# High-Fidelity Novel View Synthesis via Splatting-Guided Diffusion

XIANG ZHANG, ETH Zürich, Switzerland and DisneyResearch|Studios, Switzerland
YANG ZHANG, DisneyResearch|Studios, Switzerland
LUKAS MEHL, DisneyResearch|Studios, Switzerland
MARKUS GROSS, ETH Zürich, Switzerland and DisneyResearch|Studios, Switzerland
CHRISTOPHER SCHROERS, DisneyResearch|Studios, Switzerland

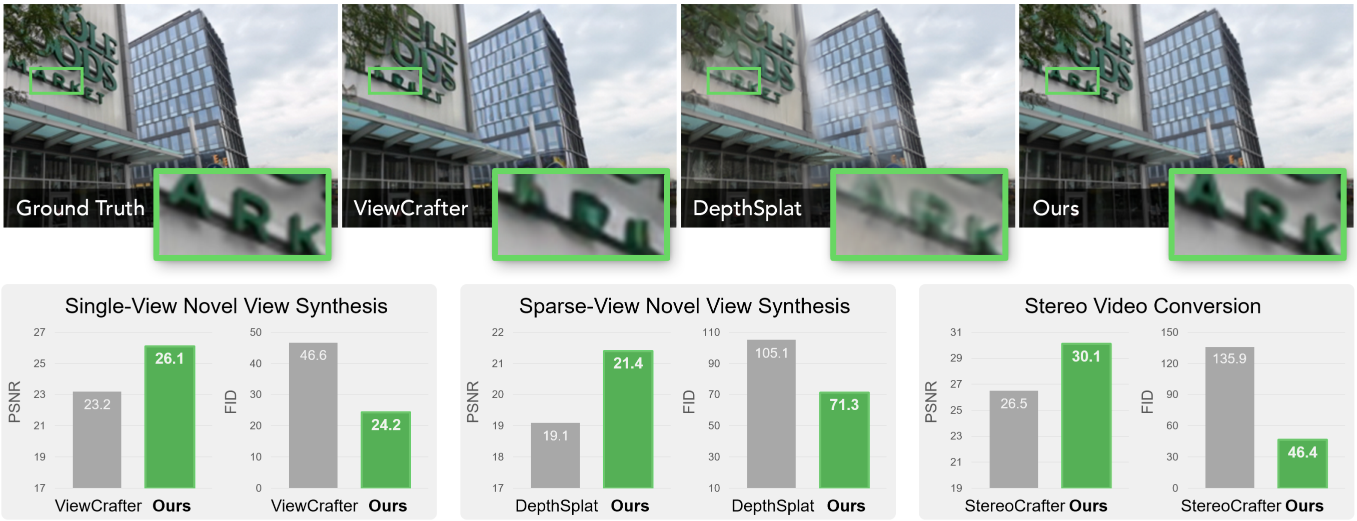


Fig. 1. **Performance comparison.** Diffusion-based methods, *e.g.*, ViewCrafter, usually hallucinate contents that are inconsistent with the input view. Splatting-based approaches, *e.g.*, DepthSplat, often suffer from distorted geometry due to splatting errors. By contrast, our method produces novel views with consistent geometry and high-fidelity texture, achieving significantly better performance than previous arts on different tasks. Note that our model is trained only on the single-view novel view synthesis and is directly applied to the other tasks, showing promising cross-domain and cross-task performance. Images credited to [Ling et al. 2024].

Despite recent advances in Novel View Synthesis (NVS), generating high-fidelity views from single or sparse observations remains challenging. Existing splatting-based approaches often produce distorted geometry due to splatting errors. While diffusion-based methods leverage rich 3D priors to achieve improved geometry, they often suffer from texture hallucination. In this paper, we introduce *SplatDiff*, a pixel-splatting-guided video diffusion model designed to synthesize high-fidelity novel views from a single image. Specifically, we propose an aligned synthesis strategy for precise control of target viewpoints and geometry-consistent view synthesis. To mitigate texture hallucination, we design a texture bridge module that enables high-fidelity texture generation through adaptive feature fusion. In this manner, SplatDiff leverages the strengths of splatting and diffusion for geometrically consistent, high-fidelity view synthesis. Extensive experiments verify the state-of-the-art performance of SplatDiff in single-view NVS. Additionally, without extra training, SplatDiff shows remarkable zero-shot performance across diverse tasks, including sparse-view NVS and stereo video conversion.

CCS Concepts: • **Computing methodologies → Computational photography**; **3D imaging**.

Additional Key Words and Phrases: Novel view synthesis, pixel splatting, video diffusion model

**ACM Reference Format:**
Xiang Zhang, Yang Zhang, Lukas Mehl, Markus Gross, and Christopher Schroers. 2025. High-Fidelity Novel View Synthesis via Splatting-Guided Diffusion. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3721238.3730669

Authors' Contact Information: Xiang Zhang, ETH Zürich, Zürich, Switzerland and DisneyResearch|Studios, Zürich, Switzerland, xiangz.ethz@gmail.com; Yang Zhang, DisneyResearch|Studios, Zürich, Switzerland, yang.zhang@disneyresearch.com; Lukas Mehl, DisneyResearch|Studios, Zürich, Switzerland, lukas.mehl.-nd@disneyresearch.com; Markus Gross, ETH Zürich, Zürich, Switzerland and DisneyResearch|Studios, Zürich, Switzerland, grossm@inf.ethz.ch; Christopher Schroers, DisneyResearch|Studios, Zürich, Switzerland, christopher.schroers@disneyresearch.com.

*SIGGRAPH Conference Papers '25, Vancouver, BC, Canada*
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1540-2/25/08
https://doi.org/10.1145/3721238.3730669

## 1 Introduction

Novel view synthesis (NVS) has attracted considerable interest in the fields of computer vision and computer graphics, showing various applications like augmented/virtual reality, 3D generation, and stereo video conversion [Gao et al. 2024; Mehl et al. 2024]. Compared with previous optimization-based NVS approaches, *e.g.*, neural radiance field (NeRF) [Mildenhall et al. 2020] and 3D Gaussian splatting (3DGS) [Kerbl et al. 2023], that usually require dense input views and per-scene optimization, an emerging trend is to generate novel views from sparse views or even a single image in a feed-forward manner [Chen et al. 2025; Szymanowicz et al. 2024a]. Due to the limited information from single/sparse views, such an NVS task is highly ill-posed and requires comprehensive scene understanding, including geometry, texture, and occlusion. Early works have proposed several techniques to tackle this challenging task, *e.g.*, GAN-based inpainting for disocclusion regions [Wiles et al. 2020], neural scene prediction [Yu et al. 2021], and implicit 3D transformers [Rombach et al. 2021]. In addition, various 3D representations are designed for efficient view synthesis, such as density fields [Wimbauer et al. 2023], multi-plane images [Han et al. 2022], and layered depth images [Shih et al. 2020]. Despite significant progress being achieved, previous methods are often confined to specific domains and struggle to generalize to complex in-the-wild scenes. This usually arises from the limited prior knowledge of the 3D world.

Recent advancements in generative diffusion models have shown promising performance in a variety of 3D vision tasks [Ke et al. 2024; Zhang et al. 2024], including novel view synthesis [Gu et al. 2025; You et al. 2025]. In order to generate high-quality images/videos across a wide array of domains, diffusion models are generally trained over internet-scale datasets, gaining rich prior knowledge of the visual world [Rombach et al. 2022; Xing et al. 2025]. Benefiting from this, generative diffusion models exhibit outstanding performance in generating geometry-consistent content, naturally fitting the requirements of novel view synthesis. Many works are devoted to repurposing diffusion models for high-quality NVS, like semantic-preserving generative warping [Seo et al. 2024] and point-conditioned video diffusion [Yu et al. 2024]. However, because of the generative nature, diffusion models often introduce hallucinated contents, such as different textures, when generating novel views. Consequently, existing diffusion-based NVS methods usually struggle with texture hallucination, failing to preserve the original appearance present in the input view (*e.g.*, Fig. 1).

Another popular trend is to render novel views with splatting-based approaches, *e.g.*, Gaussian splatting [Chen et al. 2025; Xu et al. 2024]. For example, Flash3D employs a zero-shot depth estimator to predict the 3D Gaussian position and directly estimates the parameters of the 3D Gaussian for novel view rendering [Szymanowicz et al. 2024a]. By enforcing the appearance consistency with pixel-/feature-level constraints, splatting-based NVS methods generally preserve better textures than diffusion-based approaches. However, since single or sparse observations provide only limited cues for the scene geometry, splatting-based methods often suffer from splatting errors, *e.g.*, misalignment due to inaccurate depth, resulting in novel views with distorted geometry (*e.g.*, Fig. 1).

We present **SplatDiff**, a video diffusion model guided by pixel splatting, designed to leverage the strengths of splatting and diffusion for high-fidelity novel view synthesis. The motivations behind our designs are as follows: (i) *Pixel Splatting:* Compared with the popular 3D Gaussian splatting techniques, we found that the simple pixel splatting, such as forward warping [Niklaus and Liu 2020], better preserves appearance under single or sparse input views. While 3D Gaussians can theoretically model complex visual effects, *e.g.*, view-dependent appearance, estimating accurate Gaussian parameters (such as opacity) from limited observations remains a significant challenge. In addition, the estimation errors of Gaussian parameters often result in artifacts and cloudy effects in novel views, yielding worse visual results than pixel splatting (*e.g.*, see Fig. 8). (ii) *Video Diffusion:* Our video diffusion model is designed to synthesize consistent and high-fidelity novel views with the guidance of splatted results. When input observations are limited, the splatted views often exhibit disocclusion regions that vary across different viewpoints. By training on large-scale video datasets, video diffusion models gain a deep understanding of visual elements such as geometry and texture. Leveraging this video diffusion prior, we synthesize realistic and consistent contents across varying viewpoints.

To achieve consistent geometry and high-fidelity texture, we incorporate two key components in SplatDiff: a novel strategy to create training pairs for *aligned synthesis* and a *texture bridge* to inject texture details into the diffusion decoder. Specifically, we first fine-tune the pre-trained video diffusion model with Training Pair Alignment (TPA) and Splatting Error Simulation (SES) for aligned synthesis. TPA enforces the geometry and brightness consistency between the splatted and generated views, enabling precise control of target viewpoints. Meanwhile, our video diffusion model learns to eliminate splatting errors (*e.g.*, flying pixels in Fig. 4b) with SES, generating aligned novel views with consistent geometry. To tackle texture hallucination, we propose a texture bridge that aggregates the features from the splatted views for texture preservation. Additionally, a texture degradation strategy is introduced to facilitate the adaptive fusion of splatted views and diffusion outputs for high-quality synthesis. In summary, our main contributions are:

- We introduce SplatDiff, a pixel-splatting-guided video diffusion model for synthesizing novel views with consistent geometry and high-fidelity texture from a single image.
- An aligned synthesis method to enable precise control of novel views while maintaining consistent geometry. In addition, we design a texture bridge module to achieve high-fidelity synthesis through adaptive feature fusion.
- SplatDiff excels in single-view NVS, sparse-view NVS, and stereo video conversion, demonstrating remarkable cross-domain and cross-task performance with training only on the single-view NVS task, as illustrated in Fig. 1.

## 2 Related Work

### 2.1 Feed-Forward Novel View Synthesis

A significant number of attempts are devoted to synthesizing novel views from single/sparse observations in a feed-forward manner. Due to the limited input information, early works usually adopt depth estimation methods to model scene geometry and then utilize

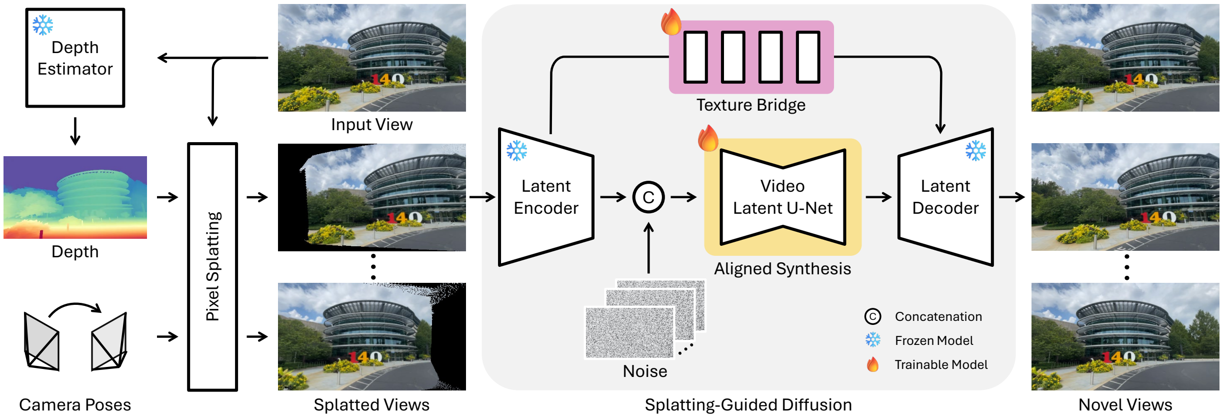


Fig. 2. **Overview of SplatDiff.** Given the input view, we first estimate the depth information from a depth estimator and then perform pixel splatting to generate splatted views as diffusion conditioning. In our splatting-guided diffusion, we fine-tune the latent U-Net for aligned synthesis, producing consistent novel views while correcting splatting errors. Meanwhile, a texture bridge module is designed to aggregate the encoder features for high-fidelity synthesis. Images credited to [Ling et al. 2024].

inpainting approaches for content synthesis [Rockwell et al. 2021; Rombach et al. 2021; Wiles et al. 2020]. To generate realistic novel views, several techniques are developed, including GAN-based inpainting [Wiles et al. 2020], VQ-VAE outpainter [Rockwell et al. 2021], and implicit 3D transformer [Rombach et al. 2021]. Recently, novel scene representations are proposed to achieve high-quality view synthesis. For instance, pixelNeRF combines convolutional networks with NeRF representation to render novel views from two images [Yu et al. 2021]. Meanwhile, layer-based representations, *e.g.*, multi-plane images (MPI) [Han et al. 2022; Khan et al. 2023; Li et al. 2021; Tucker and Snavely 2020] and layered depth images (LDI) [Jiang et al. 2023; Shih et al. 2020], are exploited for efficient rendering. However, since previous feed-forward NVS methods are mainly designed for specific domains, they often suffer from performance drops in complex scenes due to the limited model capability.

## 2.2 Diffusion-Based Novel View Synthesis

Diffusion models have demonstrated exceptional performance in generating realistic images and videos [Rombach et al. 2022], reflecting a profound understanding of the 3D world. To utilize the diffusion prior for novel view synthesis, previous attempts develop conditional diffusion frameworks, *e.g.*, 3D feature-conditioned diffusion [Chan et al. 2023] and viewpoint-conditioned diffusion [Liu et al. 2023], to generate novel views for simple inputs like 3D objects [Zheng and Vedaldi 2024]. Considering complex real-world scenes, multi-view diffusion models are often employed to synthesize high-quality novel views, which are then used to generate 3D scenes (*e.g.*, 3D Gaussians) for rendering [Liu et al. 2024; Wu et al. 2024]. Based on this, ZeroNVS combines diverse training datasets to acquire zero-shot NVS performance [Sargent et al. 2024], and Cat3D designs an efficient parallel sampling strategy for fast generation of 3D-consistent images [Gao et al. 2024]. In addition, GenWarp exploits the diffusion prior to achieve semantic-preserving warping [Seo et al. 2024]. Recent works also explore the potential of video diffusion models for novel view synthesis [Bian et al. 2025; Liang et al. 2024]. For instance, ViewCrafter constructs a point-conditioned diffusion model to iteratively complete the point cloud for consistent view rendering [Yu et al. 2024], and StereoCrafter proposes a tiled processing strategy to generate stereoscopic videos with video diffusion [Zhao et al. 2024]. While diffusion-based NVS approaches excel at synthesizing realistic novel views, the generative nature of diffusion models often introduces hallucinated content (*e.g.*, Fig. 1), leading to inconsistent texture across different viewpoints.

## 2.3 Splatting-Based Novel View Synthesis

Splatting-based NVS approaches are typically trained in a regression manner with pixel-level or feature-level constraints [Zhang et al. 2018]. As a result, they often preserve better textures compared to diffusion-based methods. Previous study employs depth-based warping to achieve real-time rendering [Cao et al. 2022]. With the advancement of 3DGS techniques [Kerbl et al. 2023], a considerable amount of attention has been drawn to feed-forward Gaussian splatting methods. The pioneer work pixelSplat estimates Gaussian parameters from neural networks and dense probability distributions, achieving efficient novel view synthesis from two images [Charatan et al. 2024]. Following this, several techniques are developed for improved performance and efficiency, including cost volume encoding [Chen et al. 2025] and depth-aware transformer [Zhang et al. 2025]. Recent method DepthSplat integrates monocular features from depth models and achieves better geometry in the estimated 3D Gaussians [Xu et al. 2024]. Another line of work focuses on predicting Gaussian parameters from a single image. Splatter Image obtains 3D Gaussian parameters from pure image features [Szymanowicz et al. 2024b], and Flash3D employs zero-shot depth models for generalizable single-view NVS [Szymanowicz et al. 2024a]. However, due to the challenges of estimating accurate geometry from limited observations, existing splatting-based methods often suffer from splatting errors, resulting in novel views with distorted geometry (*e.g.*, Fig. 1). By contrast, our SplatDiff leverages

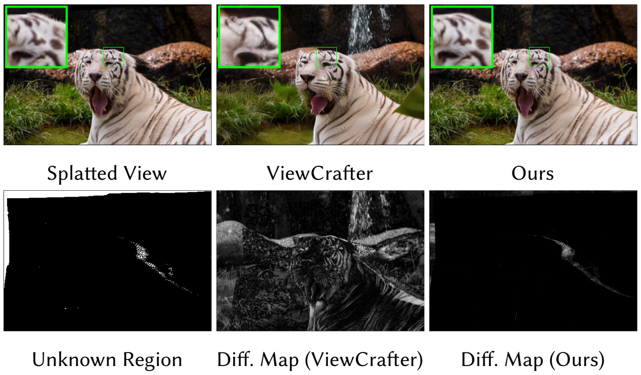


Fig. 3. **Misalignment.** The difference map shows the absolute difference between the splatted view and the generated view. Diffusion-based methods, *e.g.*, ViewCrafter, often generate misaligned contents in novel views, resulting in significant differences across the image. In contrast, our SplatDiff is faithful to inputs and shows differences mainly around the unknown region.

the geometric priors of diffusion models to correct splatting errors, achieving geometry-consistent and high-fidelity view synthesis.

## 3 Splatting-Guided Diffusion

As depicted in Fig. 2, SplatDiff synthesizes novel views from a single image in a feed-forward manner. Given the input image, we predict the depth using an off-the-shelf depth estimator and then perform pixel splatting according to the camera poses. Since the splatted views often contain splatting errors and unknown regions, *e.g.*, disocclusion, we leverage the video diffusion prior and propose splatting-guided diffusion to refine the splatted results. Specifically, we propose a fine-tuning strategy to synthesize geometry-aligned contents while correcting the distortions caused by splatting errors. To tackle texture hallucination, we design a texture bridge to aggregate encoder features for high-fidelity novel view synthesis. In the following sections, we first provide a brief introduction to pixel splatting and video diffusion models in Sec. 3.1. The designs of the aligned synthesis strategy and the texture bridge module are then presented in Secs. 3.2 and 3.3, respectively.

### 3.1 Preliminaries

*Pixel Splatting.* Given a transformation map, *e.g.*, optical flow, pixel splatting projects pixels from the input image to the target image, which has been widely used in computer vision tasks like frame interpolation [Niklaus and Liu 2020]. To generate novel views, we compute the view transformation map with camera poses and the estimated depth. Similar to prior works [Mehl et al. 2024; Niklaus and Liu 2020], we employ softmax splatting to resolve splatting collisions and assign pixel importance based on the depth information. This approach gives higher blending weights to foreground objects, preserving the geometric layout of the input view.

*Video Diffusion.* Diffusion models typically consist of a forward process and a reverse process [Ho et al. 2020; Song et al. 2021]. The forward process $q(\mathbf{x}_t|\mathbf{x}_0, t)$ converts data $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x})$ into Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ by gradually adding noise at each step $t \in \{1, \ldots, T\}$, and the learned reverse process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, t)$ transforms random Gaussian noise to a new sample with a denoising network $\epsilon_\theta$. At each denoising step, the network $\epsilon_\theta$ is supervised by

$$\min_\theta \mathbb{E}_{\mathbf{x}\sim p_{\text{data}},\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t\sim\mathcal{U}(T)} \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|_2^2, \tag{1}$$

where $\epsilon$ is the Gaussian noise, and $\mathbf{x}_t$ denotes the noisy sample at step $t$. By learning to estimate the added noise with $\epsilon_\theta$, diffusion models are able to generate new data samples via iterative denoising.

We build SplatDiff upon latent video diffusion models to balance performance and computational complexity. Given $L$ splatted views, we encode each view with a latent encoder and generate latent representations $\mathbf{Z} \in \mathbb{R}^{L\times C\times h\times w}$ as conditioning. The forward and reverse processes are performed in the latent space for training. During inference, we start with Gaussian noise $\epsilon \in \mathbb{R}^{L\times C\times h\times w}$ and use the latents of all pixel-splatted views $\mathbf{Z}$ as conditioning. All target novel views are generated simultaneously by decoding the denoised latent codes to the image space. Based on this pipeline, we propose the aligned synthesis strategy and the texture bridge module to achieve high-fidelity novel view synthesis.

### 3.2 Aligned Synthesis

Previous diffusion-based NVS approaches often generate textures and geometry that are misaligned with the conditioning (Fig. 3), making precise control of target viewpoints challenging. This misalignment typically stems from the use of unaligned pairs during diffusion training. Given the depth $\mathbf{d}$ and the camera poses $\mathbf{p}$, naive training methods often construct training pairs $\{\mathbf{v}_{\text{tgt}}, \mathbf{x}_{\text{tgt}}\}$ with

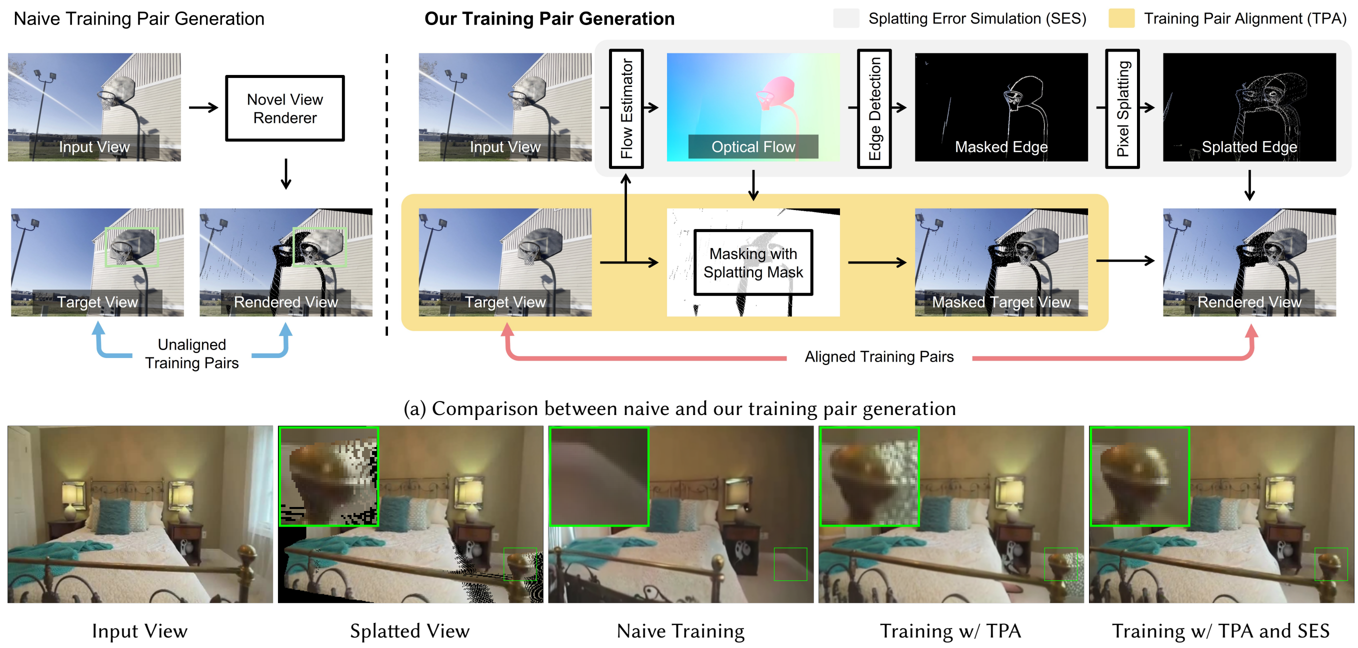
$$\mathbf{v}_{\text{tgt}} = \text{Render}(\mathbf{x}_{\text{src}}, \mathbf{d}, \mathbf{p}), \tag{2}$$

where Render$(\cdot)$ denotes a novel view renderer, *e.g.*, point cloud renderer [Yu et al. 2024]. $\mathbf{x}_{\text{src}}$, $\mathbf{x}_{\text{tgt}}$, and $\mathbf{v}_{\text{tgt}}$ correspond to the source input view, the target view, and the rendered view, respectively. Then, the diffusion model is trained to predict $\mathbf{x}_{\text{tgt}}$ conditioned on $\mathbf{v}_{\text{tgt}}$. However, the conditioning $\mathbf{v}_{\text{tgt}}$ often shows different texture and geometry with the target view $\mathbf{x}_{\text{tgt}}$ (*e.g.*, Fig. 4a) due to several factors, *e.g.*, different lighting and depth estimation error. This renders the diffusion model to produce misaligned novel views with the conditioned view, as shown in Fig. 4b. To this end, we propose the Training Pair Alignment (TPA) strategy for aligned synthesis.

*Training Pair Alignment.* Instead of generating the diffusion conditioning from the input view $\mathbf{x}_{\text{src}}$, TPA utilizes the target view $\mathbf{x}_{\text{tgt}}$ to construct aligned training pairs. As illustrated in Fig. 4a, we first estimate the view transformation map from $\mathbf{x}_{\text{src}}$ and $\mathbf{x}_{\text{tgt}}$ with an optical flow estimator [Xu et al. 2023]. With the estimated flow, we then generate a splatting mask $\mathbf{m}_{\text{splat}}$ via pixel splatting, where $\mathbf{m}_{\text{splat}}$ indicates the valid splatting regions. Finally, we construct the aligned training pairs $\{\tilde{\mathbf{v}}_{\text{tgt}}, \mathbf{x}_{\text{tgt}}\}$ by masking the target view, *i.e.*,

$$\tilde{\mathbf{v}}_{\text{tgt}} = \mathbf{x}_{\text{tgt}} \odot \mathbf{m}_{\text{splat}}. \tag{3}$$

Since $\tilde{\mathbf{v}}_{\text{tgt}}$ aligns with $\mathbf{x}_{\text{tgt}}$ in both geometry and texture, the diffusion model trained with TPA learns to adhere to the conditioned view, producing consistent novel views (*e.g.*, Fig. 4b). However, due to the existence of splatting errors in real splatted views, the model trained solely with TPA tends to be misled by the splatting errors, resulting in artifacts as shown in the green box in Fig. 4b. To address this, we



(a) Comparison between naive and our training pair generation

(b) Results of the proposed training pair alignment (TPA) and splatting error simulation (SES)

Fig. 4. **Aligned Synthesis.** Naive training pair generation often produces pairs that are locally unaligned in geometry and texture (green box in (a)), which results in unaligned novel view synthesis (naive training in (b)). By masking the target view, the proposed training pair alignment (TPA) enables aligned synthesis, but the generated contents tend to be misled by the splatting errors in the splatted view (training with TPA in (b)). Combining TPA with splatting error simulation (SES) in training, our SplatDiff learns to handle splatting errors and generates geometry- and texture-aligned novel views. Images credited to [Ling et al. 2024] and [Zhou et al. 2018].

propose the Splatting Error Simulation (SES) method to enhance the model's robustness against splatting errors.

*Splatting Error Simulation.* Splatting errors in pixel splatting often appear as flying pixels, *e.g.*, green box in Fig. 4b, which typically arise from inaccuracies in the transformation map, *e.g.*, blurred depth discontinuities [Shih et al. 2020]. As a result, pixels around object boundaries might be projected to incorrect positions in the target view, leading to distorted geometry with flying pixels. To resolve this, we propose to simulate splatting errors in the training pairs (Fig. 4a). Since the flying pixels often stem from the object boundaries, we first generate an edge mask $\mathbf{m}_{\text{edge}}$ from the transformation map using a Sobel operator. Next, we extract the edge regions $\mathbf{e}_{\text{src}}$ from the input view $\mathbf{x}_{\text{src}}$ by $\mathbf{e}_{\text{src}} = \mathbf{x}_{\text{src}} \odot \mathbf{m}_{\text{edge}}$. The extracted edge regions $\mathbf{e}_{\text{src}}$ are then splatted to the target view using the optical flow to generate the splatted edge $\mathbf{e}_{\text{tgt}}$. Finally, we simulate the splatting errors in $\tilde{\mathbf{v}}_{\text{tgt}}$ by random perturbation:

$$\hat{\mathbf{v}}_{\text{tgt}} = \mathbf{e}_{\text{tgt}} \odot \mathbf{m}_{\text{error}} + \tilde{\mathbf{v}}_{\text{tgt}} \odot (1 - \mathbf{m}_{\text{error}}), \tag{4}$$

where $\hat{\mathbf{v}}_{\text{tgt}}$ denotes the rendered view with simulated splatting errors, and $\mathbf{m}_{\text{error}}$ is a randomly generated mask indicating where to introduce splatting errors. By using $\{\hat{\mathbf{v}}_{\text{tgt}}, \mathbf{x}_{\text{tgt}}\}$ as training pairs, our diffusion model learns to correct splatting errors by utilizing its rich geometric prior, while maintaining aligned synthesis (Fig. 4b).

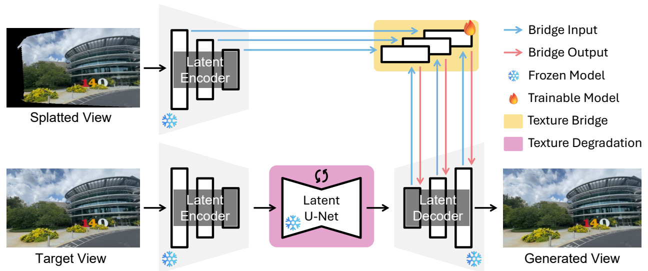


Fig. 5. **Texture bridge.** The texture bridge is designed to complement multi-scale features from the encoder to the decoder, enabling high-fidelity synthesis. For training, we intentionally degrade the texture of the target view with the diffusion model, simulating a degraded view with hallucinated textures. This facilitates the texture bridge to learn how to adaptively fuse information from the splatted view and the diffusion outputs. Images credited to [Ling et al. 2024].

## 3.3 Texture Bridge

Although the diffusion model can generate geometry-consistent novel views with our aligned synthesis strategy, its outputs often contain hallucinated textures that are inconsistent with the input view (Fig. 1). One potential solution is to preserve the texture of the input image by utilizing the splatted view. However, directly blending the splatted view and the diffusion output faces several

Table 1. **Quantitative evaluation of single-view novel view synthesis.** * denotes methods with two input views. Best and second-best results are marked.

| Method | RealEstate10K Dataset | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Easy Set | | | | | Hard Set | | | | |
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | FID ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | FID ↓ |
| Syn-Sin | - | - | - | - | - | 22.30 | 0.740 | - | - | - |
| SV-MPI | 27.10 | 0.870 | - | - | - | 23.52 | 0.785 | - | - | - |
| BTS | - | - | - | - | - | 24.00 | 0.755 | 0.194 | - | - |
| Splatter Image | 28.15 | 0.894 | 0.110 | - | - | 24.15 | 0.810 | 0.177 | - | - |
| MINE | 28.45 | 0.897 | 0.111 | - | - | 24.75 | 0.820 | 0.179 | - | - |
| AdaMPI | 28.03 | 0.892 | 0.104 | - | - | 23.54 | 0.809 | 0.184 | - | - |
| GenWarp | 16.94 | 0.519 | 0.318 | 0.137 | 12.27 | 16.05 | 0.488 | 0.356 | 0.149 | 12.66 |
| ViewCrafter | 20.61 | 0.705 | 0.242 | 0.139 | 13.71 | 16.64 | 0.588 | 0.347 | 0.185 | 18.30 |
| Flash3D | 28.46 | 0.899 | 0.100 | 0.062 | 4.55 | 24.93 | 0.833 | 0.160 | 0.098 | 8.42 |
| SplatDiff (Ours) | 28.53 | 0.895 | 0.096 | 0.059 | 3.96 | 25.17 | 0.820 | 0.154 | 0.088 | 6.04 |

| Method | DL3DV-10K Dataset | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Easy Set | | | | | Hard Set | | | | |
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | FID ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | FID ↓ |
| GenWarp | 17.72 | 0.463 | 0.309 | 0.357 | 77.31 | 16.74 | 0.409 | 0.350 | 0.352 | 81.58 |
| Diffusion as Shader | 17.19 | 0.505 | 0.363 | 0.150 | 62.07 | 16.38 | 0.542 | 0.457 | 0.190 | 89.55 |
| NVS-Solver | 19.97 | 0.660 | 0.309 | 0.155 | 81.68 | 18.47 | 0.660 | 0.374 | 0.183 | 100.42 |
| ViewCrafter | 23.21 | 0.694 | 0.182 | 0.349 | 46.60 | 19.77 | 0.565 | 0.249 | 0.349 | 59.56 |
| DepthSplat* | 24.37 | 0.790 | 0.168 | 0.109 | 52.01 | 21.79 | 0.709 | 0.221 | 0.124 | 61.37 |
| SplatDiff (Ours) | 26.14 | 0.826 | 0.113 | 0.068 | 24.23 | 22.48 | 0.732 | 0.181 | 0.092 | 41.22 |

challenges: (i) the splatted view usually contains aliasing artifacts, such as jagged edges, and may exhibit blurred details due to the resampling process, which can degrade the quality of the novel view, and (ii) detecting and removing splatting errors in the splatted view is crucial for achieving high-quality NVS. Thus, we propose the texture bridge module to adaptively fuse the splatted view with the diffusion output for high-fidelity synthesis. As depicted in Fig. 5, the texture bridge consists of a fusion block at each feature scale. Let $\mathbf{f}^i_{\text{enc}}$ denote the $i$-th scale encoder feature extracted from the splatted view, and let $\mathbf{f}^i_{\text{dec}}$ represent the $i$-th scale decoder feature. We first fuse the features at each scale with the texture bridge, *i.e.*,

$$\mathbf{f}^i_{\text{fuse}} = \text{Fuse}_i(\mathbf{f}^i_{\text{enc}}\,,\ \mathbf{f}^i_{\text{dec}}), \tag{5}$$

and then restore the novel view by passing the fused features $\{\mathbf{f}^i_{\text{fuse}}\}$ through the latent decoder. Each fusion block $\text{Fuse}_i(\cdot)$ is implemented using two residual blocks [He et al. 2016], though more advanced architecture, *e.g.*, self-attention, could be employed for better performance. With multi-scale fusion, the texture bridge effectively aggregates fine-grained features for high-quality synthesis.

*Training with Texture Degradation.* Our texture bridge is designed to adaptively select optimal features from the splatted view and the diffusion output for view synthesis. For example, the texture bridge should mainly utilize the diffusion output when encountering splatting errors, while relying more on the splatted view in the regions with texture hallucination. To achieve this, one possible training approach is to feed the texture bridge with the splatted view and the corresponding diffusion output, then supervise the decoded results using the target view. However, the diffusion outputs often differ significantly from the target view in the unknown regions (*e.g.*, disocclusion), resulting in sub-optimal training performance. To overcome this, we propose a texture degradation strategy to facilitate model training. Specifically, we employ the diffusion model to degrade the texture of the target view and train the texture bridge using the degraded view (Fig. 5). As a result, the degraded view shares similar contents to the target view while containing hallucinated textures for training. For supervision, we employ the $\ell_1$ loss $\mathcal{L}_1$ and the perceptual loss $\mathcal{L}_{\text{LPIPS}}$ [Zhang et al. 2018], *i.e.*,

$$\mathcal{L} = \mathcal{L}_1 + \alpha\mathcal{L}_{\text{LPIPS}}, \tag{6}$$

where $\alpha = 0.1$. With the texture bridge, our SplatDiff addresses texture hallucination through the adaptive fusion of the splatted view and the diffusion output. Meanwhile, the texture bridge also learns to refine the input features, *e.g.*, aliasing artifacts and blurry details in the splatted view, for high-fidelity novel view synthesis.

## 4 Experiments and Analysis

### 4.1 Experimental Settings

*Implementation Details.* We implement SplatDiff based on the open-source video diffusion model DynamiCrafter [Xing et al. 2025] with ViewCrafter weight initialization [Yu et al. 2024]. We employ the AdamW optimizer [Loshchilov and Hutter 2019] to train SplatDiff under 320 × 512 patches and batch size 16. For aligned synthesis, we only fine-tune the latent U-Net for 1.5K steps with a learning rate $1{\times}10^{-5}$. Afterward, we train the texture bridge for 10K steps with a learning rate $1{\times}10^{-4}$. The total training takes around 2.5

Table 2. **Ablation on single-view novel view synthesis.** TPA, SES, TB, and TD represent training pair alignment, splatting error simulation, texture bridge, and texture degradation. Best and second-best results are marked.

| ID | TPA | SES | TB | TD | DL3DV-10K Dataset | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | FID ↓ |
| #1 | | | | | 23.21 | 0.694 | 0.182 | 0.349 | 46.60 |
| #2 | ✓ | | | | 24.90 | 0.749 | 0.178 | 0.100 | 43.97 |
| #3 | ✓ | ✓ | | | 25.00 | 0.749 | 0.178 | 0.098 | 43.59 |
| #4 | ✓ | ✓ | ✓ | | 26.05 | 0.825 | 0.118 | 0.070 | 25.49 |
| #5 | ✓ | ✓ | ✓ | ✓ | 26.14 | 0.826 | 0.113 | 0.068 | 24.23 |

days on a single NVIDIA RTX A6000 GPU. At inference, we apply the DDIM scheduler [Song et al. 2021] with 50-step sampling.

*Datasets and Evaluation.* We use two datasets **RealEstate10K** [Zhou et al. 2018] and **DL3DV-10K** [Ling et al. 2024] for training and evaluation. For each dataset, two evaluation sets (easy and hard sets) are created with different baseline ranges. In RealEstate10K, we follow the setting in Flash3D [Szymanowicz et al. 2024a] to skip 5 and random ±30 frames for the easy and hard sets. Since DL3DV-10K features faster camera motion and more complex scenes, we skip 3 and 6 frames for the easy and hard sets. For the methods requiring depth as inputs, we use the same depth models for fair comparisons (UniDepth [Piccinelli et al. 2024] in RealEstate10K and DepthSplat [Xu et al. 2024] in DL3DV-10K). Quantitative evaluation is conducted with pixel-level metrics (PSNR and SSIM), feature-level metrics (LPIPS [Zhang et al. 2018] and DISTS [Ding et al. 2022]), and distribution-level metric FID [Heusel et al. 2017]. Since the generated novel views are often not perfectly aligned with the ground-truth due to depth estimation errors, the importance of these metrics follows the hierarchy: distribution-level > feature-level > pixel-level. In-the-wild samples are also collected for qualitative evaluation.

## 4.2 Benchmarking

Tab. 1 shows the single-view NVS results of SplatDiff compared with prior arts. Previous diffusion-based NVS approaches, *e.g.*, ViewCrafter, struggle to achieve good metrics due to the hallucinated geometry and textures as depicted in Figs. 6 and 7. Although splatting-based methods, *e.g.*, Flash3D, better preserve texture from the input view, the generated novel views often suffer from geometry distortion due to splatting errors (Fig. 6a). Compared with them, our SplatDiff produces the best novel views with consistent geometry and high-fidelity textures as shown in Fig. 6. In addition, when better depth maps are available (*e.g.*, on DL3DV-10K, where the depth is estimated from two views), SplatDiff shows significantly better performance than both diffusion-based and splatting-based approaches (Tab. 1). With our aligned synthesis strategy and texture bridge module, SplatDiff even outperforms the two-view method DepthSplat and generates better visual results with fine-grained details (Fig. 6b).

## 4.3 Ablation Study

In Tab. 2, we study the effectiveness of each design: training pair alignment (TPA), splatting error simulation (SES), texture bridge (TB), and texture degradation (TD). (i) **Aligned Synthesis**: Compared with the baseline model (#1), the model with TPA (#2) enforces the consistency between the conditioned view and the generated view, boosting the novel view synthesis performance (*e.g.*, 1.69 dB PSNR gain). With SES, the diffusion model further learns to correct splatting errors by leveraging its rich geometric prior. Consequently, the model with TPA and SES achieves aligned synthesis while preserving geometric consistency (Fig. 4b). (ii) **Texture Bridge**: To handle texture hallucination, TB aggregates the multi-scale features from the splatted view and the diffusion output, leading to significant improvements across all metrics (#4 in Tab. 2). Furthermore, training with TD improves feature fusion and refines the input features for better synthesis. As a result, the model incorporating all designs achieves the best NVS performance (#5 in Tab. 2).

Table 3. **Quantitative evaluation of sparse-view novel view synthesis.** † indicates the averaged results of the two generated novel views. The best and second-best results are marked.

| Method | DL3DV-10K Dataset (In-domain) | | | | |
|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | FID ↓ |
| MVSplat | 17.54 | 0.529 | 0.402 | - | - |
| DepthSplat | 19.05 | 0.610 | 0.313 | 0.163 | 105.09 |
| SplatDiff (Ours) | 21.42 | 0.619 | 0.294 | 0.130 | 71.26 |
| Method | DTU Dataset (Cross-domain) | | | | |
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | FID ↓ |
| pixelSplat | 12.89 | 0.382 | 0.560 | - | - |
| MVSplat | 13.94 | 0.473 | 0.385 | - | - |
| TranSplat | 14.93 | 0.531 | 0.326 | - | - |
| DepthSplat | 16.01 | 0.612 | 0.334 | 0.201 | 130.75 |
| SplatDiff (Ours) | 15.96 | 0.590 | 0.264 | 0.147 | 82.45 |
| SplatDiff† (Ours) | 16.33 | 0.616 | 0.285 | 0.164 | 98.98 |

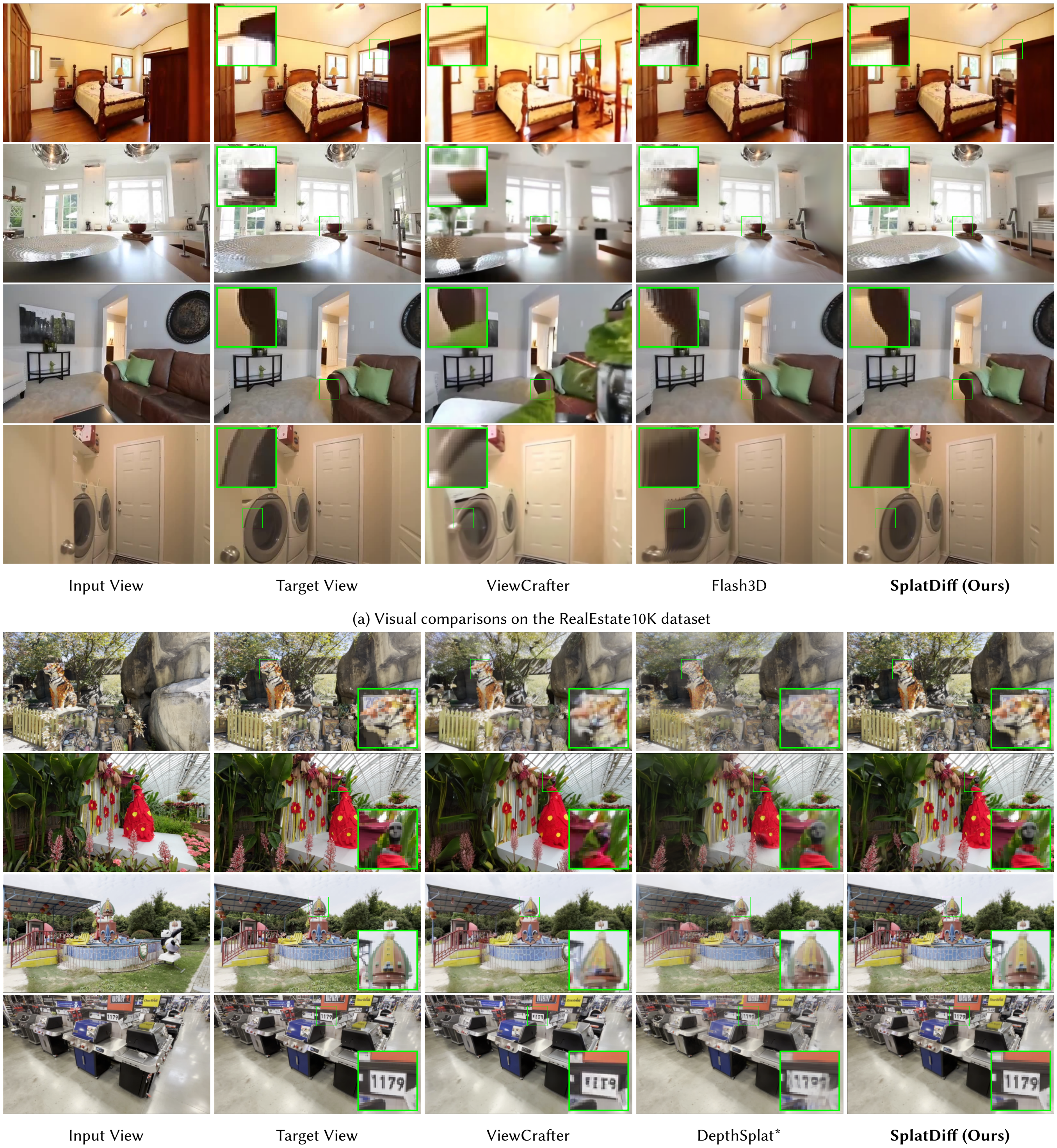
Table 4. **Quantitative evaluation of stereo video conversion.** The best and second-best results are marked.

| Method | Spring Dataset | | | | |
|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | DISTS ↓ | FID ↓ |
| ViewCrafter | 18.41 | 0.526 | 0.266 | 0.187 | 143.31 |
| StereoCrafter | 26.46 | 0.765 | 0.192 | 0.175 | 135.86 |
| SplatDiff (Ours) | 30.12 | 0.908 | 0.073 | 0.070 | 46.37 |

## 4.4 Applications

We directly apply the SplatDiff pre-trained on single-view NVS to different tasks, *i.e.*, sparse-view NVS and stereo video conversion. By simply modifying the model inputs, SplatDiff demonstrates remarkable cross-domain performance as shown in Tabs. 3 and 4.

*Sparse-View Novel View Synthesis.* We evaluate the sparse-view NVS performance with two input views. Given the two splatted views, we simply select the one with fewer unknown regions as the primary input and use the other one to fill the unknown regions with a blurred blending mask. We follow DepthSplat to conduct



(a) Visual comparisons on the RealEstate10K dataset

(b) Visual comparisons on the DL3DV-10K dataset

Fig. 6. **Qualitative comparisons on the single-view novel view synthesis task.** * indicates that the method uses two views as input. Images credited to [Ling et al. 2024] and [Zhou et al. 2018].

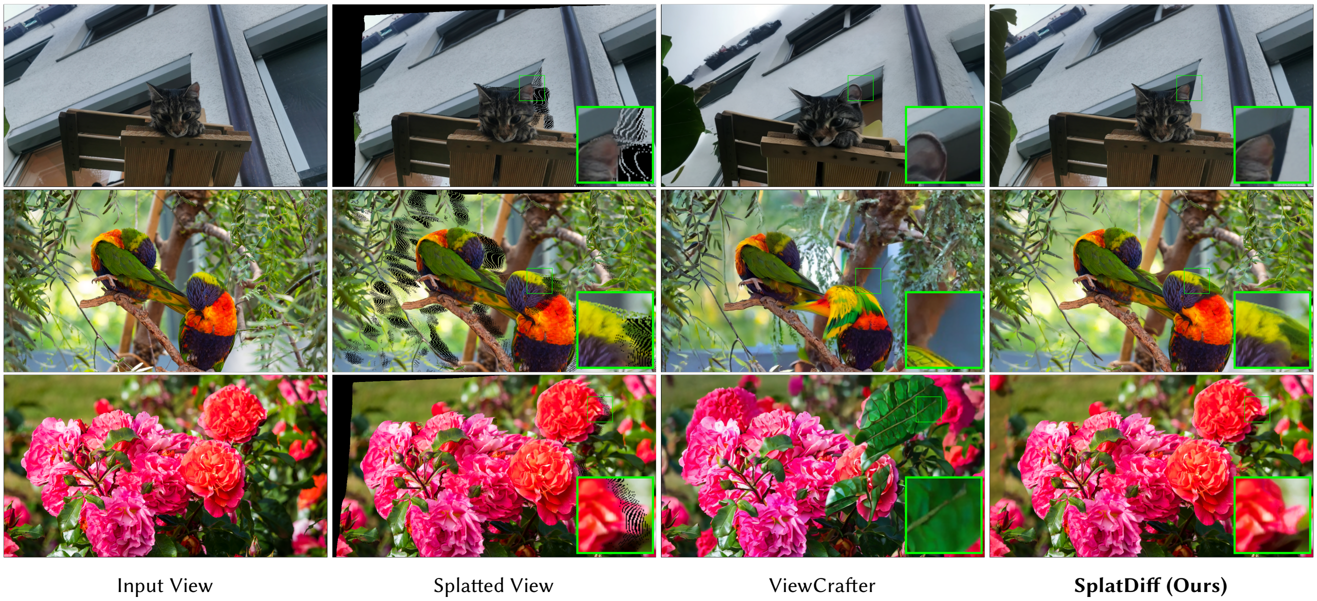
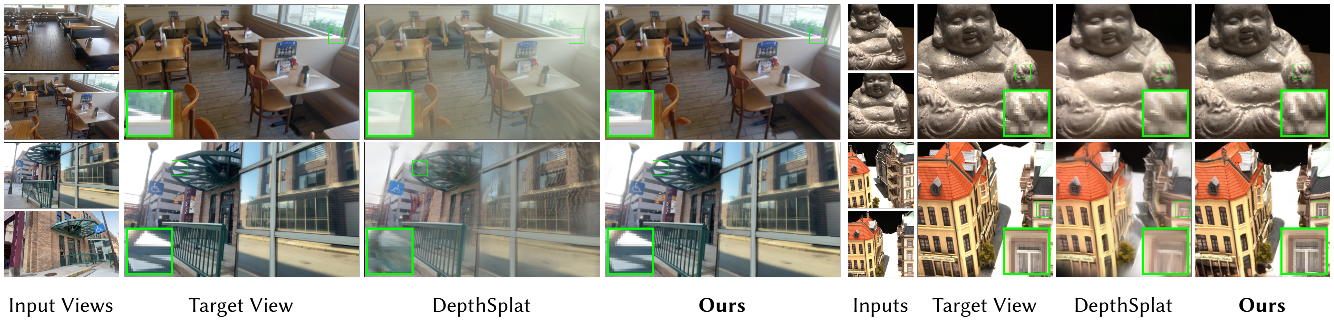


Fig. 7. **Qualitative comparisons on in-the-wild high-resolution (576×1024) samples.**



(a) Visual comparisons on the DL3DV-10K dataset

(b) Visual comparisons on the DTU dataset

Fig. 8. **Sparse-view novel view synthesis results on the DL3DV-10K (in-domain) and DTU (cross-domain) datasets.** Images credited to [Zhou et al. 2018] and [Jensen et al. 2014].

the in-domain evaluation on the DL3DV-10K dataset. Under limited observations, it is challenging to estimate accurate parameters for 3D Gaussians, which often leads to cloudy results in novel views (Fig. 8a). With pixel splatting, SplatDiff achieves state-of-the-art performance as shown in Tab. 3. We also follow MVSplat to conduct the cross-domain evaluation on the DTU dataset [Jensen et al. 2014]. Due to the domain gap, the estimated depth is less accurate, leading to misalignment between the splatted view and the target view. In such cases, Gaussian splatting approaches, *e.g.*, DepthSplat, often blur details to handle the misalignment, resulting in slightly higher pixel-level metrics in Tab. 3. By contrast, our SplatDiff achieves significantly better visual quality with fine-grained details (Fig. 8b). To mitigate the misalignment in SplatDiff, we additionally average the novel views generated from the two splatted views, which outperform DepthSplat across all metrics (SplatDiff$^{\dagger}$ in Tab. 3).

*Stereo Video Conversion.* Stereo video conversion is widely used in movie production, and thus we employ the Spring dataset [Mehl et al. 2023] for evaluation, which features high-resolution stereo videos from the Blender movie "Spring". We generate the splatted right-eye views for SplatDiff from the input left-eye videos, and the ground-truth disparity is used in all methods for fair comparisons. As shown in Fig. 9, it is challenging for ViewCrafter to handle dynamic scenes, resulting in significantly different contents in the novel views. Although StereoCrafter generates more consistent novel views with the inputs, the synthesized views often suffer from texture hallucination and blurry details. Benefiting from the aligned synthesis strategy, our SplatDiff can be directly applied to dynamic videos without additional training. Meanwhile, the proposed texture bridge preserves high-fidelity details from inputs, achieving the best stereo video conversion performance (Tab. 4 and Fig. 9).

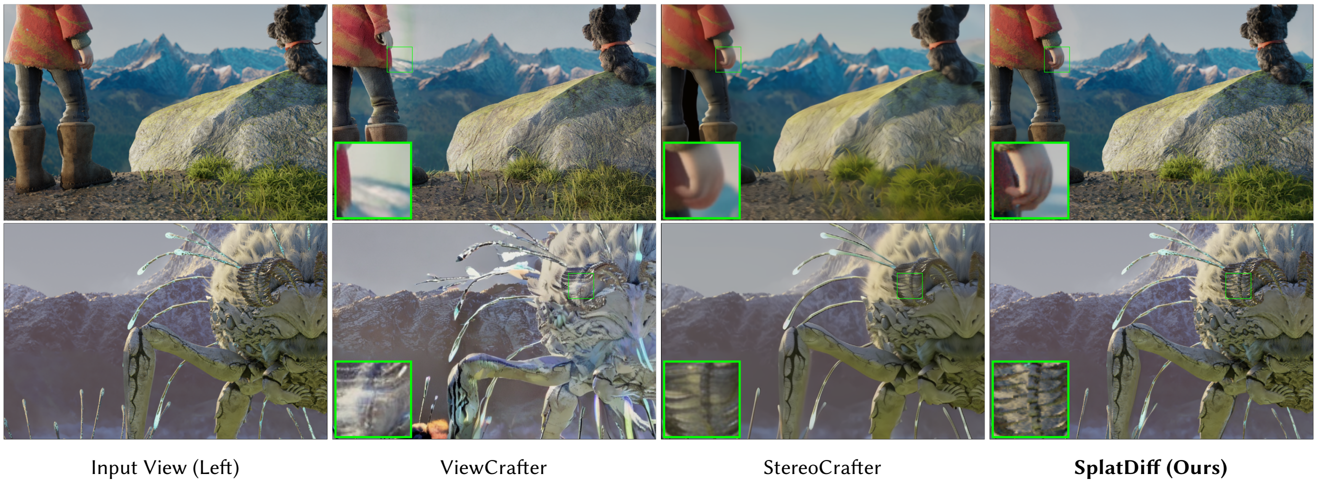


Fig. 9. **Stereo video conversion results on the Spring dataset.** Right-eye views are synthesized based on the input left-eye views. Images credited to [Mehl et al. 2023].

## 5 Conclusion

We propose SplatDiff, a pixel-splatting-guided video diffusion model designed for geometry-consistent and high-fidelity novel view synthesis. With the aligned synthesis strategy, SplatDiff achieves precise control of target viewpoints while effectively correcting geometry distortion caused by splatting errors. In addition, the proposed texture bridge recovers high-fidelity texture via adaptive feature fusion. Extensive experiments across single-view novel view synthesis, sparse-view novel view synthesis, and stereo video conversion verify the versatility and state-of-the-art performance of SplatDiff.

## References

Weikang Bian, Zhaoyang Huang, Xiaoyu Shi, Yijin Li, Fu-Yun Wang, and Hongsheng Li. 2025. GS-DiT: Advancing Video Generation with Pseudo 4D Gaussian Fields through Efficient Dense 3D Point Tracking. *arXiv preprint arXiv:2501.02690* (2025).

Ang Cao, Chris Rockwell, and Justin Johnson. 2022. Fwd: Real-time novel view synthesis with forward warping and depth. In *CVPR*. 15713–15724.

Eric R Chan, Koki Nagano, Matthew A Chan, Alexander W Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. 2023. Generative novel view synthesis with 3d-aware diffusion models. In *ICCV*. 4217–4229.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. 2024. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*. 19457–19467.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. 2025. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *ECCV*. Springer, 370–386.

Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. 2022. Image quality assessment: Unifying structure and texture similarity. *IEEE TPAMI* 44, 5 (2022), 2567–2581. https://doi.org/10.1109/TPAMI.2020.3045810

Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. 2024. Cat3d: Create anything in 3d with multi-view diffusion models. In *NeurIPS*.

Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, et al. 2025. Diffusion as Shader: 3D-aware Video Diffusion for Versatile Video Generation Control. *arXiv preprint arXiv:2501.03847* (2025).

Yuxuan Han, Ruicheng Wang, and Jiaolong Yang. 2022. Single-View View Synthesis in the Wild with Learned Adaptive Multiplane Images. In *ACM SIGGRAPH 2022 Conference Proceedings* (Vancouver, BC, Canada) *(SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, Article 14, 8 pages. https://doi.org/10.1145/3528233.3530755

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS* 30 (2017).

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *NeurIPS*, Vol. 33. 6840–6851.

Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. 2014. Large scale multi-view stereopsis evaluation. In *CVPR*. 406–413.

Yutao Jiang, Yang Zhou, Yuan Liang, Wenxi Liu, Jianbo Jiao, Yuhui Quan, and Shengfeng He. 2023. Diffuse3D: Wide-Angle 3D Photography via Bilateral Diffusion. In *ICCV*. 8998–9008.

Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. 2024. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*. 9492–9502.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.* 42, 4 (July 2023). https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

Numair Khan, Lei Xiao, and Douglas Lanman. 2023. Tiled multiplane images for practical 3d photography. In *ICCV*. 10454–10464.

Jiaxin Li, Zijian Feng, Qi She, Henghui Ding, Changhu Wang, and Gim Hee Lee. 2021. Mine: Towards continuous depth mpi with nerf for novel view synthesis. In *ICCV*. 12578–12588.

Hanwen Liang, Junli Cao, Vidit Goel, Guocheng Qian, Sergei Korolev, Demetri Terzopoulos, Konstantinos N Plataniotis, Sergey Tulyakov, and Jian Ren. 2024. Wonderland: Navigating 3D Scenes from a Single Image. *arXiv preprint arXiv:2412.12091* (2024).

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. 2024. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*. 22160–22169.

Fangfu Liu, Wenqiang Sun, Hanyang Wang, Yikai Wang, Haowen Sun, Junliang Ye, Jun Zhang, and Yueqi Duan. 2024. Reconx: Reconstruct any scene from sparse views with video diffusion model. *arXiv preprint arXiv:2408.16767* (2024).

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. 2023. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*. 9298–9309.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *ICLR*.

Lukas Mehl, Andrés Bruhn, Markus Gross, and Christopher Schroers. 2024. Stereo Conversion with Disparity-Aware Warping, Compositing and Inpainting. In *WACV*. 4260–4269.

Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. 2023. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *CVPR*. 4981–4991.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

Simon Niklaus and Feng Liu. 2020. Softmax splatting for video frame interpolation. In *CVPR*. 5437–5446.

Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. 2024. UniDepth: Universal Monocular Metric Depth Estimation. In *CVPR*. 10106–10116.

Chris Rockwell, David F Fouhey, and Justin Johnson. 2021. Pixelsynth: Generating a 3d-consistent experience from a single image. In *ICCV*. 14104–14113.

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

Robin Rombach, Patrick Esser, and Björn Ommer. 2021. Geometry-free view synthesis: Transformers and no 3d priors. In *ICCV*. 14356–14366.

Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, et al. 2024. ZeroNVS: Zero-Shot 360-Degree View Synthesis from a Single Image. In *CVPR*. 9420–9429.

Junyoung Seo, Kazumi Fukuda, Takashi Shibuya, Takuya Narihira, Naoki Murata, Shoukang Hu, Chieh-Hsin Lai, Seungryong Kim, and Yuki Mitsufuji. 2024. GenWarp: Single Image to Novel Views with Semantic-Preserving Generative Warping. In *NeurIPS*.

Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 2020. 3d photography using context-aware layered depth inpainting. In *CVPR*. 8028–8038.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising diffusion implicit models. In *ICLR*.

Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, João F Henriques, Christian Rupprecht, and Andrea Vedaldi. 2024a. Flash3D: Feed-Forward Generalisable 3D Scene Reconstruction from a Single Image. *arXiv preprint arXiv:2406.04343* (2024).

Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. 2024b. Splatter image: Ultra-fast single-view 3d reconstruction. In *CVPR*. 10208–10217.

Richard Tucker and Noah Snavely. 2020. Single-view view synthesis with multiplane images. In *CVPR*. 551–560.

Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. 2020. Synsin: End-to-end view synthesis from a single image. In *CVPR*. 7467–7477.

Felix Wimbauer, Nan Yang, Christian Rupprecht, and Daniel Cremers. 2023. Behind the scenes: Density fields for single view reconstruction. In *CVPR*. 9076–9086.

Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. 2024. Reconfusion: 3d reconstruction with diffusion priors. In *CVPR*. 21551–21561.

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

Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. 2024. Depthsplat: Connecting gaussian splatting and depth. *arXiv preprint arXiv:2410.13862* (2024).

Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. 2023. Unifying flow, stereo and depth estimation. *IEEE TPAMI* (2023).

Meng You, Zhiyu Zhu, Hui Liu, and Junhui Hou. 2025. Nvs-solver: Video diffusion model as zero-shot novel view synthesizer. In *ICLR*.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. 2021. pixelnerf: Neural radiance fields from one or few images. In *CVPR*. 4578–4587.

Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. 2024. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048* (2024).

Chuanrui Zhang, Yingshuang Zou, Zhuoling Li, Minmin Yi, and Haoqian Wang. 2025. Transplat: Generalizable 3d gaussian splatting from sparse multi-view images with transformers. In *AAAI*.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*. 586–595.

Xiang Zhang, Bingxin Ke, Hayko Riemenschneider, Nando Metzger, Anton Obukhov, Markus Gross, Konrad Schindler, and Christopher Schroers. 2024. Betterdepth: Plug-and-play diffusion refiner for zero-shot monocular depth estimation. In *NeurIPS*.

Sijie Zhao, Wenbo Hu, Xiaodong Cun, Yong Zhang, Xiaoyu Li, Zhe Kong, Xiangjun Gao, Muyao Niu, and Ying Shan. 2024. Stereocrafter: Diffusion-based generation of long and high-fidelity stereoscopic 3d from monocular videos. *arXiv preprint arXiv:2409.07447* (2024).

Chuanxia Zheng and Andrea Vedaldi. 2024. Free3d: Consistent novel view synthesis without 3d representation. In *CVPR*. 9720–9731.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. 2018. Stereo magnification: learning view synthesis using multiplane images. *ACM Trans. Graph.* 37, 4, Article 65 (July 2018), 12 pages. https://doi.org/10.1145/3197517.3201323