering. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2024, p. 20310–20.

[7] Mou L, Chen JK, Wang YX. Instruct 4D-to-4D: Editing 4D scenes as pseudo-3D scenes using 2D diffusion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2024, p. 20176–85.

[8] Shao R, Sun J, Peng C, Zheng Z, Zhou B, Zhang H, Liu Y. Control4D: Dynamic portrait editing by learning 4D GAN from 2D diffusion-based editor. 2023, arXiv preprint arXiv:2305.20082.

[9] Teed Z, Deng J. Raft: Recurrent all-pairs field transforms for optical flow. In: Computer vision–ECCV 2020: 16th European conference, glasgow, UK, August 23–28, 2020, proceedings, part II 16. Springer; 2020, p. 402–19.

[10] Lee Sg, Kim H, Shin C, Tan X, Liu C, Meng Q, Qin T, Chen W, Yoon S, Liu TY. Priorgrad: Improving conditional denoising diffusion models with data-dependent adaptive prior. 2021, arXiv preprint arXiv:2106.06406.

[11] Shi Y, Wang P, Ye J, Long M, Li K, Yang X. Mvdream: Multi-view diffusion for 3d generation. 2023, arXiv preprint arXiv:2308.16512.

[12] Ge S, Nah S, Liu G, Poon T, Tao A, Catanzaro B, Jacobs D, Huang JB, Liu MY, Balaji Y. Preserve your own correlation: A noise prior for video diffusion models. In: Proceedings of the IEEE/CVF international conference on computer vision. 2023, p. 22930–41.

[13] Luo Z, Chen D, Zhang Y, Huang Y, Wang L, Shen Y, Zhao D, Zhou J, Tan T. Videofusion: Decomposed diffusion models for high-quality video generation. 2023, arXiv preprint arXiv:2303.08320.

[14] Li R, Chen L, Zhang Z, Jampani V, Patel VM, Zhang L. SyncNoise: Geometrically consistent noise prediction for text-based 3D scene editing. 2024, arXiv preprint arXiv:2406.17396.

[15] Ruiz N, Li Y, Jampani V, Pritch Y, Rubinstein M, Aberman K. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023, p. 22500–10.

[16] Couairon G, Verbeek J, Schwenk H, Cord M. DiffEdit: Diffusion-based semantic image editing with mask guidance. In: The eleventh international conference on learning representations. 2023, URL: https://openreview.net/forum?id=3lge0p5o-M-.

[17] Tumanyan N, Geyer M, Bagon S, Dekel T. Plug-and-play diffusion features for text-driven image-to-image translation. 2022, arXiv preprint arXiv:2211.12572.

[18] Hertz A, Mokady R, Tenenbaum J, Aberman K, Pritch Y, Cohen-Or D. Prompt-to-prompt image editing with cross attention control. 2022, arXiv preprint arXiv:2208.01626.

[19] Meng C, Song Y, Song J, Wu J, Zhu JY, Ermon S. Sdedit: Image synthesis and editing with stochastic differential equations. 2021, arXiv preprint arXiv:2108.01073.

[20] Parmar G, Singh KK, Zhang R, Li Y, Lu J, Zhu JY. Zero-shot image-to-image translation. 2023, arXiv preprint arXiv:2302.03027.

[21] Brooks T, Holynski A, Efros AA. Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023, p. 18392–402.

[22] Wu JZ, Ge Y, Wang X, Lei SW, Gu Y, Shi Y, Hsu W, Shan Y, Qie X, Shou MZ. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In: Proceedings of the IEEE/CVF international conference on computer vision. 2023, p. 7623–33.

[23] Singer U, Polyak A, Hayes T, Yin X, An J, Zhang S, Hu Q, Yang H, Ashual O, Gafni O, Parikh D, Gupta S, Taigman Y. Make-a-video: Text-to-video generation without text-video data. In: The eleventh international conference on learning representations. 2023, URL: https://openreview.net/forum?id=nJfylDvgzlq.

[24] Zhou D, Wang W, Yan H, Lv W, Zhu Y, Feng J. MagicVideo: Efficient video generation with latent diffusion models. 2022, arXiv preprint arXiv:2211.11018.

[25] Liu S, Zhang Y, Li W, Lin Z, Jia J. Video-p2p: Video editing with cross-attention control. 2023, arXiv preprint arXiv:2303.04761.

[26] Ceylan D, Huang CHP, Mitra NJ. Pix2video: Video editing using image diffusion. 2023, arXiv preprint arXiv:2303.12688.

[27] Qiao YL, Gao A, Lin M. Neuphysics: Editable neural geometry and physics from monocular videos. Adv Neural Inf Process Syst 2022;35:12841–54.

[28] Jiang D, Ke Z, Zhou X, Shi X. 4D-editor: Interactive object-level editing in dynamic neural radiance fields via 4d semantic segmentation. 2023, arXiv preprint arXiv:2310.16858.

[29] Huang X, Salaün C, Vasconcelos C, Theobalt C, Öztireli C, Singh G. Blue noise for diffusion models. 2024, arXiv preprint arXiv:2402.04930.

[30] Lu K, Cai Y, Li L, Qin D, Li G. Improve temporal consistency in diffusion models through noise correlations. 2023, Submitted to ICLR 2024, myehost=https://openreview.net/forum?id=59nCKifDtm.

[31] Kerbl B, Kopanas G, Leimkühler T, Drettakis G. 3D Gaussian splatting for real-time radiance field rendering. 2023, https://arxiv.org/abs/2308.04079.

[32] Yang Z, Gao X, Zhou W, Jiao S, Zhang Y, Jin X. Deformable 3d Gaussians for high-fidelity monocular dynamic scene reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2024, p. 20331–41.

[33] Song J, Meng C, Ermon S. Denoising diffusion implicit models. 2020, arXiv preprint arXiv:2010.02502.

[34] Park DH, Azadi S, Liu X, Darrell T, Rohrbach A. Benchmark for compositional text-to-image synthesis. In: Thirty-fifth conference on neural information processing systems datasets and benchmarks track. 2021.

[35] Ravi N, Gabeur V, Hu YT, Hu R, Ryali C, Ma T, Khedr H, Rädle R, Rolland C, Gustafson L, et al. Sam 2: Segment anything in images and videos. 2024, arXiv preprint arXiv:2408.00714.

[36] Park K, Sinha U, Hedman P, Barron JT, Bouaziz S, Goldman DB, Martin-Brualla R, Seitz SM. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. 2021, arXiv preprint arXiv:2106.13228.

[37] Gao H, Li R, Tulsiani S, Russell B, Kanazawa A. Monocular dynamic view synthesis: A reality check. Adv Neural Inf Process Syst 2022;35:33768–80.

[38] Li T, Slavcheva M, Zollhoefer M, Green S, Lassner C, Kim C, Schmidt T, Lovegrove S, Goesele M, Newcombe R, et al. Neural 3d video synthesis from multi-view video. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022, p. 5521–31.

[39] Haque A, Tancik M, Efros AA, Holynski A, Kanazawa A. Instruct-nerf2nerf: Editing 3d scenes with instructions. In: Proceedings of the IEEE/CVF international conference on computer vision. 2023, p. 19740–50.

[40] Unterthiner T, Van Steenkiste S, Kurach K, Marinier R, Michalski M, Gelly S. Towards accurate generative models of video: A new metric & challenges. 2018, arXiv preprint arXiv:1812.01717.

[41] Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. Adv Neural Inf Process Syst 2017;30.

[42] Radford A, Kim JW, 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. PMLR; 2021, p. 8748–63.

[43] Zhang R, Isola P, Efros AA, Shechtman E, Wang O. The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 586–95.

[44] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 2004;13(4):600–12.

[45] Wang Z, Simoncelli EP, Bovik AC. Multiscale structural similarity for image quality assessment. In: The thrity-seventh asilomar conference on signals, systems & computers, 2003, vol. 2, IEEE; 2003, p. 1398–402.

[46] Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, Aleman FL, Almeida D, Altenschmidt J, Altman S, Anadkat S, et al. Gpt-4 technical report. 2023, arXiv preprint arXiv:2303.08774.

[47] Lin B, Ye Y, Zhu B, Cui J, Ning M, Jin P, Yuan L. Video-llava: Learning united visual representation by alignment before projection. 2023, arXiv preprint arXiv:2311.10122.