International Journal of Computer Vision (2026) 134:14
https://doi.org/10.1007/s11263-025-02615-z

Efficient4D: Fast Dynamic 3D Object Generation from a Single-view Video

Zijie Pan[1] · Zeyu Yang[1] · Xiatian Zhu[2] · Li Zhang[1]

Received: 22 July 2024 / Accepted: 12 November 2025

Abstract

Generating dynamic 3D object from a single-view video is challenging due to the lack of 4D labeled data. An intuitive approach is to extend previous image-to-3D pipelines by transferring off-the-shelf image generation models such as score distillation sampling. However, this approach would be slow and expensive to scale due to the need for back-propagating the information-limited supervision signals through a large pretrained model. To address this, we propose an efficient video-to-4D object generation framework called **Efficient4D**. It generates high-quality spacetime-consistent images under different camera views, and then uses them as labeled data to directly reconstruct the 4D content through a 4D Gaussian splatting model. Importantly, our method can achieve real-time rendering under continuous camera trajectories. To enable robust reconstruction under sparse views, we introduce inconsistency-aware confidence-weighted loss design, along with a lightly weighted score distillation loss. Extensive experiments on both synthetic and real videos show that Efficient4D offers a remarkable 10-fold increase in speed when compared to prior art alternatives while preserving the quality of novel view synthesis. For example, Efficient4D takes only 10 minutes to model a dynamic object, *vs* 120 minutes by the previous art model Consistent4D. Our code and video results are publicly available at https://fudan-zvg.github.io/Efficient4D/.

Keywords 4D Generation · Efficiency · Diffusion Model · Gaussian Splatting · Video

1 Introduction

Humans possess a remarkable capacity to comprehensively comprehend the spatial and temporal characteristics of a dynamic object in a brief video, even with a limited perspective, enabling them to predict its appearance in unseen viewpoints over time. Despite the significant advancement of 3D object generation, existing works (Poole et al., 2023; Lin et al., 2023; Chen et al., 2023; Wang et al., 2023) mostly consider static scenes or objects. With the availability of large-scale 3D datasets (Deitke et al., 2023a, b), training generalizable models capable of directly generating multi-view images becomes possible (Liu et al., 2023, 2024; Long et al., 2023). These generated images can be turned into a 3D object through reconstruction techniques (Mildenhall et al., 2021; Wang et al., 2021). By further augmenting these generated static objects with predefined animations (Maximo, 2023), dynamic 3D content can be generated. However, this approach is still limited due to the need for fine-grained meshes as well as rigid restrictions.

Directly generating 4D object/scene content from text description has been recently attempted (Singer et al., 2023). To bypass the need of exhaustively labeled training data pairs in form of (text, 4D), it trains a Neural Radiance Fields (NeRF)-like representation (Mildenhall et al., 2021) via score distillation sampling (Poole et al., 2023) and separates the modeling of static scene and its dynamics. Not only is this method computationally inefficient caused by heavy supervision back-propagation through a large pretrained model, but also its textual condition is highly ambiguous in expressing the intended visual content. In quest of the afore-

Communicated by Guofeng Zhang.

✉ Li Zhang
lizhangfd@fudan.edu.cn

Zijie Pan
zjpan19@fudan.edu.cn

Zeyu Yang
zeyuyang21@m.fudan.edu.cn

Xiatian Zhu
xiatian.zhu@surrey.ac.uk

[1] School of Data Science, Fudan University, Shanghai, China

[2] University of Surrey, Surrey, UK

Published online: 24 December 2025

mentioned human's capability, a recent work (Jiang et al., 2024) proposed to generate dynamic 3D object images from a single-view video (statically captured monocular video from a fixed view), namely as *video-to-4D* object generation. However, similar as Singer et al. (2023) this method is also slow to train (*e.g.*, 120 minutes to model a single dynamic object) in addition to complex design, hence unscalable and expensive in practice.

To address identified limitations, we formulate an efficient video-to-4D two staged object generation method called *Efficient4D*. In the first stage, we generate spacetime-consistent images across different camera views as synthetic training data. This is realized by imposing temporal smoothing into a multi-view image generator (*e.g.*, SyncDreamer Liu et al. 2024) in tandem with frame interpolation. In the second stage, we use these training data to optimize a 4D Gaussian splatting model (Yang et al., 2024). This is an extension of the 3D Gaussian splatting (Kerbl et al., 2023), originally designed for static 3D scene representation, with the temporal dimension introduced additionally, allowing for real-time rendering under continuous camera trajectories. Using Gaussian representation brings about further computational efficiency gain, when compared with NeRF based designs (Figure 1). To tackle the challenging discontinuity between the generated sparse frames, we design an inconsistency-aware loss function based on whether there is confidence in the consistency of a frame with its adjacent frames. Along with a lightly weighted score distillation sampling loss for smooth viewpoint transitions, this technique enables robust 4D reconstruction. Notably, although our method takes a single-view video as input, Efficient4D can be easily extended to image-to-4D task by leveraging a image-to-video diffusion model (Blattmann et al., 2023).

Our **contributions** are summarized as follows: **(i)** We consider for the first time the efficiency challenge with the under-studied video-to-4D object generation problem. **(ii)** We propose an efficient video-to-4D object generation pipeline, *Efficient4D*, characterized by directly generating high-quality training data without the need for heavy supervision back-propagation through a large pretrained model as suffered by most 3D/4D object generation approaches. We also extends Efficient4D to image-to-4D task. **(iii)** We introduce a inconsistency-aware confidence-weighted loss for reconstructing a 4D Gaussian splatting model using the generated training data efficiently and robustly. **(iv)** Extensive experiments on both synthetic and real data validate the significant efficiency advantage (*e.g.*, 10× speedup) of our Efficient4D over the prior art, whilst maintaining the quality of novel view synthesis. Also, our method can work well under the more challenging few-shot setting where only a handful of key frames are available for training, further extending the application scope.

2 Related work

3D generation 3D generation takes two main settigns: text-to-3D and image-to-3D. The pioneering work, DreamFusion (Poole et al., 2023), introduces the score distillation sampling (SDS) loss for optimizing 3D shapes with diffusion models. SDS's generality has prompted numerous subsequent efforts in both text-to-3D tasks (Lin et al., 2023; Metzer et al., 2023; Seo et al., 2024; Shi et al., 2023; Chen et al., 2023; Li et al., 2023; Tsalicoglou et al., 2024; Huang et al., 2023; Wu et al., 2024; Wang et al., 2023; Zhu and Zhuang, 204) and image-to-3D tasks (Xu et al., 2023; Tang et al., 2023; Melas-Kyriazi et al., 2023; Qian et al., 2024) across various dimensions. However, SDS-based approaches often suffer from difficulty to converge and extended optimization times. Conversely, some efficient methods (Nichol et al., 2022; Jun and Nichol, 2023; Liu et al., 2023, 2024, 2023) have emerged. Notably, Point-E (Nichol et al., 2022) and Shap-E (Jun and Nichol, 2023) train models to directly generate 3D point clouds or meshes. Zero123 (Liu et al., 2023) focuses on generating a 2D image from an unseen view based on a single image, convertible to a 3D shape through SDS or Liu et al. (2023). Importantly, SyncDreamer (Liu et al., 2024) produces multi-view consistent images, offering inspiration for reconstructing 3D objects.

4D representation Efforts to synthesize videos with free-viewpoint control in dynamic scenes have a well-documented history (Zitnick et al., 2004). For example, pre-NeRF (Mildenhall et al., 2021) approaches challenges in reconstructing intricate scene details. Recent advancements in 4D representations, particularly those based on neural rendering, include D-NeRF (Pumarola et al., 2021), DeVRF (Liu et al., 2022) and HyperNeRF (Park et al., 2021), which decouple geometry and motion, utilizing a canonical space and a learned deformation field. DynIBaR (Li et al., 2023) deploys an image-based rendering paradigm for representing long videos with complex camera and object motions. Another group of methods (Fridovich-Keil et al., 2023; Cao and Johnson, 2023; Shao et al., 2023) adopt tensor decomposition of 4D volumes to represent dynamic 3D scenes.

Recently, Gaussian Splatting (Kerbl et al., 2023) has received widespread attention for its real-time high-fidelity rendering, especially its explicit point-based representation which holds great potential in modeling dynamic scenes. Consequently, a significant amount of work has been proposed to explore its extension to dynamic scene modeling. Among them, Deformable 3D Gaussians (Yang et al., 2024) and 4DGaussian (Wu et al., 2024) integrated the deformation field with 3D Gaussian Splatting for the joint learning of the scene's geometry and dynamics. SC-GS (Huang et al., 2024) represents the motion with a set of sparse control points to achieve reconstruction and motion editing. Unlike the previous radiance field-based representations often involving

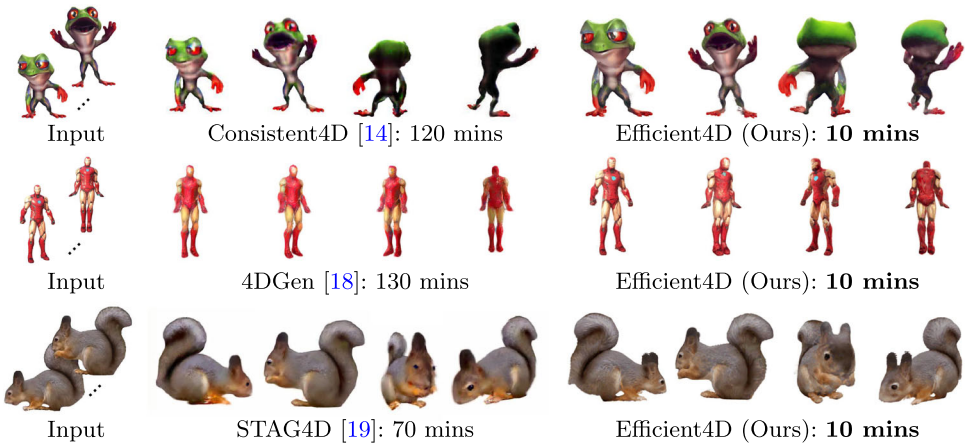

Fig. 1 Examples of video-to-4D generation. **Input**: A brief video of a dynamic object, as represented by 2 frames per case; **Output**: Generated novel views at different timestamps. The generation time is also shown for each method. More visualized comparisons can be found in Figure 5

complex training schedules or suffering from slow convergence, these methods can be optimized efficiently, achieving real-time rendering while surpassing the past methods in terms of quality.

Our work is perpendicular to all the above works, where any of them can be deployed in our reconstruction phase. But considering both optimization efficiency and expressive capability, we choose to represent dynamic 3D assets by a set of native 4D scene primitives, which is proved to be superior in 4DGS (Yang et al., 2024).

4D generation There are a few recent works dedicated for more challenging 4D object generation. For instance, MAV3D (Singer et al., 2023) deals with a text-to-4D problem by training Hexplane (Cao and Johnson, 2023) with a video diffusion model and SDS loss. Instead of text input, Consistent4D (Jiang et al., 2024) conditions the generation of 4D object over time on a monocular video with richer and more specific information. However, it is computationally inefficient due to inheriting the previous SDS loss, along with complex pipeline design. To overcome this limitation, we present a novel two-staged pipeline in a generation-and-reconstruction strategy, drastically boosting the training speed by 20× whilst maintaining the quality of novel view synthesis.

Recent works have attempted to generate 4D videos from a single input video. For example, TrajectoryCrafter (YU et al., 2025) leverages rendered point cloud as the condition of diffusion model to facilitate novel view synthesis. However, the point cloud is projected from the source video and its estimated depth, so there are no points from the back view, limiting the range of novel views. Another work ReCamMaster (Bai et al., 2025) trains a diffusion model which introduces frame dimension conditioning for input video and camera pose conditioning for target trajectory. But the trajectory is still limited to the front 180°. In contrast, our goal is to generate the complete 360° views for a dynamic object.

3 Preliminary

4D Gaussian splatting The 4D Gaussian splatting (4DGS Yang et al., 2024) builds upon the 3D Gaussian splatting technique introduced in Kerbl et al. (2023), originally designed for static scene representation. To address the complexities of dynamic scenes, 4DGS represents each Gaussian defined as:

$$G(\mathbf{p}|\mu, \Sigma) = e^{-\frac{1}{2}(\mathbf{p}-\mu)^\top \Sigma^{-1}(\mathbf{p}-\mu)}, \tag{1}$$

where $\mu \in \mathbb{R}^4$ is the mean vector, and $\Sigma \in \mathbb{R}^{4\times4}$ is the anisotropic covariance matrix. The input $\mathbf{p} = (\mathbf{x}, t) \in \mathbb{R}^4$ represents a spacetime position with a spatial coordinate $\mathbf{x}$ and time t. The covariance matrix Σ decomposes into a diagonal scaling matrix $S \in \mathbb{R}^{4\times4}$ and a rotation matrix $R \in \mathbb{R}^{4\times4}$ through $\Sigma = RSS^\top R^\top$. The 4D rotation R is represented by a pair of iso-rotations, each characterized by a quaternion.

For rendering, each Gaussian includes opacity α and view-dependent color $\mathbf{c}$ represented by spherical harmonics (SH). Given an arbitrary view $\mathcal{I}$ defined by intrinsic and extrinsic parameters, we render the pixel at position (u, v) with timestamp t by blending visible Gaussians:

$$\mathcal{I}(u, v, t) = \sum_i f_i(u, v, t)\alpha_i \mathbf{c}_i(d)T_i \tag{2}$$

$$\text{with } T_i = \prod_{j=1}^{i-1}(1 - f_j(u, v, t)\alpha_j). \tag{3}$$

Here, i indexes the visible Gaussians sorted by depth, d refers to the direction of the pixel under the view $\mathcal{I}$, and $f_i(u, v, t)$ denotes the influence of a Gaussian on this position. To obtain the influence, unlike 3D Gaussian Splatting (Kerbl et al., 2023) which directly projects 3D Gaussians to image space, we need condition 4D Gaussians on time and then project

them. More specifically, $f(u, v, t)$ is expressed by:

$$f(u, v, t) = G_t(t|\mu_t, \Sigma_t) G_{P(\mathbf{x})|t}(u, v|P(\mu_{\mathbf{x}|t}), P(\Sigma_{\mathbf{x}|t})), \tag{4}$$

where G_t is the marginal distribution of the 4D Gaussian in time, and $G_{P(\mathbf{x})|t}$ is the projected version of the conditional 3D Gaussian with

$$\mu_{\mathbf{x}|t} = \mu_{\mathbf{x}} + \Sigma_{\mathbf{x},t} \Sigma_t^{-1} (t - \mu_t), \tag{5}$$

$$\Sigma_{\mathbf{x}|t} = \Sigma_{\mathbf{x}} - \Sigma_{\mathbf{x},t} \Sigma_t^{-1} \Sigma_{t,\mathbf{x}}. \tag{6}$$

The projection operation P (Kerbl et al., 2023; Zwicker et al., 2001) projects the world point $\mu_{\mathbf{x}|t}$ to image space and transforms the covariance by $P(\Sigma_{\mathbf{x}|t}) = JW\Sigma_{\mathbf{x}|t}W^\top J^\top$ where W is the extrinsic matrix of $\mathcal{I}$ and J is the Jacobian of the affine approximation of the projective transformation.

Score distillation sampling Score distillation sampling (SDS) is first introduced by DreamFusion (Poole et al., 2023), which is used to distill the knowledge from a pretrained diffusion model ϵ_ϕ. Specifically, given an image I rendered from a scene representation (*e.g.*3DGS) parameterized by θ, the gradient of SDS loss is calculated as:

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, I_t) = \mathbb{E}\left[w(t) \left(\epsilon_\phi(I_t; t, c) - \epsilon \right) \frac{\partial I_t}{\partial \theta} \right], \tag{7}$$

where I_t is the perturbed image with noise ϵ at time step t, and c is the condition (*e.g.*one frame of the input video in this paper).

4 Method

Our Efficient4D addresses the challenge of efficiently generating dynamic objects under novel views from a single-view video. The input single-view video can be either provided by user or generated by a video generation model. The latter approach extends the application of our method beyond video-to-4D. For example, we can also achieve image-to-4D through a image-to-video diffusion model (Blattmann et al., 2023). As illustrated in Figure 2, it comprises two key components:

- An image synthesis pipeline (Figure 2(A)) generates images across views and timestamps, ensuring sufficient geometry and temporal consistency.
- An efficient and robust reconstruction process (Figure 2(B)) efficiently utilizes the synthetic images for accurate dynamic object reconstruction and novel view synthesis.

We will elaborate on these components in Section 4.1 and 4.2, respectively.

4.1 Image synthesis across views and timestamps

Due to the difficulty of obtaining calibrated 4D scans, our approach involves the direct generation of high-quality consistent 4D data from a single-view video which is much easier to capture (Figure 2(a)). Specifically, we seek to produce a $M \times N$ *image matrix* $\mathcal{D} = \{e_{ij}\}_{i,j=1}^{M,N}$ representing 2D images with geometrical and temporal consistency. Here, M denotes timestamps, and N represents views, with each matrix element corresponding to an image (Figure 2(c)). This approach combines conventional video (capturing time variation, represented by a single column in the image matrix) and 3D (capturing view variation, represented by a single row in the image matrix) generation (Singer et al., 2023; Liu et al., 2024), offering comprehensive information for modeling a dynamic object.

To initiate the image matrix $\mathcal{D}$, we set the first view (*i.e.*, the first column) with K frames from the input video and proceed to generate the remaining views. Our task involves generating multi-view consistent images from a single image for each row. Existing image-to-3D methods, such as SyncDreamer (Liu et al., 2024), can be leveraged for this purpose. However, these methods often struggle with temporal inconsistency within a specific view (*i.e.*, continuity in the column direction) due to the independent synthesis of multi-frame images. To address this issue, we propose an enhanced version of SyncDreamer with improved temporal continuity, referred to as *SyncDreamer-T*.

Specifically, SyncDreamer generates N multi-view images $\{\mathbf{x}_0^{(1)}, \ldots, \mathbf{x}_0^{(N)}\}$ of a static object using a synchronized N-view noise predictor $\{\epsilon_\theta^{(n)} | n = 1, \ldots, N\}$ that predicts synchronized noise for noisy multi-view images $\mathbf{x}_t^{1:N}$. The noise predictor is conditioned on information correlated with all views. Cross-view conditioning is achieved through a spatial feature volume $\mathcal{V} \in \mathbb{R}^{F \times V \times V \times V}$ unprojected by $\mathbf{x}_t^{1:N}$ to inject 3D-aware features into the shared noise predictor, ensuring geometrical consistency across views for static moments.

To impose temporal consistency, the information from $\mathcal{V}$ at different timestamps is aggregated using a time-synchronous spatial volume we design here (Figure 2(b)). A smoothing filter layer is introduced into the spatial volumes of different frames/timestamps, incorporating a weight vector $\mathbf{w} = (w_{-k}, \ldots, w_0, \ldots, w_k)$ which serves as the smooth filter. At each denoising step, time-synchronized spatial volumes $\tilde{\mathcal{V}}_i$ for each input frame $i \in \{1, 2, \ldots, M\}$ are constructed as:

$$\tilde{\mathcal{V}}_i = \sum_{j=-k}^{k} w_{i+j} \mathcal{V}_{i+j}. \tag{8}$$

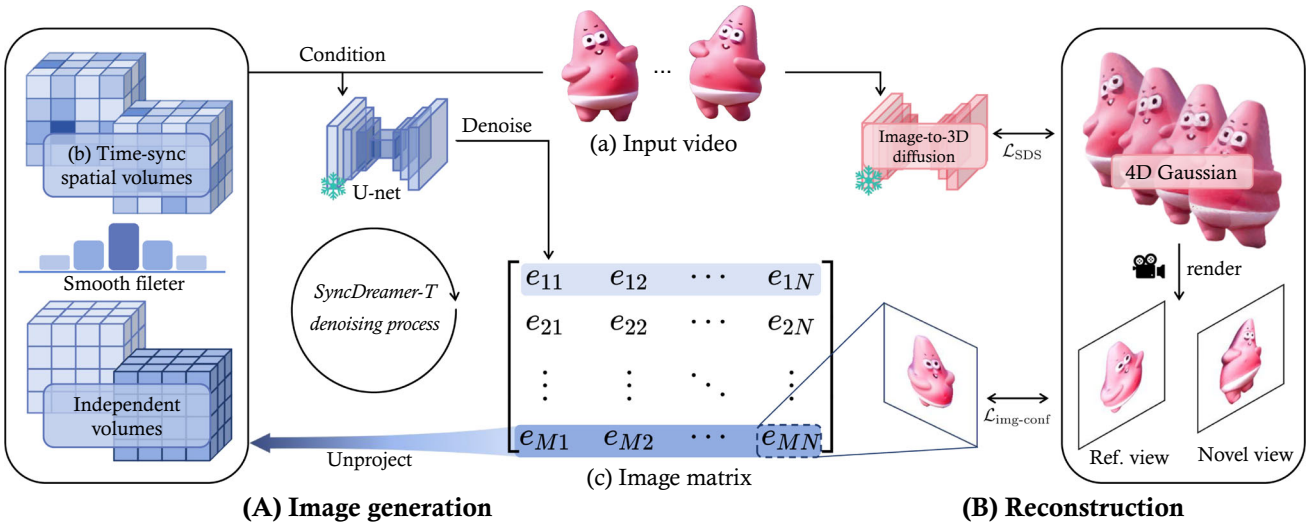

Fig. 2 Overview of our Efficient4D approach. Given as the input (a) a brief video depicting a dynamic object from a single perspectives, our model aims to generate this object with geometrical and temporal consistency under any specific view and time. Efficient4D comprises two components: **(A)** Image sequence synthesis through (b) time-synchronous spatial volumes, resulting in (c) an *image matrix* where each row consists of multi-view geometrically consistent images and each column consists of view-specific temporally consistent images. **(B)** 4D Reconstruction using the generated images in (A). The 4D Gaussian representation can be trained efficiently and robustly under the confidence-weighted loss $\mathcal{L}_{\text{img-conf}}$ and the low-weighted SDS loss $\mathcal{L}_{SDS}$

This synchronization ensures consistent features across k past and k future frames during the denoising process, thus enhancing temporal consistency. With this time-synchronized spatial volumes, the proposed *SyncDreamer-T* is entirely *training-free* established on the pretrained SyncDreamer.

To further improve temporal resolution, video frame interpolation (*e.g.*, RIFE Huang et al. 2022) can be applied after generating the image matrix $\mathcal{D}$. The midpoint interpolation is applied twice recursively, giving three additional frames between each pair of consecutive frames. This results in a total of $M = 4K - 3$ images in a column of $\mathcal{D}$.

Analysis on temporal synchronization For a clearer insight into our temporal synchronization design, we undertake a simplified experiment involving two input frames, each with its feature volumes labeled as $\mathcal{V}_1$ and $\mathcal{V}_2$. The fusion process is carried out by combining them as follows:

$$\tilde{\mathcal{V}}_i = (1 - w)\mathcal{V}_i + w\mathcal{V}_{3-i}, \tag{9}$$

where $i \in \{1, 2\}$ and w denotes the fusion ratio, we systematically vary the ratio w from 0 to 0.5, resulting in distinct columns of "Feature fusion" as illustrated in Figure 3. For $w = 0$, the original generation process is represented, where the spatial volumes of the two frames are independent, leading to temporal inconsistency. As w approaches 0.5, we observe a gradual convergence of the generated astronauts conditioned on different input images, achieving similarity in both texture and geometry. This suggests that smoothing at the feature level is effective in aligning frames over time.

However, a challenge arises as the fused features may induce similar motion. While the bottom-left astronaut is stepping forward, the one at the bottom-right column does not exhibit forward motion like the top astronaut. Thus, a trade-off is necessary between achieving temporal consistency and preserving motion independence. In practical terms, it is recommended to set the ratio just above 0.5, striking a balance that ensures temporal texture consistency at a moderate cost of entangled geometry. Also note that the results in Figure 3 are not sensitive to the choice.

4.2 4D generation through reconstruction

Aiming for 3D dynamic object modeling, discrete images not suffice. Our next goal is to model a truly 4D content from the image matrix $\mathcal{D}$. For efficient modeling, we have formulated the 4DGS representation model in Section 3, departing from previous slow-to-train 4D reconstruction models (Fridovich-Keil et al., 2023).

Optimization In the training of 4DGS, optimization is performed on the mean (μ), covariance (Σ), opacity (α), and spherical harmonic (SH) coefficients, as well as density control including densification and pruning for each Gaussian. The original loss function, as presented in Yang et al. (2024), involves both RGB loss and SSIM loss with balancing weights fixed. Specifically, the loss formulation is defined

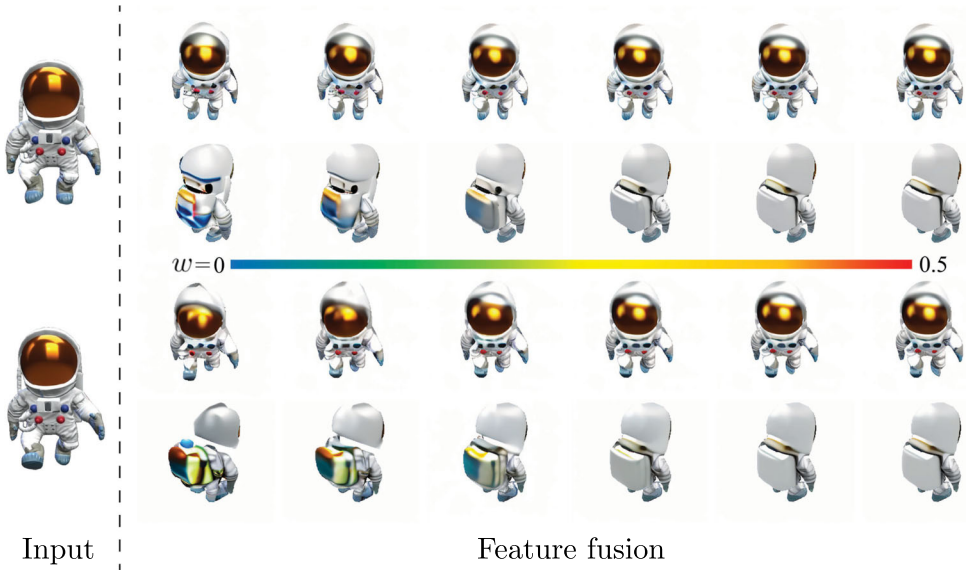

Fig. 3 Example analysis on temporal synchronization. In this illustration, we manipulate the fusion ratio (w) across a range from 0 to 0.5 for the spatial feature volumes of two input frames. The results indicate that a moderate ratio can achieve superior outcomes by balancing both temporal consistency and motion independence

as:

$$\mathcal{L}_{\text{img}} = \lambda_{\text{rgb}}\mathcal{L}_{\text{rgb}} + \lambda_{\text{ssim}}\mathcal{L}_{\text{ssim}}, \tag{10}$$

where $\mathcal{L}_{\text{rgb}}$ is the L_1 loss in RGB space, $\mathcal{L}_{\text{ssim}}$ is the SSIM loss (Wang et al., 2004), and $\lambda_{\text{rgb/ssim}}$ is the respective weight hyper-parameter.

However, such optimization approach assumes clean training data, which may not valid for synthetic data with inherent imperfections. To address this, we first introduce a inconsistency-aware loss formulation with adaptive balancing weights:

$$\lambda_{\text{rgb}} = \tilde{\lambda}_{\text{rgb}}\mathcal{C}_{\text{rgb}}, \quad \lambda_{\text{ssim}} = \tilde{\lambda}_{\text{ssim}}\mathcal{C}_{\text{ssim}}, \tag{11}$$

where $\tilde{\lambda}_{\text{rgb/ssim}}$ is fixed weight and $\mathcal{C}_{\text{rgb/ssim}}$ is the adaptive confidence score of a generated image $\mathbf{I}$ calculated as

$$\mathcal{C}_{\text{rgb}} = 1 - |\mathbf{I} - \hat{\mathbf{I}}|, \quad \mathcal{C}_{\text{ssim}} = \text{SSIM}(\mathbf{I}, \hat{\mathbf{I}}), \tag{12}$$

where $\hat{\mathbf{I}}$ is the unwarped image from adjacent frames estimated by optical flow. In such way, the confidence $\mathcal{C}_{\text{rgb/ssim}}$ function as an adaptive role in controlling the loss and gradient by assigning lower weights to inconsistent regions, thus enhancing overall reconstruction quality.

The confidence design can guarantee temporal consistency, but achieving high quality of novel view synthesis still remains challenging due to the sparsity of the generated images. Therefore, we also incorporate SDS loss with a relatively small weight λ_{SDS} for smooth transition across different supervised views. We use the image-to-3D diffusion model (Blattmann et al., 2023) in SDS loss conditioned on the frame of input video at each timestamp. The total loss function is expressed by:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{img-conf}} + \lambda_{\text{SDS}}\mathcal{L}_{\text{SDS}} \tag{13}$$

with $\mathcal{L}_{\text{img-conf}} = \tilde{\lambda}_{\text{rgb}}\mathcal{C}_{\text{rgb}}\mathcal{L}_{\text{rgb}} + \tilde{\lambda}_{\text{ssim}}\mathcal{C}_{\text{ssim}}\mathcal{L}_{\text{ssim}}$.

5 Experiments

5.1 Experiment setup

Competitors For comparison, we mainly focus on video-to-4D and image-to-4D task. The competitors include Consistent4D (Jiang et al., 2024) (ICLR2024), 4DGen (Yin et al., 2023) (ArXiv2024) and STAG4D (Zeng et al., 2024) (ECCV2024) for video-to-4D, and Animate124 (Zhao et al., 2023) (ArXiv2023) and DreamGaussian4D (Ren et al., 2023) (ArXiv2024) for image-to-4D. We obtained their results by running their released official code. We also compare SyncDreamer (Liu et al., 2024) partially by replacing SyncDreamer-T in ablation study (Section 5.6).

Evaluation data To showcase the versatility of our proposed method, we conducted extensive experiments using a diverse set of data sources. For video-to-4D, we focused on 36 sequences: 32 sequences released by Jiang et al. (2024) and 4 sequences processed by ourselves. Among the released data, seven sequences are synthetic data where ground truth are available. For image-to-4D, we used the ten images released by Zhao et al. (2023). Our four sequences are used for sparse input evaluation in Section 5.5, which only contain two frames for each sequence. Three of them, named *dragon*, *guard*, and *yoxi*, are rendered from 3D animated models obtained from Sketchfab (Skectchfab, 2023). The other one

named *yellow face* was collected from the internet. All the data are publicly available.

Evaluation metrics As 4D generation research is still at early stage, there is no well established metric yet. However, we adopt multiple metrics by referring related works for comprehensive evaluation. For evaluating on synthetic data, we use LPIPS score (Zhang et al., 2018) and CLIP similarity (Radford et al., 2021) between rendered images and ground truth following Jiang et al. (2024). For the cases without ground truth, we also use CLIP-similarity between generated images and input frames as Tang et al. (2024); Ren et al. (2023) to measure image quality. For temporal smoothness, following Yin et al. (2023); Esser et al. (2023); Geyer et al. (2023) we use CLIP-T to measure the similarity between adjacent frames of a generated video from different views, including front (CLIP-T-f), side (CLIP-T-s) and back (CLIP-T-b) views. To evaluate a 4D object completely for different methods, we obtain the generated images by rendering 320 images uniformly distributed in space and time, covering 16 viewpoints and 20 timestamps.

Implementation details In equation (8), the spatial volumes are smoothed locally in a sliding window style, so the consistency of generated images may be weakened as the frame number increase. In practice, we can adjust the weight following two rules: (i) the middle weight should be at least 0.5; (ii) At least half of the volumes should have a positive weight. Here we provide some sample weights which have a good effect:

$$\mathbf{w} = \begin{cases} (1.5, 7, 1.5), & 3 \le N < 5 \\ (1, 1, 6, 1, 1), & 5 \le N < 7 \\ (1, 1, 1, 6, 1, 1, 1), & 7 \le N < 13 \\ (1, 1, 1, 1, 1, 2, 14, 2, 1, 1, 1, 1, 1), & N \ge 13, \end{cases} \tag{14}$$

where N denotes the frame number. The weight will be normalized to ensure the sum is 1.

In denosing process, we follow the default setting of Liu et al. (2024) using a improved sampling strategy introduced by HarmonyView (Woo et al., 2024). HarmonyView redefines the score estimation by decomposing consistency and diversity. Please refer to Woo et al. (2024) for details.

In the reconstruction stage, Gaussians are initialized randomly inside the sphere with radius space 0.5 with identity rotations and a initial number of 50,000 without densification. Training of 4D Gaussian Splatting is carried out using the Adam optimizer for 500 iterations with batch size 1. All other 4DGS hyperparameters remain consistent with those in Yang et al. (2024). The balancing weights in loss function (equation 13) are set as $\tilde{\lambda}_{\text{rgb}} = 8000$, $\tilde{\lambda}_{\text{ssim}} = 2000$, $\lambda_{\text{SDS}} = 1$. In equation (12), the calculation of estimated image $\mathbf{I}$ is implemented by frame interpolation using a optical flow based model (Huang et al., 2022). For each generated frame I_t, we use its adjacent four frames to interpolate the estimated frame as follows:

$$\hat{I}_{t1} = \text{Interp}(I_{t-1}, I_{t+1}), \quad \hat{I}_{t2} = \text{Interp}(I_{t-2}, I_{t+2}). \tag{15}$$

Then we can compute the mean RGB confidence score:

$$\mathcal{C}_{\text{rgb}} = 1 - \frac{1}{2}\left(|I_t - \hat{I}_{t1}| + |I_t - \hat{I}_{t2}|\right). \tag{16}$$

The SSIM confidence score is similar by using equation (12).

For speed efficiency, our proposed method only costs about 2 minutes for image generation if parallel denoising is allowed and 8 minutes for reconstruction on one A6000 GPU.

5.2 Evaluation on synthetic data

We first present the results on synthetic data in Figure 4 with ground truth shown. Since the data only include videos, we compare with two video-to-4D methods: Consistent4D (Jiang et al., 2024), 4DGen (Yin et al., 2023) and STAG4D (Zeng et al., 2024). Our visual results exhibit superior accuracy in both texture and geometry when compared to the ground truth, such as the direction of dinosaur's head, the color of trump's arm and the teeth of skull. The quantitative metrics are also presented in Table 1. The better CLIP and LPIPS scores further validate our superior generation results.

5.3 Video-to-4D comparison

For data without ground truth, we present qualitative comparisons in Figure 5. When assessing texture quality, it is important to note that all methods fall under the category of lifting 2D to 4D. However, Consistent4D produce watercolor-like images with low fidelity, which can be exemplified by the blurry edges of their rendered images. We attribute this blur to conflicts from multiple supervisory signals during prolonged optimization. In contrast, our method excels in directly generating high-quality 2D images, thus providing a strong and consistent supervision in the reconstruction stage. Although 4DGen and STAG4D also use pseudo labels, their images are inconsistent in the temporal coordinate and SDS loss still dominates the optimization. Therefore, 4DGen lacks details and STAG4D suffers from floaters and blur. Besides, our method's reconstruction stage supports rendering under any viewpoints while the images generated in stage-1 are discrete and sparse with a fixed elevation 30°. Overall, our method consistently outperforms the baseline methods in most cases.

In Table 2, we compare our method with baselines quantitatively. The metrics and data used are described in Section 5.1. We draw several observations: (i) Our method achieves superior image quality and temporal smoothness on

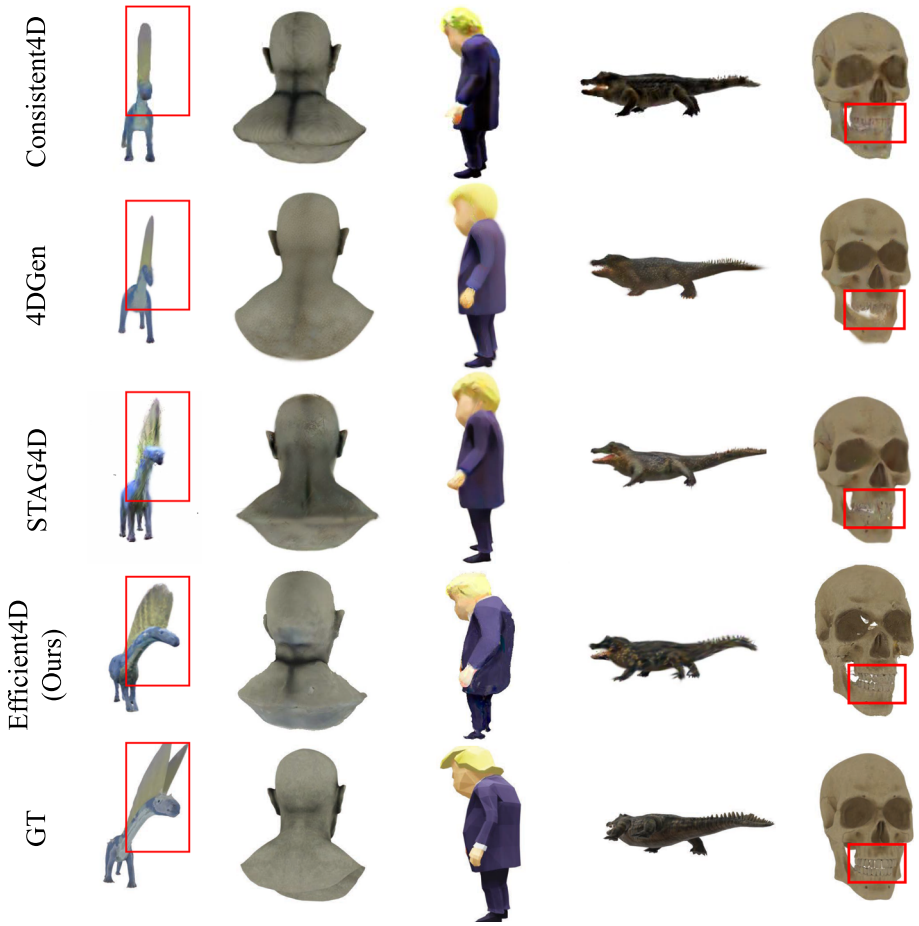

Fig. 4 Qualitative evaluation against ground truth (GT) on synthetic data. We compare our Efficient4D with Consistent4D (Jiang et al., 2024), 4DGen (Yin et al., 2023) and STAG4D (Zeng et al., 2024)

Table 1 Quantitative evaluation on synthetic data. We report CLIP and LPIPS scores between rendered images and ground truth images

	Consistent4D (Jiang et al., 2024)	4DGen (Yin et al., 2023)	STAG4D (Zeng et al., 2024)	Efficient4D(Ours)
CLIP ↑	0.87	0.89	0.91	**0.92**
LPIPS ↓	0.16	0.14	**0.13**	**0.13**

Table 2 Quantitative comparisons with state-of-the-art methods on both video-to-4D and image-to-4D generation

Method	CLIP↑	CLIP-T-f↑	CLIP-T-s↑	CLIP-T-b↑	Generation time↓
- *Video-to-4D comparison*					
Consistent4D Jiang et al. (2024)	0.8471	0.9692	0.9658	0.9697	120 mins
4DGen Yin et al. (2023)	0.8730	0.9568	0.9568	0.9573	130 mins
STAG4D Zeng et al. (2024)	0.8398	**0.9766**	**0.9731**	0.9760	70 mins
Efficient4D (Ours)	**0.8745**	**0.9766**	0.9609	**0.9780**	**10 mins**
- *Image-to-4D comparison*					
Animate124 Zhao et al. (2023)	0.8076	**0.9673**	**0.9639**	**0.9541**	540 mins
DreamGaussian4D Ren et al. (2023)	0.8242	0.8999	0.9072	0.9038	12 mins
Efficient4D (Ours)	**0.8350**	0.9346	0.9297	0.9321	**10 mins**

Fig. 5 Qualitative comparisons on video-to-4D generation. We compare our Efficient4D with Consistent4D (Jiang et al., 2024), 4DGen (Yin et al., 2023) and STAG4D (Zeng et al., 2024). For each case, we show four images per method with 0° elevation. Our Efficient4D comprises two stages: image generation stage (Our stage-1, 30° elevation) and reconstruction stage (Our stage-2, 0° elevation)

all cases validated by the higher CLIP and CLIP-T scores; (ii) Our method significantly accelerates the generation process, achieving over 10× speed improvement (10 mins *vs* 120 mins). More specifically, the speed improvement is attributed to (i) our image supervision design, converging optimization faster and reduce the training iterations, and (ii) each iteration in our method requiring much less time thanks to the efficiency of Gaussian representation.

5.4 Image-to-4D comparison

Next, we will evaluate our method on the image-to-4D task. In Figure 6, we present the visual results of three different methods. It can be observed that our extended method is capable of generating 4D assets with higher quality compared to those produced by state-of-the-art image-to-4D methods. Animate124 generates textures lacking in detail, and DreamGaussian4D tends to produce fragmented meshes. In contrast, the results of our Efficient4D exhibit both good geometry and high-quality textures. Additionally, Table 2 supports the similar conclusion that our method is superior, as seen in the video-to-4D comparison. Although Animate124 achieves the best CLIP-T score, This score merely indicates that the range of motion is small rather than the motion being continuous. From Figure 6 we can easily find that the quality of Animate124 results are lower than any other methods. Another notable point is that the overall CLIP scores for image-to-4D methods are lower than those for video-to-4D methods. This may be because the reference single-view videos in image-to-4D are generated by video diffusion models, which are inconsistent and have lower quality compared to real-world videos.

5.5 Challenging cases

We assessed our method's performance on two challenging cases: extremely sparse input and rotation dynamics. In sparse case, the input comprising only two discrete frames. In such cases, we set $w = 0.25$ in equation (9). We also modified the code of Consistent4D (Jiang et al., 2024) to make it adapt to such case. As illustrated in Figure 7, our approach successfully generates images featuring smooth motion and high spatiotemporal consistency. In contrast, Consistent4D fails to operate effectively under such conditions. Consider a scenario where we seek 4D modeling for a static toy. While the toy can take different poses, it lacks autonomous movement, posing a challenge for continuous video capturing. In these instances, our method demonstrates effectiveness by requiring only a few key frames to produce dynamic content, thereby expanding the potential applications of the 4D generation task.

For another rotation case, we adopt the data introduced in Yang et al. (2025). As shown in the first row of Figure 8, the character will rotate its body in the monocular video input, which is complex and challenging. However, our approach can also successfully infer the accurate geometry while Con-

Fig. 6 Qualitative comparisons on image-to-4D generation. We compare our Efficient4D with Animate124 (Zhao et al., 2023) and DreamGaussian4D (Ren et al., 2023) (DG4D). For each case, we show four images per method with 0° elevation. Our Efficient4D comprises two stages: image generation stage (Our stage-1, 30° elevation) and reconstruction stage (Our stage-2, 0° elevation)

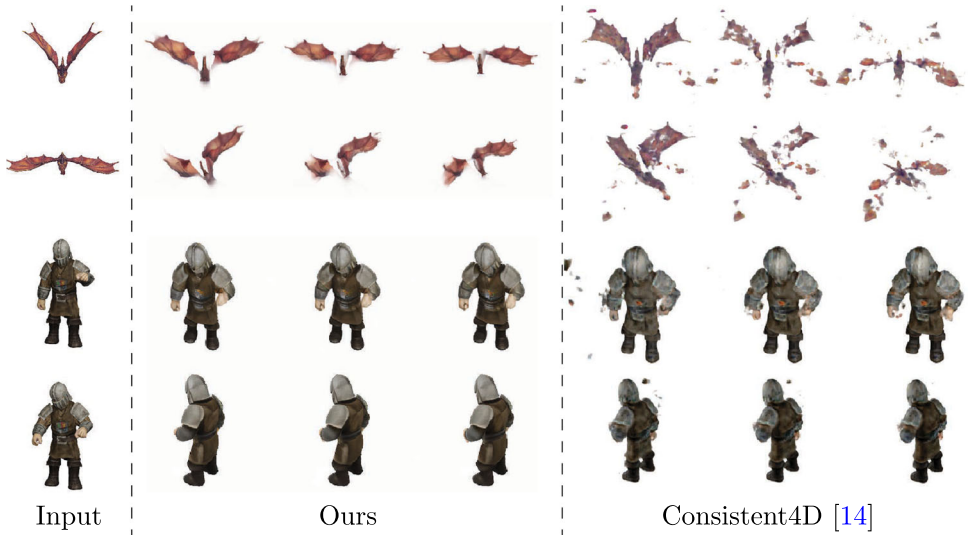

Fig. 7 Given only two input frames, our method is able to generate smooth dynamics. For each case, we show three internal images from two novel views

sistent4D tends to generate fragments. For example, from the Consistent4D results in Figure 8, we observe that the punching figure and the kicking man both suffer fragmented arms or legs.

5.6 Ablation studies

5.6.1 Ablation of image generation

We first performed ablation studies to assess the influence of different components in our image generation stage. To better evaluate the image quality itself, we reconstruct the 4D Gaussian without SDS loss. We compared our full method against three baseline settings: **(1)** Only input video are used for reconstruction; **(2)** Time-synchronous spatial volume is excluded; **(3)** Frame interpolation is excluded. Note that setting **(2)** is equivalent to the case where SyncDreamer-T is replaced with original SyncDreamer. The results corresponding to these settings can be found in Figure 9. Table 3 also gives quantitative evaluations. Beyond the above components, we also ablate the design of temporal synchronization in Figure 10.

Importance of synthetic data We compare our generated image matrix with utilizing only input video using the same 4D Gaussian representation model. As shown in the first row of Figure 9, when only relying on a single-view video,

Fig. 8 Performance on cases including rotation. We show two novel views for each case

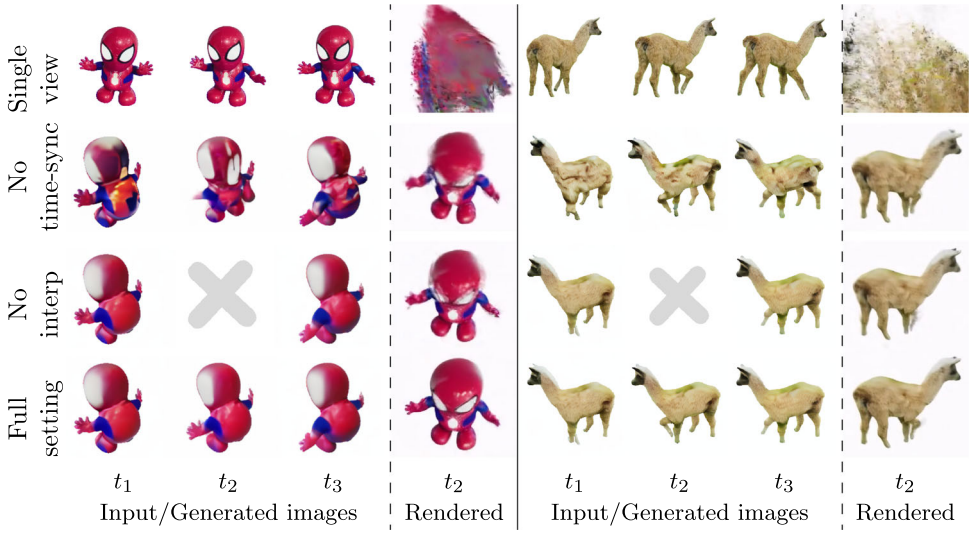

Fig. 9 Ablation study on image generation, time-synchronous spatial volumes and frame interpolation. For each case, we show the input images in the first row and three novel-view images generated by SyncDreamer-T at different timestamps in the other rows, followed by one rendered image from the reconstructed 4DGS at the middle timestamp. Each column refers to a specific timestep. For the row "No interp", there is no interpolated input image in the middle, thus leading to a vacancy at timestamp t_2

Table 3 Quantitative ablation study on all our modules mesured on 4 sequences: alpaca, astronaut, rabbit and spiderman

Method	CLIP ↑	CLIP-T-f ↑	CLIP-T-s ↑	CLIP-T-b ↑
Single-view	0.6684	-	-	-
No time-sync	0.8270	0.9258	0.9197	0.8819
No interp	0.8595	0.9409	0.9443	0.9336
Full setting	**0.8702**	**0.9689**	**0.9673**	**0.9684**

the model cannot produce any meaningful results for novel views. This indicates the importance of constructing proper training data.

Effect of time-synchronous spatial volume We assess the impact of the time-synchronous spatial volume concept introduced in SyncDreamer, as depicted in the contrast of the second and last rows in Figure 9. Without time-synchronous spatial volume, the back aspects of the toy Spiderman exhibit inconsistencies, leading to distorted geometry. In contrast, the proposed time-synchronous spatial volume enhances both spatial and temporal consistency while preventing geometry collapse, resulting in more visually appealing image generation and higher CLIP-T scores in Table 3.

Effect of frame interpolation As illustrated in the contrast between the third and last rows in Figure 9, frame interpolation is effective in mitigating the blurring observed in the

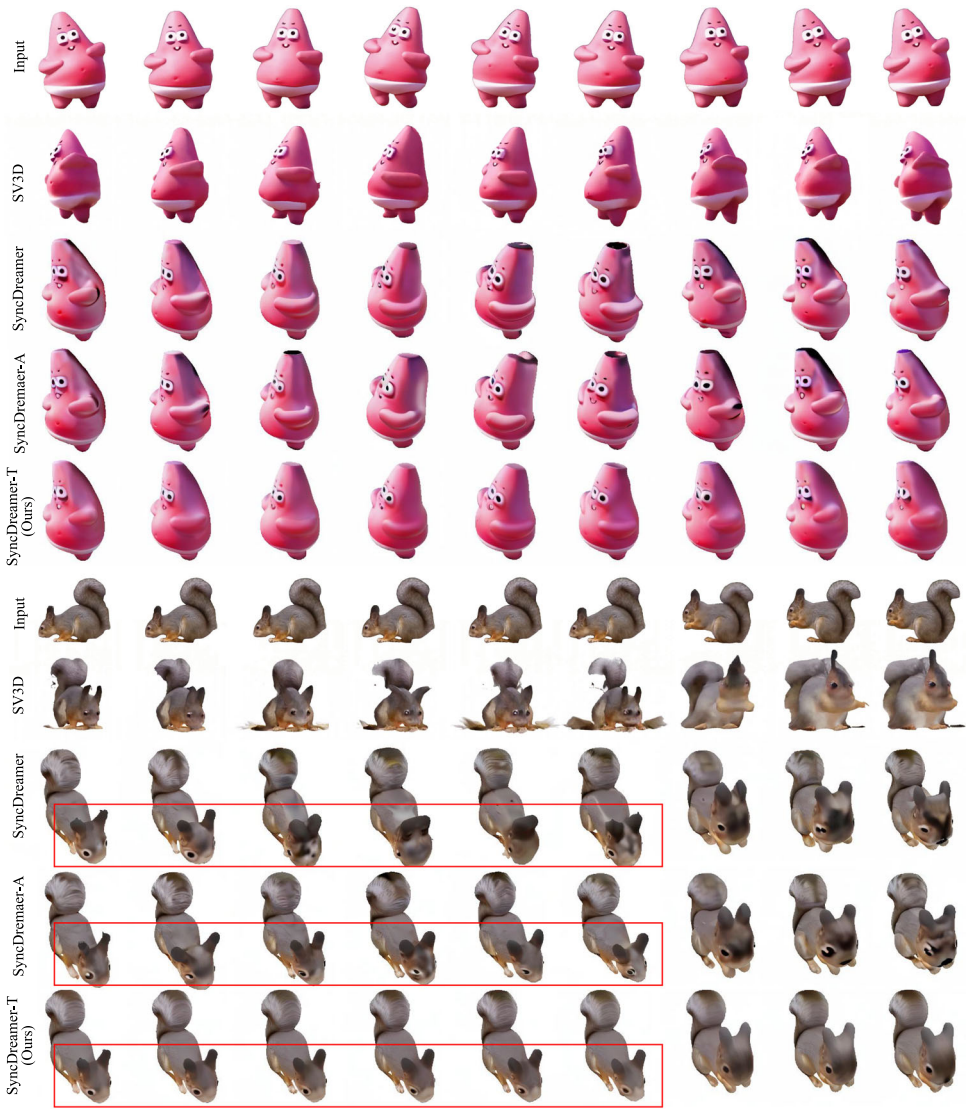

Fig. 10 Ablation study on the temporal synchronization. We compare SV3D (Voleti et al., 2024), SyncDreamer (Liu et al., 2024), SyncDreamer with consistent self-attention (Zhou et al., 2024) (SyncDreamer-A) and our proposed Syncdreamer-T. We show nine consecutive frames for each case to better evaluate the temporal consistency

rendered image from novel views, thus also delivering higher CLIP-T scores in Table 3. This is attributed to the low frame rates of the image matrix, which results in noticeable discontinuities.

Ablation on temporal synchronization We also examine the design of temporal synchronization. In conceptual terms, directly applying image-to-3D methods to each frame independently is inferior due to intrinsic generative randomness. This can be validated in Figure 10 by comparing SV3D (Voleti et al., 2024), SyncDreamer (Liu et al., 2024) and SyncDremaer-T(Ours). The image-to-3D methods (SV3D and SyncDreamer) cannot generate a consistent shape or texture across frames. Therefore, a proper temporal synchronization mechanism is necessary. In Section 4.1, we propose SyncDreamer-T, which blends the 3D volume feature. We also implement a consistent self-attention mechanism introduced in Zhou et al. (2024) in SyncDreamer (Liu et al., 2024), resulting in Syncdreamer-A. Specifically, in each self-attention block that projects Q_i, K_i, V_i corresponding to the image I_i, we randomly sample another image I_j's K_j, V_j, and concatenate two Ks and Vs as the new K_i and V_i. The results in Figure 10 indicate that our method generates a more consistent texture and smoother motion than SyncDreamer-A. We ascribe this to the following reason: The cross-attended features may focus too much on the details that its control on global texture and motion is weakened.

5.6.2 Ablation of reconstruction stage

In the reconstruction stage, we will study the effect of three components: **(1)** confidence map, **(2)** supervision of generated images, **(3)** SDS loss, and **(4)** choice of 4D representation. The visual results are shown in Figure 11, 12 and 13.

Effect of confidence map The integration of a confidence-weighted loss in our design serves as a strategy to mitigate

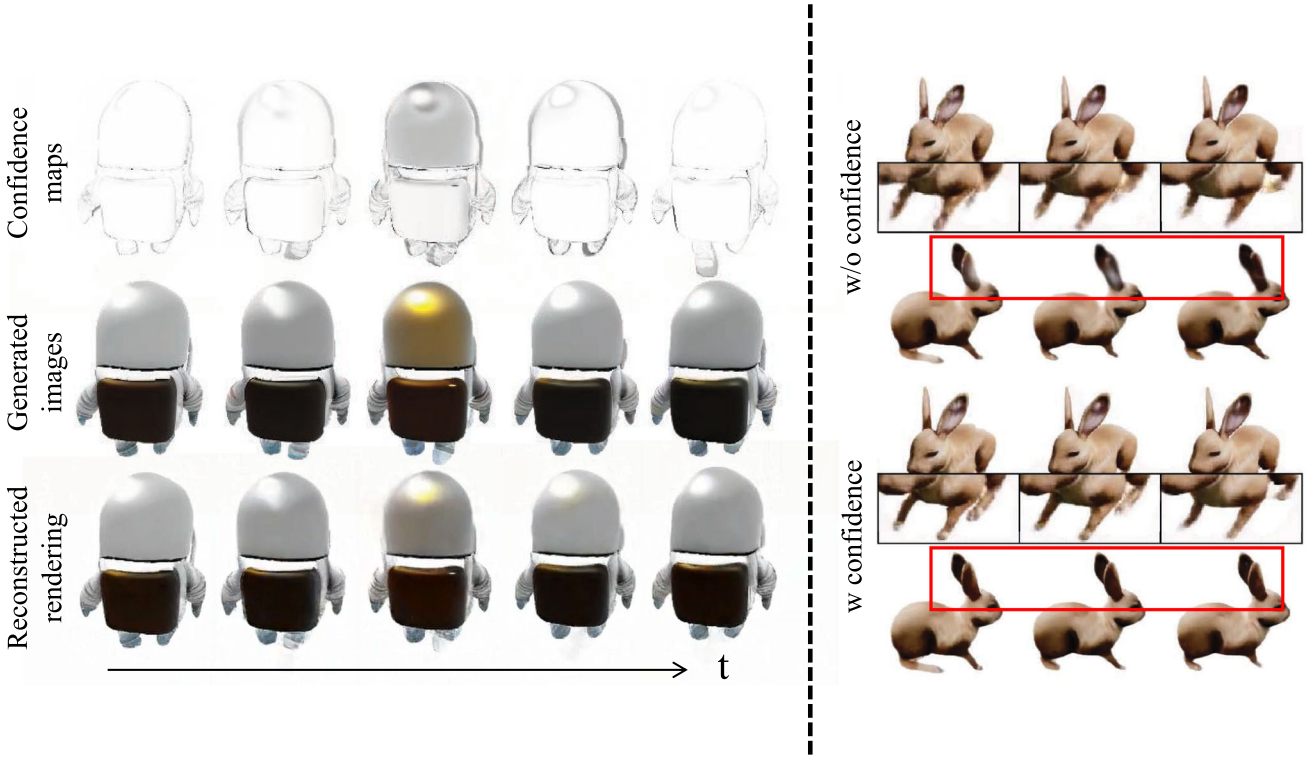

Fig. 11 Ablation study on confidence maps. The images follow a chronological order from left to right

Fig. 12 **Ablation study** on image supervision and SDS loss

training data noise. In the left of Figure 11, we study the effect of confidence map when obvious inconsistency exists in the generated images. Our temporal smooth mechanism can support consistency for most areas, whilst a small fraction of lower-quality regions may introduce conflicting gradients in the reconstruction stage, thus hurting the overall quality. To mitigate this, our confidence aware design comes into play for weakening the supervision from those inconsistent regions. Since these weakly supervised regions only occupy a small proportion across all views, the missing information about texture or geometry can be compensated by the redundancy of other views and generalization ability of 4D Gaussians. As shown in the left of Figure 11, the inclusion of confidence maps effectively reduces blurry and inconsistent rendering even when the generated images have plausible consistency, resulting in a significant overall improvement in quality and temporal smoothness.

Importance of image supervision and SDS loss In Figure 12, we compare the full setting of our method with baselines without image supervision (w/o image) or SDS loss (w/o SDS). The absence of anchored image supervision results in bad geometry, such as the hole in the elephant ears and multiple legs of the lion. This can be also attributed to the conflicts of SDS loss during prolonged optimization. In

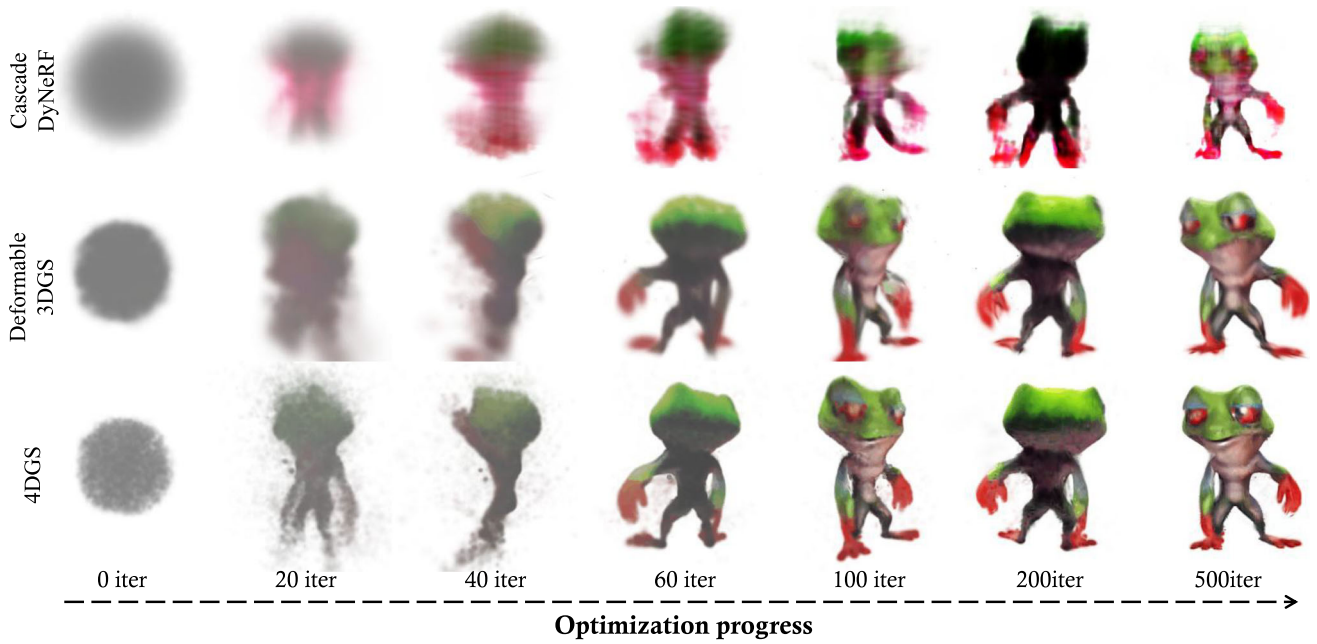

Fig. 13 Ablation study on the choice of 4D representation. We apply the same optimization strategy to three 4D representations: Cascade-DyNeRF (Jiang et al., 2024), Deformable-3DGS (Wu et al., 2024) and 4DGS (Yang et al., 2024). 4DGS features the fastest convergence speed

addition, due to the sparsity of the generated images, the novel views may not be reconstructed well, leading to blur and floaters. By contrast, our full method avoids the disadvantages of both, being able to render clean images with good geometry.

Choice of 4D representation Our synthetic training data is versatile and supports the optimization of 4D representation models, such as Cascade-DyNeRF (Jiang et al., 2024) and Deformable-3DGS (Wu et al., 2024) used in other 4D generation works (Jiang et al., 2024; Ren et al., 2023). In the comparison shown in Figure 13, 4DGS (Yang et al., 2024) converge in 500 iterations while the others do not. Also, thanks to the fast convergence speed of 4DGS, our method achieves significant efficiency advantage.

6 Conclusion

This study introduces a new framework, Efficient4D, designed for generating efficiently dynamic 4D objects seamlessly from monocular videos captured by a stationary camera. The Efficient4D consists of two main stages: first, generating consistent multi-view videos with spatial and temporal coherence, and second, rapidly producing 4D object reconstructions. Our approach, utilizing image supervision with lightly weighted SDS loss, significantly accelerates the generation process, achieving about 10 times faster speeds compared to previous works, while still delivering superior reconstruction and novel view synthesis results. Moreover, our model is effective in extremely sparse input scenarios, requiring only two available images, thereby expanding its application scope.

Acknowledgements This work was supported in part by New Generation Artificial Intelligence-National Science and Technology Major Project (2025ZD0123004), Ningbo grant (2025Z038) and National Natural Science Foundation of China (Grant No. 62376060).

Data Availability The datasets used for evaluation and comparison are available at Consistent4D (Jiang et al., 2024) (https://github.com/yanqinJiang/Consistent4D), Animate124 (Zhao et al., 2023) (https://github.com/HeliosZhao/Animate124/tree/main/benchmark) and Sketchfab (Skectchfab, 2023) (https://sketchfab.com/).

References

Bai, J., Xia, M., Fu, X., Wang, X., Mu, L., Cao, J., Liu, Z., Hu, H., Bai, X. Wan, P., and others. (2025). Recammaster: Camera-controlled generative rendering from a single video. arXiv preprint.

Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al. (2023). Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint*.

Cao, A., & Johnson, J. (2023). Hexplane: A fast representation for dynamic scenes. In: *ICCV*.

Chen, R., Chen, Y., Jiao, N., & Jia, K. (2023). Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In: *ICCV*.

Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., and others. (2023). Objaverse-xl: A universe of 10m+ 3d objects. In: *NeurIPS*.

Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., & Farhadi, A. (2023). Objaverse: A universe of annotated 3d objects. In: *CVPR*.

Esser, P., Chiu, J., Atighehchian, P., Granskog, J., & Germanidis, A. (2023). Structure and content-guided video synthesis with diffusion models. In: *ICCV*.

Fridovich-Keil, S., Meanti, G., Warburg, F.R., Recht, B., & Kanazawa, A. (2023). K-planes: Explicit radiance fields in space, time, and appearance. In: *CVPR*.

Geyer, M., Bar-Tal, O., Bagon, S., & Dekel, T. (2023). Tokenflow: Consistent diffusion features for consistent video editing. arXiv preprint.

Huang, Y.-H., Sun, Y.-T., Yang, Z., Lyu, X., Cao, Y.-P., & Qi, X. (2024). Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In: *CVPR*.

Huang, Y., Wang, J., Shi, Y., Qi, X., Zha, Z.-J., & Zhang, L. (2023). Dreamtime: An improved optimization strategy for text-to-3d content creation. In: *CVPR*.

Huang, Z., Zhang, T., Heng, W., Shi, B., & Zhou, S. (2022). Real-time intermediate flow estimation for video frame interpolation. In: *ECCV*.

Jiang, Y., Zhang, L., Gao, J., Hu, W., & Yao, Y. (2024). Consistent4d: Consistent 360° dynamic object generation from monocular video. In: *ICLR*.

Jun, H., & Nichol, A. (2023). Shap-e: Generating conditional 3d implicit functions. arXiv preprint.

Kerbl, B., Kopanas, G., Leimkühler, T., & Drettakis, G. (2023). 3d gaussian splatting for real-time radiance field rendering. In: *ACM TOG*.

Li, Y., Dou, Y., Shi, Y., Lei, Y., Chen, X., Zhang, Y., Zhou, P., & Ni, B. (2023). Focaldreamer: Text-driven 3d editing via focal-fusion assembly. In: *PLMR*.

Li, Z., Wang, Q., Cole, F., Tucker, R., & Snavely, N. (2023). Dynibar: Neural dynamic image-based rendering. In: *CVPR*.

Lin, C.-H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.-Y., & Lin, T.-Y. (2023). Magic3d: High-resolution text-to-3d content creation. In: *CVPR*.

Liu, J.-W., Cao, Y.-P., Mao, W., Zhang, W., Zhang, D.J., Keppo, J., Shan, Y., Qie, X., & Shou, M.Z. (2022). Devrf: Fast deformable voxel radiance fields for dynamic scenes. In: *NeurIPS*.

Liu, Y., Lin, C., Zeng, Z., Long, X., Liu, L., Komura, T., & Wang, W. (2024). Syncdreamer: Learning to generate multiview-consistent images from a single-view image. In: *ICLR*.

Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., & Vondrick, C. (2023). Zero-1-to-3: Zero-shot one image to 3d object. In: *ICCV*.

Liu, M., Xu, C., Jin, H., Chen, L., Xu, Z., Su, H., and others. (2023). One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. arXiv preprint.

Long, X., Guo, Y.-C., Lin, C., Liu, Y., Dou, Z., Liu, L., Ma, Y., Zhang, S.-H., Habermann, M., Theobalt, C., and others. (2023). Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv preprint*.

Maximo. (2023). https://www.mixamo.com/

Melas-Kyriazi, L., Laina, I., Rupprecht, C., & Vedaldi, A. (2023). Realfusion: 360deg reconstruction of any object from a single image. In: *CVPR*.

Metzer, G., Richardson, E., Patashnik, O., Giryes, R., & Cohen-Or, D. (2023). Latent-nerf for shape-guided generation of 3d shapes and textures. In: *CVPR*.

Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., & Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM.

Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., & Chen, M. (2022). Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint*.

Park, K., Sinha, U., Hedman, P., Barron, J.T., Bouaziz, S., Goldman, D.B., Martin-Brualla, R., & Seitz, S.M. (2021). Hypernerf: a higher-dimensional representation for topologically varying neural radiance fields. In: *ACM TOG*.

Poole, B., Jain, A., Barron, J.T., & Mildenhall, B. (2023). Dreamfusion: Text-to-3d using 2d diffusion. In: *ICLR*.

Pumarola, A., Corona, E., Pons-Moll, G., & Moreno-Noguer, F. (2021). D-nerf: Neural radiance fields for dynamic scenes. In: *CVPR*.

Qian, G., Mai, J., Hamdi, A., Ren, J., Siarohin, A., Li, B., Lee, H.-Y., Skorokhodov, I., Wonka, P., Tulyakov, S., and others. (2024). Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. In: *ICLR*.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., and others. (2021). Learning transferable visual models from natural language supervision. In: *ICML*.

Ren, J., Pan, L., Tang, J., Zhang, C., Cao, A., Zeng, G., & Liu, Z. (2023). Dreamgaussian4d: Generative 4d gaussian splatting. arXiv preprint

Seo, J., Jang, W., Kwak, M.-S., Ko, J., Kim, H., Kim, J., Kim, J.-H., Lee, J., & Kim, S. (2024). Let 2d diffusion model know 3d-consistency for robust text-to-3d generation. In: *ICLR*.

Shao, R., Zheng, Z., Tu, H., Liu, B., Zhang, H., & Liu, Y. (2023). Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In: *CVPR*.

Shi, Y., Wang, P., Ye, J., Long, M., Li, K., & Yang, X. (2023). Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint*.

Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., and others. (2023). Make-a-video: Text-to-video generation without text-video data. In: *ICLR*.

Singer, U., Sheynin, S., Polyak, A., Ashual, O., Makarov, I., Kokkinos, F., Goyal, N., Vedaldi, A., Parikh, D., Johnson, J., and others. (2023). Text-to-4d dynamic scene generation. In: *ICML*.

Sketchfab. (2023). https://sketchfab.com/.

Tang, J., Ren, J., Zhou, H., Liu, Z., & Zeng, G. (2024). Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In: *ICLR*.

Tang, J., Wang, T., Zhang, B., Zhang, T., Yi, R., Ma, L., & Chen, D. (2023). Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. In: *ICCV*.

Tsalicoglou, C., Manhardt, F., Tonioni, A., Niemeyer, M., & Tombari, F. (2024). Textmesh: Generation of realistic 3d meshes from text prompts. In: *3DV*.

Voleti, V., Yao, C.-H., Boss, M., Letts, A., Pankratz, D., Tochilkin, D., Laforte, C., Rombach, R., & Jampani, V. (2024). Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In: *ECCV*.

Wang, Z., Bovik, A.C., Sheikh, H.R., & Simoncelli, E.P. (2004). Image quality assessment: from error visibility to structural similarity. IEEE TIP.

Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., & Wang, W. (2021). Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In: *NeurIPS*.

Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., & Zhu, J. (2023). Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In: *NeurIPS*.

Woo, S., Park, B., Go, H., Kim, J.-Y., & Kim, C. (2024). Harmonyview: Harmonizing consistency and diversity in one-image-to-3d. In: *CVPR*.

Wu, J., Gao, X., Liu, X., Shen, Z., Zhao, C., Feng, H., Liu, J., & Ding, E. (2024). Hd-fusion: Detailed text-to-3d generation leveraging multiple noise estimation. In: *WACV*.

Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., & Xinggang, W. (2024). 4d gaussian splatting for real-time dynamic scene rendering. In: *CVPR*.

Xu, D., Jiang, Y., Wang, P., Fan, Z., Wang, Y., & Wang, Z. (2023). Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360deg views. In: *CVPR*.

YU, M., Hu, W., Xing, J., & Shan, Y..(2025). Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. arXiv preprint

Yang, Z., Gao, X., Zhou, W., Jiao, S., Zhang, Y., & Jin, X. (2024). Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In: *CVPR*.

Yang, Z., Pan, Z., Gu, C., & Zhang, L. (2025). Diffusion ^{2}: Dynamic 3d content generation via score composition of video and multi-view diffusion models. In: *ICLR*.

Yang, Z., Yang, H., Pan, Z., Zhu, X., & Zhang, L. (2024). Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. In: *ICLR*.

Yin, Y., Xu, D., Wang, Z., Zhao, Y., & Wei, Y. (2023). 4dgen: Grounded 4d content generation with spatial-temporal consistency. *arXiv preprint*.

Zeng, Y., Jiang, Y., Zhu, S., Lu, Y., Lin, Y., Zhu, H., Hu, W., Cao, X., & Yao, Y. (2024). Stag4d: Spatial-temporal anchored generative 4d gaussians. In: *ECCV*.

Zhang, R., Isola, P., Efros, A.A., Shechtman, E., & Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In: *CVPR*.

Zhao, Y., Yan, Z., Xie, E., Hong, L., Li, Z., & Lee, G.H. (2023). Animate124: Animating one image to 4d dynamic scene. arXiv preprint.

Zhou, Y., Zhou, D., Cheng, M.-M., Feng, J., & Hou, Q. (2024). Storydiffusion: Consistent self-attention for long-range image and video generation. In: *NeurIPS*.

Zhu, J., & Zhuang, P. (204). Hifa: High-fidelity text-to-3d with advanced diffusion guidance. In: *ICLR*.

Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S., & Szeliski, R. (2004). High-quality video view interpolation using a layered representation. In: *ACM TOG*.

Zwicker, Pfister, Baar, V., & Gross (2001). Ewa volume splatting. In: *Visualization, Vis*.