DualDiff3D: Dual Structure-Appearance Diffusion Priors for Reliability-Enhanced 3D Gaussian Splatting

Qian Wang[1,2,*], Yu Wang[1,*], Weiqi Li[1,2], Xinhua Cheng[1], Xiandong Meng[2], Ronggang Wang[1,2,3], and Jian Zhang[1,2,3,✉]

[1] School of Electronic and Computer Engineering, Peking University, Shenzhen, China
[2] Peng Cheng Laboratory, Shenzhen, China
[3] Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology, Shenzhen, China

Fig. 1: Comparison of refined results for novel views with artifacts. DualDiff maintains structural consistency with the rendered novel view while matching the appearance of the leftmost reference view, achieving better performance.

Abstract. While 3D Gaussian Splatting (3DGS) has revolutionized 3D reconstruction and novel-view synthesis, scenarios with limited input views often lead to poor reconstruction quality and artifacts in rendered novel views. Recent efforts attempt to utilize powerful diffusion priors, yet they typically process rendered and reference views concatenated along an additional dimension in a single network. These methods overlook an inherent nature that different views should maintain appearance similarity but differ in structure due to view shifts, leading to blur caused by conflicts between the two properties. In this paper, we propose **DualDiff**, a novel pipeline that leverages dual diffusion priors with a Structure-Appearance Attention (SAA) module to introduce

∗ Equal contribution
✉ Corresponding author: zhangjian.sz@pku.edu.cn

reference guidance for refining low-quality novel views rendered from flawed 3D representations. Specifically, we retain one diffusion branch to focus on extracting structural information from the low-quality novel views, while introducing another branch to ensure appearance consistency with reference views. Furthermore, we present a 3D reconstruction framework named **DualDiff3D**, which integrates a reliability-enhanced Render-Refine-Optimize (RRO) loop to progressively and robustly incorporate the refined novel views, yielding more accurate 3DGS. Extensive experiments demonstrate that our approach outperforms state-of-the-art methods even in the inference-only setting, with further performance gains achievable through training. Our code and pre-trained weights will be publicly released.

1 Introduction

3D reconstruction and novel-view synthesis (NVS) [1, 3, 4, 19, 45] are important tasks in computer vision that aim to generate realistic images of a scene from previously unseen viewpoints by giving a limited set of input images, which drives diverse applications such as virtual/augmented reality (VR/AR), autonomous driving, and digital content creation. In recent years, breakthroughs like 3D Gaussian Splatting (3DGS) [12, 46] have demonstrated remarkable capability to produce photorealistic renderings. However, methods developed based on 3DGS encounter challenges in scenarios with sparse inputs for synthesizing novel views that are far from the input views, due to insufficient information and constraints provided by sparse viewpoints.

To address the limitations in scenes with sparse views, recent works [17, 35] have explored generative models, particularly diffusion models, for enhancing 3D reconstruction and NVS. These methods typically construct an initial low-quality 3D representation from sparse views, render a series of novel views, and then take the original sparse views as references to refine these rendered novel views. The refined novel views are subsequently incorporated into the training set for subsequent 3DGS optimization. Based on the type of diffusion prior adopted by the refinement model, these approaches can be categorized into two classes.

One class, exemplified by 3DGS-Enhancer [17], leverages video diffusion priors for refinement. They sample sequentially along a trajectory containing both reference views and novel views, then concatenate sampled views along the temporal dimension and feed them into the video diffusion model to achieve enhanced results. These works leverage the inherent temporal continuity of video models to introduce cross-view consistency constraints. However, high inference latency and computational cost make them impractical for online 3DGS optimization.

The other class, such as DIFIX3D+ [35], adopts image diffusion priors for refinement, which reduces both inference latency and computational overhead. To enable 2D image diffusion models to handle the additional view dimension, these methods directly embed the view dimension into the batch dimension. For supporting cross-view reference, they concatenate features from multiple views along the spatial dimension before the self-attention module, and restore the

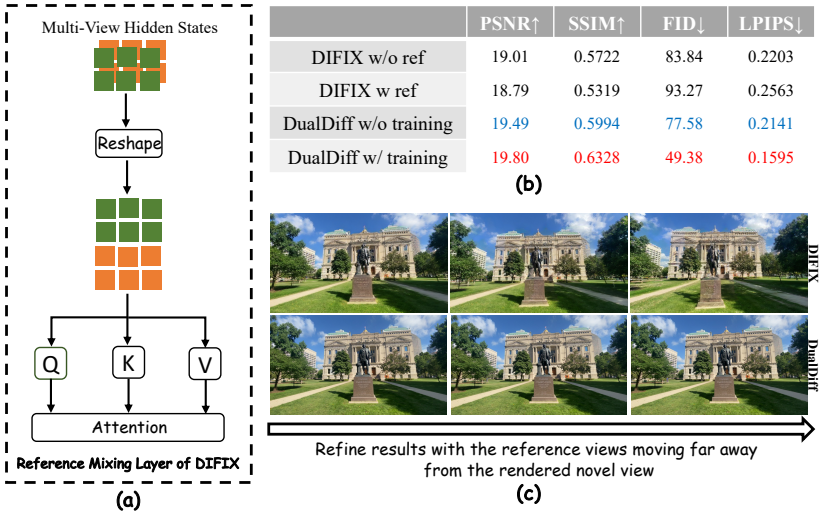

Fig. 2: Motivation of our method. (a) Illustrates the reference fusion adopted by DIFIX. (b) Shows that on the LLFF dataset, DIFIX achieves better performance in reference-free generation than in reference-based generation; in contrast, DualDiff yields performance gains merely through structural modification without any additional training. (c) Presents the subjective results of DIFIX and DualDiff as the reference view becomes increasingly distant, where DualDiff can effectively avoid blurring.

view dimension to the batch dimension after this module, as shown in Fig. 2(a). Although such a method is a straightforward strategy for introducing cross-view information interaction into image diffusion models, we identify notable drawbacks in this design, *i.e.*, novel views and reference views provide distinct information in 3D reconstruction and NVS tasks. Given a rendered novel view with artifacts, we hope the refined result to maintain consistency with the original novel view in terms of perspective (referred to as structural consistency), while filling the artifact regions with appropriately warped content from corresponding parts of the reference views (referred to as appearance consistency). Thus, we clarify that handling both structural and appearance information using a single network is inappropriate. As illustrated in Fig. 2(b), the evaluation results of DIFIX3D+ verify our argument. Surprisingly, the refinement performance with reference views is inferior to that without references. Furthermore, as the angular distance between reference views and novel views increases, the blurriness caused by conflicts between structural and appearance information in refined results becomes increasingly pronounced.

Therefore, we propose DualDiff, a novel pipeline that addresses these limitations by leveraging dual diffusion priors with a Structure-Appearance Attention (SAA) module to introduce reference guidance as presented in Fig. 3. Specifically, we retain one diffusion branch to focus on extracting structural information from the low-quality rendered novel views, and introduce another diffusion

branch to ensure appearance consistency with reference views. During refinement, we concatenate the hidden states from the self-attention module of the appearance branch to those from structure branch to produce new key and value, transforming the self-attention module of structure branch into SAA module to enable information interaction between dual branches. We also interestingly discover that the proposed DualDiff pipeline can be regarded as a "free lunch" for existing reference-based novel view refinement models leveraging diffusion priors. When using the DualDiff pipeline, even if the weights of the U-Nets in both branches are only initialized with those from DIFIX3D+ without any additional training, it can still yield a 0.7 dB gain in PSNR, as shown in Fig. 2(b). This validates the rationality of our approach, decoupling structural and appearance processing while fusing information via attention mechanisms.

Furthermore, we present a 3D reconstruction framework named DualDiff3D, which integrates a reliability-enhanced Render-Refine-Optimize (RRO) loop equipped with a Progressive Sampling and Filtering (PSF) strategy and a Confidence-Driven Weighting (CDW) process. This cyclic process iteratively strengthens the 3D representation by incorporating increasingly high-quality novel view constraints.

Finally, our contributions can be summarized as follows:

- We propose DualDiff, a novel two-branch pipeline to refine low-quality rendered novel views, guided by reference views. We are the first to recognize the necessity of decoupling structure and appearance for this task. By fully leveraging dual diffusion priors and introducing a Structure-Appearance Attention (SAA) module, our model independently ensures structural consistency and appearance consistency with distinct inputs.
- We introduce DualDiff3D, a 3D reconstruction framework based on a reliability-enhanced Render-Refine-Optimize (RRO) loop, integrated with a Progressive Sampling and Filtering (PSF) strategy and a Confidence-Driven Weighting (CDW) process. We effectively address the critical instability issue caused by directly feeding refined outputs from diffusion models into 3DGS optimization, achieving a logical and complete closed-loop solution.
- Extensive experiments on benchmark datasets demonstrate that DualDiff3D improves the quality of 3D reconstruction and novel view synthesis (NVS) in both objective metrics and subjective quality even in the inference-only scenario, with further performance gains attainable via fine-tuning.

2 Related Works

2.1 Diffusion Models

The Denoising Diffusion Probabilistic Model [8,32] has proven to be highly successful in generating high-quality images, outperforming previous approaches such as generative adversarial networks (GANs) [7,43], variational autoencoders (VAEs) [13,33], and flow-based methods [16]. With text guidance during training, users can generate images based on textual input. Noteworthy examples

include GLIDE [23], which utilizes textual prompts as conditions and adopts classifier-free guidance [9]. DALLE-2 [26] leverages the latent space of CLIP [25] as a condition, and Imagen [29] injects features from a large language model into cascaded diffusion models. To address the computational burden of the iterative denoising process, LDM [27] conducts the diffusion process on a compressed latent space rather than the original pixel space, achieving success in reducing computational requirements. This accomplishment has prompted further exploration in customization [6, 28], image guidance [37], and precise control [20, 44].

2.2 3D Reconstruction and NVS Methods

3D reconstruction and NVS have evolved around balancing rendering fidelity, efficiency, and scalability. Traditional approaches relied on explicit primitives or early implicit functions [18, 24], which either suffered from high memory costs or failed to jointly model geometry and appearance, limiting realism and scalability. Typical works like KinectFusion [22] realized real-time dense surface mapping but were confined to small-scale scenes. Neural Radiance Fields (NeRF) [1, 3–5, 19, 21] revolutionized the field by encoding scene geometry and appearance into a continuous implicit neural network. Despite generating photorealistic novel views, NeRF and its variants faced critical bottlenecks in inference speed and computational efficiency, hindering real-time applications. 3DGS [12] later emerged as a landmark explicit method that combines the advantages of implicit neural modeling and traditional point-based representations. It enables real-time photorealistic rendering but struggles with novel views far from input perspectives under sparse-view conditions.

2.3 Diffusion Priors for NVS

Diffusion priors have emerged as a transformative force in novel view synthesis, offering robust solutions to challenges like sparse input views and geometric inconsistency. A key application lies in enhancing 3D representations with view consistency. 3DGS-Enhancer [17] integrates 2D video diffusion priors into 3DGS, reformulating view consistency as temporal coherence in video generation. Similarly, GaussianSR [42] employs 2D diffusion priors via score distillation sampling to super-resolve low-resolution inputs, addressing redundancy in 3D Gaussian primitives through timestep range shrinking and primitive pruning. For dynamic scenes, DiffusionPriors [34] fine-tunes an RGB-D diffusion model on monocular video frames, distilling knowledge into 4D neural radiance fields to disentangle motion and structure—even with unknown camera poses. Handling extreme sparsity, DreamSparse [41] combines a geometry module (capturing 3D spatial priors) with a spatial guidance model to convert rendered features, guiding 2D diffusion models toward geometrically consistent novel views. DIFIX3D+ [35] trains a single-step diffusion model to enhance and remove artifacts in rendered novel views caused by suboptimal 3D representation.

3 Preliminary

3.1 3D Gaussian Splatting

3DGS is an innovative and state-of-the-art approach in the field of 3D reconstruction and NVS. Distinguished from implicit representation methods such as NeRF [19], which utilize volume rendering, 3DGS leverages the splatting technique [39] to generate images, achieving remarkable real-time rendering speed. Specifically, 3DGS represents the scene through a set of anisotropic Gaussians, defined with its center position $\boldsymbol{\mu} \in \mathbb{R}^3$, covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times3}$ which can be decomposed into scaling factor $\boldsymbol{s} \in \mathbb{R}^3$ and rotation factor $\boldsymbol{q} \in \mathbb{R}^4$, color defined by spherical harmonic coefficients $\boldsymbol{h} \in \mathbb{R}^{3\times(k+1)^2}$ (where k represents the order of spherical harmonics), and opacity $\alpha \in \mathbb{R}^1$. Then, the 3D Gaussian can be queried as follows:

$$\mathcal{G}(\mathbf{x}) = \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}, \tag{1}$$

where $\mathbf{x}$ represents the position of the query point. Subsequently, an efficient 3D to 2D Gaussian mapping [47] is employed to project the Gaussian onto the image plane:

$$\hat{\boldsymbol{\mu}} = \mathbf{PW}\boldsymbol{\mu}, \quad \hat{\boldsymbol{\Sigma}} = \mathbf{JW}\boldsymbol{\Sigma}\mathbf{W}^\top\mathbf{J}^\top, \tag{2}$$

where $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$ separately represent the 2D mean position and covariance of the projected 3D Gaussian. $\mathbf{P}$, $\mathbf{W}$ and $\mathbf{J}$ denote the projective transformation, viewing transformation, and Jacobian of the affine approximation of $\mathbf{P}$, respectively. The color of the pixel on the image plane, denoted by $\mathbf{p} = (u, v)$, uses a typical neural point-based rendering [14]. Let $\mathbf{C} \in \mathbb{R}^{H\times W\times3}$ represent the color of the rendered image, where H and W represent the height and width of the image. The rendering process is outlined as follows:

$$\begin{aligned} \mathbf{C}[\mathbf{p}] &= \sum_{i=1}^{N} \boldsymbol{c}_i \sigma_i \prod_{j=1}^{i-1}(1-\sigma_j), \\ \sigma_i &= \alpha_i \, \mathrm{e}^{-\frac{1}{2}(\mathbf{p}-\hat{\boldsymbol{\mu}})^\top \hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{p}-\hat{\boldsymbol{\mu}})}, \end{aligned} \tag{3}$$

where N represents the number of Gaussians that overlap the pixel $\mathbf{p}$. $\boldsymbol{c}_i \in \mathbb{R}^3$ and $\alpha_i \in \mathbb{R}^1$ denote the color calculated from $\boldsymbol{h}_i$ and opacity of the i-th Gaussian.

3.2 Diffusion Model

Given an input signal x_0, a diffusion forward process in DDPM [8] is defined as:

$$p_\theta(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_{t-1}}x_{t-1}, \beta_t I), \tag{4}$$

for $t = 1, ..., T$, where T is the total timestep of the diffusion process. A noise depending on the variance β_t is gradually added to x_{t-1} to obtain x_t at the next timestep and finally reach $x_T \in \mathcal{N}(0, I)$. The goal of the diffusion model is to learn to reverse the diffusion process (denoising). Given a random noise x_t, the

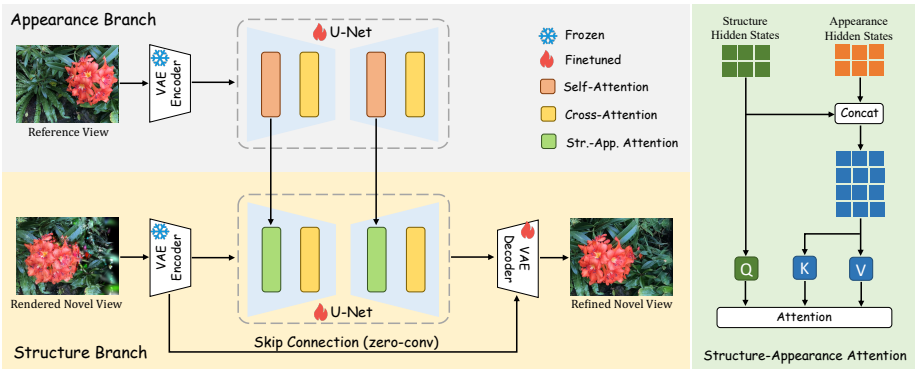

Fig. 3: Pipeline of DualDiff. DualDiff leverages dual diffusion branches integrated with a Structure-Appearance Attention (SAA) module to incorporate reference guidance, thereby refining low-quality novel views rendered from defective 3D representations.

model predicts the added noise at the next timestep x_{t-1} until the origin signal x_0:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{5}$$

for $t = T, ..., 1$. We fix the variance $\Sigma_\theta(x_t, t)$ and utilize the diffusion model with parameter θ to predict the mean of the inverse process $\mu_\theta(x_t, t)$. The model can be simplified as denoising models $\epsilon_\theta(x_t, t)$, which are trained to predict the noise of x_t with a noise prediction loss:

$$\mathcal{L} = \mathbb{E}_{x_0, y, \epsilon \sim \mathcal{N}(0,I), t}[\|\epsilon - \epsilon_\theta(x_t, t, \tau_\theta(y))\|_2^2], \tag{6}$$

where ϵ is the added noise to the input image x_0, y is the corresponding textual description, $\tau_\theta(\cdot)$ is a text encoder mapping the string to a sequence of vectors.

4 DualDiff

4.1 Inspiration

The design of DualDiff mainly draws inspiration from two sources. First, we are inspired by the latest dual diffusion branch design in other image-to-image generation tasks such as virtual try-on and image animation [11]. For example, virtual try-on tasks commonly adopt a dual-branch architecture to separately ensure structural alignment with the model image and appearance alignment with the garment image, and this design logic closely parallels our novel view refinement task. The success in these fields demonstrates the unique advantages of dual-branch architectures in addressing tasks requiring the decoupling of appearance and structural information. Furthermore, it also validates that diffusion models

trained on large datasets are naturally effective feature extractors, which can be fine-tuned to focus on specific types of image characteristics. Second, we are inspired by analyzing two primary condition injection frameworks in diffusion models. Methods like ControlNet [44] and T2I-Adapter [20] use feature addition to merge latent and conditional representations, excelling at pixel-aligned tasks (*e.g.*, skeleton-guided generation) due to element-wise fusion at consistent spatial locations. In contrast, methods like IP-Adapter [38] inject conditional features into attention mechanisms, enabling semantic region focus that suits appearance-aligned tasks (*e.g.*, style transfer). Leveraging this distinction, we adopt attention-based injection for appearance-aligned reference views.

4.2 Overview

An overview of the DualDiff pipeline is presented in Fig. 3, which is composed of a structure branch and an appearance branch. The denoising U-Net in both branches adopts a structure identical to the single-step diffusion model SD-Turbo [30] and weights initialized with DIFIX [35]. Following DIFIX, our model takes rendered novel views as input instead of Gaussian noise and sets a lower timestep $t = 200$, corresponding to a lower noise level. This design allows the refinement process to simulate a one-step denoising procedure under low-noise conditions. We utilize the same frozen VAE encoder to encode both rendered novel views and reference views into the latent space, which are then fed into the U-Net. Only the structure branch is equipped with a VAE decoder to decode the final refined novel views from the latent space to pixel space. To mitigate information loss caused by VAE encoding and decoding, we add zero-initialized skip connections between corresponding layers of the VAE encoder and decoder in the structure branch, and fine-tune the VAE decoder with LoRA [10].

4.3 Structure-Appearance Attention Module

To incorporate the appearance guidance from reference views into the structure branch, we replace the self-attention module in the denoising U-Net with our proposed Structure-Appearance Attention (SAA) module. As illustrated in Fig. 3, given the hidden states $\mathbf{H}_{\text{app}} \in \mathbb{R}^{n \times d}$ from the self-attention module of one U-Net block in the appearance branch, and the corresponding hidden states $\mathbf{H}_{\text{str}} \in \mathbb{R}^{m \times d}$ from the structure branch, we concatenate them along the sequence length dimension to obtain mixed hidden states that fuse structural and appearance features, denoted as $\mathbf{H}_{\text{mix}} \in \mathbb{R}^{(m+n) \times d}$. Since our goal is to inject information from the reference view into the novel view, we set the queries $\mathbf{Q} \in \mathbb{R}^{n \times d}$ to be derived from $\mathbf{H}_{\text{str}}$, while the keys $\mathbf{K} \in \mathbb{R}^{(m+n) \times d}$ and values $\mathbf{V} \in \mathbb{R}^{(m+n) \times d}$ are derived from $\mathbf{H}_{\text{mix}}$. This process can be formulated as:

$$\begin{aligned} \mathbf{Q} &= \mathbf{W}_q \mathbf{H}_{\text{str}}, \\ \mathbf{K} &= \mathbf{W}_k \mathbf{H}_{\text{mix}}, \\ \mathbf{V} &= \mathbf{W}_v \mathbf{H}_{\text{mix}}, \end{aligned} \tag{7}$$

where $\mathbf{W}_q$, $\mathbf{W}_k$ and $\mathbf{W}_v$ are learnable projection matrices. The structure-appearance attention is finally calculated as:

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d}}\right)\mathbf{V}. \tag{8}$$

4.4 Loss Function

Although our pipeline can achieve improvements in a training-free manner when initialized with the pre-trained weights of DIFIX, it can still benefit from training. We adopt the same loss function as DIFIX, which consists of an L2 reconstruction loss between the refined novel views and the ground truth, a perceptual LPIPS loss for better visual quality, and a Gram-Matrix loss based on features of VGG-16 [31] that encourages sharper details:

$$\mathcal{L} = \mathcal{L}_{\text{Recon}} + \mathcal{L}_{\text{LPIPS}} + \lambda_0 \mathcal{L}_{\text{Gram}}, \tag{9}$$

where λ_0 is a hyper-parameter set to 0.1 to weight Gram-Matrix loss.

5 DualDiff3D

5.1 Reliability-Enhanced RRO Loop

In scenarios with limited available views, 3DGS often yields degraded reconstructions due to insufficient scene coverage and high sensitivity to geometric inconsistencies. To address this, we propose the Render-Refine-Optimize (RRO) loop, which leverages our DualDiff pipeline to augment 3DGS's training set with refined novel views—generated from renderings of the initial low-quality 3DGS scene representation. The RRO loop consists of three core steps: (1) Render Step: render batches of novel views from the currently reconstructed 3DGS; (2) Refine Step: utilize the DualDiff pipeline to refine details and eliminate artifacts in sampled novel views; (3) Optimize Step: employ the refined novel views to further optimize the 3DGS model. However, direct incorporation of these refined images introduces three fundamental challenges: (1) Refined images inevitably exhibit minor geometric inconsistencies with ground-truth observations, which can destabilize 3DGS optimization; (2) 3DGS is highly sensitive to such geometric deviations, particularly when novel views are far from the training distribution; (3) Novel view selection is non-trivial, as arbitrary sampling may provide limited information gain while amplifying reconstruction errors. To build a more reliable and robust RRO loop, we integrate a Progressive Sampling and Filtering (PSF) strategy and a Confidence-Driven Weighting (CDW) process.

5.2 Progressive Sampling and Filtering Strategy

The PSF strategy progressively expands the training set while preserving geometric consistency. Instead of randomly sampling novel viewpoints, PSF interpolates and slightly extrapolates between existing training cameras, introducing

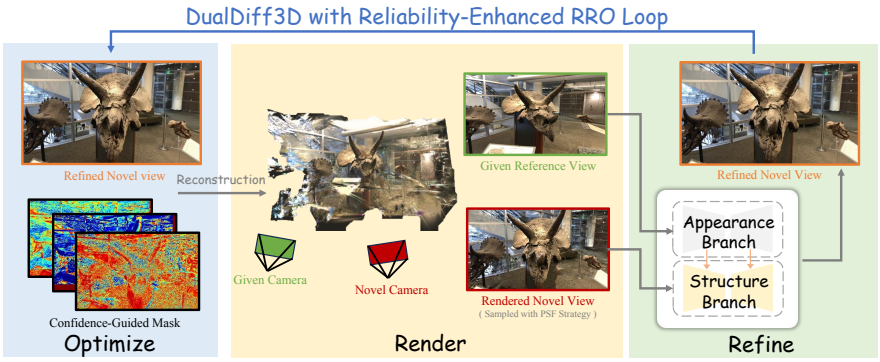

Fig. 4: Pipeline of DualDiff3D. DualDiff3D integrates a reliability-enhanced RRO loop to progressively and robustly incorporate the refined novel views, yielding more accurate 3DGS.

small perturbations to enhance diversity. Each sampled novel view is evaluated using a composite confidence mask (defined in Sec. 5.3) at both pixel and image levels. Only high-confidence views exhibiting strong geometric and appearance consistency are admitted into the augmented training pool. The renderings of these sampled novel views, paired with their closest ground-truth observations, are then fed into the DualDiff pipeline for refinement.

5.3 Confidence-Driven Weighting Process

The CDW process aims to evaluate refined novel views via a composite confidence-guided mask, ensuring only high-confidence regions are incorporated into the 3DGS training process. The confidence-guided mask $M \in [0,1]^{H\times W}$ integrates three complementary confidence measures:

Refine Variance Confidence. Each novel view is refined K times by DualDiff refinement model, producing $\{I^{(k)}_{\text{refine}}\}_{k=1}^{K}$. The per-pixel variance is used to estimate uncertainty:

$$\begin{aligned} \sigma^2(u) &= \frac{1}{K}\sum_{k=1}^{K}\left(I^{(k)}_{\text{refine}}(u) - \bar{I}_{\text{rep}}(u)\right)^2, \\ \bar{I}_{\text{rep}}(u) &= \frac{1}{K}\sum_{k=1}^{K} I^{(k)}_{\text{refine}}(u), \end{aligned} \tag{10}$$

where u denotes a pixel coordinate. The corresponding confidence is defined as:

$$C_{\text{var}}(u) = \exp(-\alpha \cdot \sigma^2(u)), \tag{11}$$

where α controls sensitivity to variance.

Pixel Discrepancy Confidence. Given the original low-quality rendering I_{low}, the absolute difference with the refined result provides another confidence cue:

$$\begin{aligned} d(u) &= \|I_{\text{refine}}(u) - I_{\text{low}}(u)\|_1, \\ C_{\text{diff}}(u) &= \exp(-\beta \cdot d(u)), \end{aligned} \tag{12}$$

where β scales the influence of discrepancies.

Reprojection Confidence. Using the rendered depth map D_{novel}, pixels in the refined novel views are reprojected into overlapping reference images $\{I^j_{\text{ref}}\}$. For each valid projection, the reprojection error is:

$$e_{\text{proj}}(u) = \frac{1}{N_u} \sum_{j=1}^{N_u} \left\| I_{\text{refine}}(u) - I^j_{\text{refine}}(\pi_j(u, D_{\text{novel}}(u))) \right\|_1, \tag{13}$$

where $\pi_j(\cdot)$ is the projection operator into reference views j, and N_u is the number of valid projections. The confidence is:

$$C_{\text{proj}}(u) = \exp(-\gamma \cdot e_{\text{proj}}(u)), \tag{14}$$

with γ controlling sensitivity.

Training Process. The final composite confidence-guided mask is:

$$\begin{aligned} M(u) &= \lambda_1 C_{\text{var}}(u) + \lambda_2 C_{\text{diff}}(u) + \lambda_3 C_{\text{proj}}(u), \\ \lambda_1 &+ \lambda_2 + \lambda_3 = 1. \end{aligned} \tag{15}$$

During optimization, the loss for refined views is defined as

$$\mathcal{L}_{\text{refine}} = \sum_u M(u) \cdot \|I_{\text{refine}}(u) - I_{\text{rend}}(u)\|_1 + \eta \cdot \text{LPIPS}(I_{\text{refine}}, I_{\text{rend}}), \tag{16}$$

where I_{rend} is the rendered image from the current 3DGS model and η balances pixel-wise and perceptual losses. To ensure more stable training, each training batch always includes a certain ratio of original training views alongside novel views. Additional validation-and-rollback mechanism monitors the model's performance on a held-out validation set, discarding newly added refined views if the 3D reconstruction quality deteriorates.

6 Experiments

6.1 Implementation Details

DualDiff. DualDiff is trained on 90% of randomly selected scenes from the DL3DV [15] benchmark dataset. For each scene, we conduct vanilla 3DGS with 3, 6, 9, and 24 views, respectively, for 30k iterations. Rendered novel views from the reconstructions, together with their corresponding ground truths and closest training views, form our training triplets. Specifically, DualDiff / DualDiff-s denotes weights trained on results from a single view count (24 views), while DualDiff-m represents weights trained on results from multiple view counts (3, 6, 9, and 24 views). We adopt the Adam optimizer with a learning rate of 2×10^{-5} and a batch size of 8, training for $10k$ iterations.

Table 1: Objective refinement results of DualDiff on the DL3DV dataset.

Method	Setting	PSNR↑	SSIM↑	FID↓	LPIPS↓	Setting	PSNR↑	SSIM↑	FID↓	LPIPS↓
3DGS	3-View	10.70	0.2608	267.77	0.6001	6-View	12.72	0.3298	214.26	0.5419
DIFIX		13.11	0.3338	133.77	0.4966		13.99	0.3641	96.87	0.4620
DualDiff-m		**13.89**	**0.3793**	103.67	**0.4407**		**14.67**	**0.4150**	66.93	**0.4020**
DualDiff-s		13.49	0.3674	**101.80**	0.4522		14.48	0.4086	**62.41**	0.4060
3DGS	9-View	14.19	0.3886	192.17	0.4950	24-View	19.42	0.6209	98.71	0.3212
DIFIX		16.02	0.4399	86.55	0.3800		19.20	0.5612	46.52	0.2730
DualDiff-m		16.76	0.4973	60.72	0.3088		20.27	0.6329	26.62	0.1989
DualDiff-s		**16.77**	**0.5001**	**58.19**	**0.3079**		**20.57**	**0.6456**	**24.77**	**0.1931**

DualDiff3D. In the 3DGS reconstruction process, we initialize the model with a randomly distributed point cloud to ensure a fair comparison. Following the setup in DIFIX3D, the 3DGS is first trained on sparse input views for 30k iterations. After the initial 3DGS reconstruction, we run another 30k iterations with a refinement step every 2k iterations. For the validation and rollback mechanism, we cache the Gaussian scene state at each refinement step. During the 2k iterations of training after caching, we evaluate the model on a held-out validation set every 500 iterations. If performance fails to improve for 3 consecutive validations, we roll back to the cached state, discard the newly added refined views, and resume 3DGS training with only the original real image set until the next refinement step. In our experiments, we set the number of refinement steps $K = 3$, and loss weights $\lambda_1 = 0.2, \lambda_2 = 0.3, \lambda_3 = 0.5$ via grid parameter search.

6.2 Refinement Results

We evaluate the novel view refinement results of DualDiff, guided by reference views on the remaining 10% of scenes from the DL3DV dataset, as well as the LLFF dataset, with comparisons against vanilla 3DGS and DIFIX (the refinement model of DIFIX3D+). Consistent with the training dataset construction process, we conduct tests on the DL3DV dataset by refining novel views rendered from 3D reconstructions using 3, 6, 9, and 24 sparse input views. As shown in Tab. 1, DualDiff outperforms both vanilla 3DGS and DIFIX by a large margin across all experimental settings. DualDiff-m demonstrates superior performance in few-view test scenarios, as it has encountered more severe rendering degradations in the training phase. Although DualDiff-s is only trained on 24-view reconstructions, its refinement capability remains effective across various test settings. The subjective results, as shown in Fig. 1, demonstrate that DualDiff significantly reduces artifacts, restores finer details, and achieves the best appearance consistency with the reference view. Due to space constraints, additional results are provided in the supplementary material.

6.3 Reconstruction Results

We compare the 3D reconstruction performance of vanilla 3DGS, DIFIX3D, and DualDiff3D on both the DL3DV dataset and the LLFF dataset. Tab. 2 summa-

Fig. 5: Comparison of subjective results of DualDiff3D and baselines on the DL3DV and LLFF datasets. Regions with significant differences are highlighted by red boxes for easy comparison.

Table 2: Objective reconstruction results of DualDiff3D on the DL3DV and LLFF datasets.

Method	3-View			6-View			9-View			24-View		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
DL3DV												
3DGS	10.70	0.2608	0.6001	12.72	0.3298	0.5419	14.19	0.3886	0.4950	19.42	0.6209	0.3212
DIFIX3D	11.36	0.3314	0.5759	13.35	0.3892	0.5084	14.96	0.4464	0.4617	20.17	0.6469	0.2971
DualDiff3D	**12.56**	**0.3731**	**0.5473**	**14.11**	**0.4122**	**0.4837**	**15.60**	**0.4697**	**0.4436**	**20.49**	**0.6576**	**0.2745**
LLFF												
3DGS	14.11	0.3819	0.4666	18.69	0.5687	0.3237	20.50	0.6508	0.2726	-	-	-
DIFIX3D	15.21	0.4645	0.4119	19.85	0.6159	0.2821	21.53	0.6807	0.2410	-	-	-
DualDiff3D	**16.34**	**0.4981**	**0.3835**	**20.35**	**0.6709**	**0.2611**	**21.89**	**0.7106**	**0.2154**	-	-	-

rizes the objective performance across key metrics, where DualDiff3D achieves the highest metric values under all view count conditions. This not only validates the superior reconstruction capability of DualDiff3D but also demonstrates its strong generalization ability considering it was not trained on the LLFF dataset for refinement. Fig. 5 visualizes the 3D reconstruction results of the mentioned methods. These visual results confirm that DualDiff3D achieves an excellent balance between photorealism and geometric coherence.

Following the evaluation protocol of GenFusion [36] and GSFixer [40], we compare DualDiff3D with more baselines on Mip-NeRF360 [2] (Tab. 4). Due to the lack of open-source code for 3DGS-Enhancer, we utilize the values reported in their paper.

Table 3: Ablation results for key components of DualDiff3D.

Variants	Components		DualDiff				DIFIX			
	PSF	CDW	PSNR↑	SSIM↑	LPIPS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	FID↓
(a)			15.43	0.4649	0.4031	107.50	15.21	0.4645	0.4119	115.23
(b)	✓		15.71	0.4733	0.3910	105.02	15.44	0.4699	0.4100	111.36
(c)		✓	16.04	0.4877	0.3881	103.71	15.79	0.4764	0.4012	108.56
(d)	✓	✓	**16.34**	**0.4981**	**0.3835**	**101.25**	**15.96**	**0.4811**	**0.3903**	**106.77**

Table 4: Comparison with baselines on the Mip-NeRF360 dataset. * denotes values reported in the original paper.

Method	**3-View**			**6-View**			**9-View**		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
3DGS	13.06	0.251	0.576	14.96	0.355	0.505	16.79	0.447	0.446
FSGS	14.17	0.318	0.578	16.12	0.415	0.517	17.94	0.492	0.468
GenFusion	15.29	0.369	0.585	17.16	0.447	0.500	18.36	0.496	0.465
3DGS-Enhancer*	-	-	-	13.96	0.260	0.570	16.22	0.399	0.454
DIFIX3D	14.32	0.313	0.571	16.07	0.408	0.459	17.82	0.477	0.418
GSFixer	15.61	0.370	0.559	**17.27**	0.426	0.478	**18.63**	0.481	0.420
Ours	**15.70**	**0.373**	**0.554**	17.22	**0.439**	**0.451**	18.45	**0.486**	**0.405**

6.4 Ablation Study

To validate the contributions of key components in DualDiff3D, we conducted ablation studies on the LLFF dataset under the 3-view setting, with quantitative results summarized in Tab. 3. Specifically, we ablate three key components one by one: the refinement model (replacing DualDiff with DIFIX), the Progressive Sampling and Filtering (PSF) strategy, and the Confidence-Driven Weighting (CDW) process. Here, the baseline refers to directly incorporating refined novel views into 3DGS reconstruction without additional optimization. The full DualDiff3D model integrates all components and delivers the best performance. We also ablate K and each CDW term in Tab. 5. $K=3$ achieves a quality/cost trade-off, and combining three terms yields the best performance.

6.5 Computational Cost

We report the runtime parameters of three methods in Tab. 6. Despite having a larger backbone network, DualDiff3D outperforms DIFIX3D in terms of average single refining runtime. This is primarily because DIFIX3D's approach of placing reference views in the batch dimension restricts the refinement of only one input view at a time, whereas our method employs two separate networks to process input views and reference views, respectively. With refined times $k = 3$ for reconstruction, our final runtime is slightly longer. However, considering the substantial performance gains and the fact that our method is not designed for real-time 3DGS reconstruction, we argue that this time overhead is justifiable.

Table 5: Ablation on K and individual CDW terms.

Variant	K	C_{var}	C_{diff}	C_{proj}	PSNR↑	SSIM↑	LPIPS↓
w/o CDW	1				19.97	0.6213	0.2704
only C_{var}	3	✓			20.06	0.6225	0.2715
only C_{diff}	3		✓		20.11	0.6310	0.2696
only C_{proj}	3			✓	20.26	0.6612	0.2633
w/ full CDW (Ours)	3	✓	✓	✓	**20.35**	**0.6709**	**0.2611**
w/ full CDW	2	✓	✓	✓	20.26	0.6633	0.2626
w/ full CDW	4	✓	✓	✓	20.37	0.6701	0.2609
w/ full CDW	5	✓	✓	✓	20.33	0.6689	0.2606

Table 6: Runtime parameters on a single RTX 4090 GPU.

	RefineTime	Recon. Time	RefineVRAM-Avg	RefineVRAM-Peak
3DGS	-	6min29s	-	-
DIFIX3D	950ms	8min52s	6048.2MB	9936.7MB
DualDiff3D	600ms	10min40s	9750.5MB	12574.6MB

7 Conclusion

In this paper, we present DualDiff3D, a framework designed to advance 3D reconstruction and novel-view synthesis. At the core of our framework lies DualDiff, a dual-branch diffusion pipeline tailored to eliminate artifacts in novel views rendered from 3DGS. A proposed SAA mechanism connects the structure branch and appearance branch, enabling image quality improvement in a "free lunch" manner. DualDiff3D leverages a Render-Refine-Optimize Loop, integrated with sampling strategy and confidence-guided masks, to progressively and robustly refine 3DGS. Extensive experiments validate the effectiveness of our approach, which delivers superior visual quality compared to baselines. We hope that DualDiff3D offers a more rational solution for diffusion-based NVS and 3D reconstruction, and may inspire more effective designs.

8 Acknowledgements

This work is financially supported in part by National Natural Science Foundation of China (62372016), National Science and Technology Major Project-Mobile Information Networks (2024ZD130060), Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology (2024B1212010006), Shenzhen Science and Technology Program (SYSPG20241211173440004) and Outstanding Talents Training Fund in Shenzhen.

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5855–5864 (2021)
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5470–5479 (2022)
3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19697–19705 (2023)
4. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: European conference on computer vision. pp. 333–350. Springer (2022)
5. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5501–5510 (2022)
6. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A.H., Chechik, G., Cohen-Or, D.: An image is worth one word: Personalizing text-to-image generation using textual inversion. arXiv preprint arXiv:2208.01618 (2022)
7. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
8. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
9. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
10. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. ICLR **1**(2), 3 (2022)
11. Hu, L.: Animate anyone: Consistent and controllable image-to-video synthesis for character animation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8153–8163 (2024)
12. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
13. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
14. Kopanas, G., Leimkühler, T., Rainer, G., Jambon, C., Drettakis, G.: Neural point catacaustics for novel-view synthesis of reflections. ACM Transactions on Graphics (TOG) **41**(6), 1–15 (2022)
15. Ling, L., Sheng, Y., Tu, Z., Zhao, W., Xin, C., Wan, K., Yu, L., Guo, Q., Yu, Z., Lu, Y., et al.: Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22160–22169 (2024)
16. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)
17. Liu, X., Zhou, C., Huang, S.: 3dgs-enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors. Advances in Neural Information Processing Systems **37**, 133305–133327 (2024)

18. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4460–4470 (2019)
19. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
20. Mou, C., Wang, X., Xie, L., Wu, Y., Zhang, J., Qi, Z., Shan, Y.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In: Proceedings of the AAAI conference on artificial intelligence. vol. 38, pp. 4296–4304 (2024)
21. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM transactions on graphics (TOG) **41**(4), 1–15 (2022)
22. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: 2011 10th IEEE international symposium on mixed and augmented reality. pp. 127–136. Ieee (2011)
23. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021)
24. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 165–174 (2019)
25. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
26. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022)
27. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
28. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 22500–22510 (2023)
29. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in neural information processing systems **35**, 36479–36494 (2022)
30. Sauer, A., Boesel, F., Dockhorn, T., Blattmann, A., Esser, P., Rombach, R.: Fast high-resolution image synthesis with latent adversarial diffusion distillation. In: SIGGRAPH Asia 2024 Conference Papers. pp. 1–11 (2024)
31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
32. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)

33. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. Advances in neural information processing systems **30** (2017)
34. Wang, C., Zhuang, P., Siarohin, A., Cao, J., Qian, G., Lee, H.Y., Tulyakov, S.: Diffusion priors for dynamic view synthesis from monocular videos. arXiv preprint arXiv:2401.05583 (2024)
35. Wu, J.Z., Zhang, Y., Turki, H., Ren, X., Gao, J., Shou, M.Z., Fidler, S., Gojcic, Z., Ling, H.: Difix3d+: Improving 3d reconstructions with single-step diffusion models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26024–26035 (2025)
36. Wu, S., Xu, C., Huang, B., Geiger, A., Chen, A.: Genfusion: Closing the loop between reconstruction and generation via videos. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 6078–6088 (2025)
37. Yang, B., Gu, S., Zhang, B., Zhang, T., Chen, X., Sun, X., Chen, D., Wen, F.: Paint by example: Exemplar-based image editing with diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18381–18391 (2023)
38. Ye, H., Zhang, J., Liu, S., Han, X., Yang, W.: Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. arXiv preprint arXiv:2308.06721 (2023)
39. Yifan, W., Serena, F., Wu, S., Öztireli, C., Sorkine-Hornung, O.: Differentiable surface splatting for point-based geometry processing. ACM Transactions On Graphics (TOG) **38**(6), 1–14 (2019)
40. Yin, X., Zhang, Q., Chang, J., Feng, Y., Fan, Q., Yang, X., Pun, C.M., Zhang, H., Cun, X.: Gsfixer: Improving 3d gaussian splatting with reference-guided video diffusion priors. arXiv preprint arXiv:2508.09667 (2025)
41. Yoo, P., Guo, J., Matsuo, Y., Gu, S.S.: Dreamsparse: Escaping from plato's cave with 2d diffusion model given sparse views. Advances in Neural Information Processing Systems **36**, 3307–3324 (2023)
42. Yu, X., Zhu, H., He, T., Chen, Z.: Gaussiansr: 3d gaussian super-resolution with 2d diffusion priors. arXiv preprint arXiv:2406.10111 (2024)
43. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 5907–5915 (2017)
44. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3836–3847 (2023)
45. Zhang, X., Srinivasan, P.P., Deng, B., Debevec, P., Freeman, W.T., Barron, J.T.: Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. ACM Transactions on Graphics (ToG) **40**(6), 1–18 (2021)
46. Zhu, Z., Fan, Z., Jiang, Y., Wang, Z.: Fsgs: Real-time few-shot view synthesis using gaussian splatting. In: European conference on computer vision. pp. 145–163. Springer (2024)
47. Zwicker, M., Pfister, H., Van Baar, J., Gross, M.: Ewa splatting. IEEE Transactions on Visualization and Computer Graphics **8**(3), 223–238 (2002)