3DENHANCER: Consistent Multi-View Diffusion for 3D Enhancement

Yihang Luo Shangchen Zhou Yushi Lan Xingang Pan Chen Change Loy
S-Lab, Nanyang Technological University
https://yihangluo.com/projects/3DEnhancer

Figure 1. Our proposed 3DEnhancer showcases excellent capabilities in enhancing multi-view images generated by various models. As shown in (a), it can significantly improve texture details, correct texture errors, and enhance consistency across views. Beyond enhancement, as illustrated in (b), 3DEnhancer also supports texture-level editing, including regional inpainting, and adjusting texture enhancement strength via noise level control. **(Zoom-in for best view)**

Abstract

Despite advances in neural rendering, due to the scarcity of high-quality 3D datasets and the inherent limitations of multi-view diffusion models, view synthesis and 3D model generation are restricted to low resolutions with suboptimal multi-view consistency. In this study, we present a novel 3D enhancement pipeline, dubbed 3DENHANCER*, which employs a multi-view latent diffusion model to enhance coarse 3D inputs while preserving multi-view consistency. Our method includes a pose-aware encoder and a diffusion-based denoiser to refine low-quality multi-view images, along with data augmentation and a multi-view attention module with epipolar aggregation to maintain consistent, high-quality 3D outputs across views. Unlike existing video-based approaches, our model supports seamless multi-view enhancement with improved coherence across diverse viewing angles. Extensive evaluations show that* 3DENHANCER *significantly outperforms existing methods, boosting both multi-view enhancement and per-instance 3D optimization tasks.*

1. Introduction

The advancements in generative models [19, 24] and differentiable rendering [43] have paved the way for a new research field known as neural rendering [60]. In addition to pushing the boundaries of view synthesis [32], the generation and editing of 3D models [13, 30, 34–36, 39, 54, 81, 84] has become achievable. These methods are trained on the large-scale 3D datasets, *e.g.*, Objaverse dataset [14], enabling fast and diverse 3D synthesis.

Despite these advances, several challenges remain in 3D generation. A key limitation is the scarcity of high-quality 3D datasets; unlike the billions of high-resolution image and video datasets available [50], current 3D datasets [15] are limited to a much smaller scale [47]. Another limitation is the dependency on multi-view (MV) diffusion models [53, 54]. Most state-of-the-art 3D generative models [58, 71] follow a *two-stage* pipeline: first, generating multi-view images conditioned on images or text [54, 65], and then reconstructing 3D models from these generated views [27, 58]. Consequently, the low-quality results and view inconsistency issues of multi-view diffusion models [54] restrict the quality of the final 3D output. Besides, existing novel view synthesis methods [32, 43] usually require dense, high-resolution input views for optimization, making 3D content creation challenging when only low-resolution sparse captures are available.

In this study, we address these challenges by introducing a versatile 3D enhancement framework, dubbed 3DENHANCER, which leverages a text-to-image diffusion model as the 2D generative prior to enhance general coarse 3D inputs. The core of our proposed method is a multi-view latent diffusion model (LDM) [48] designed to enhance coarse 3D inputs while ensuring multi-view consistency. Specifically, the framework consists of a pose-aware image encoder that encodes low-quality multi-view renderings into latent space and a multi-view-based diffusion denoiser that refines the latent features with view-consistent blocks. The enhanced views are then either used as input for multi-view reconstruction or directly serve as reconstruction targets for optimizing the coarse 3D inputs.

To achieve practical results, we introduce diverse degradation augmentations [69] to the input multi-view images, simulating the distribution of coarse 3D data. In addition, we incorporate efficient multi-view row attention [28, 36] to ensure consistency across multi-view features. To further reinforce coherent 3D textures and structures under significant viewpoint changes, we also introduce near-view epipolor aggregation modules, which directly propagate corresponding tokens across near views using epipolar-constrained feature matching [11, 18]. These carefully designed strategies effectively contribute to achieving high-quality, consistent multi-view enhancement.

The most relevant works to our study are 3D enhancement approaches using video diffusion models [52, 74]. While video super-resolution (VSR) models [76] can also be adapted for 3D enhancement, several challenges that make them less suitable for use as generic 3D enhancers. First, these methods are limited to enhancing 3D model reconstructions through per-instance optimization, whereas our approach can seamlessly enhance 3D outputs by integrating multi-view enhancement into the existing *two-stage* 3D generation frameworks (*e.g.*, from "MVDream [54] → LGM [58]" to "MVDream → 3DENHANCER → LGM"). Second, video models often struggle with long-term consistency and fail to correct generation artifacts in 3D objects under significant viewpoint variations. Besides, video diffusion models based on temporal attention [3] face limitations in handling long videos due to memory and speed constraints. In contrast, our multi-view enhancer models texture correspondences across various views both implicitly and explicitly, by utilizing multi-view raw attention and near-view epipolar aggregation, leading to superior view consistency and higher efficiency.

In summary, we present a novel 3DENHANCER for generic 3D enhancement using multi-view denoising diffusion. Our contributions include a robust data augmentation pipeline, and the hybrid view-consistent blocks that integrate multi-view row attention and a near-view epipolar aggregation module to promote view consistency. Compared to existing enhancement methods, our multi-view 3D enhancement framework is more versatile and supports texture refinement. We conduct extensive experiments on both multi-view enhancement and per-instance optimization tasks to evaluate the model's components. Our proposed pipeline significantly improves the quality of coarse 3D objects and consistently surpasses existing alternatives.

2. Related Work

3D Generation with Multi-view Diffusion. The success of 2D diffusion models [24, 57] has inspired their application to 3D generation. Score distillation sampling [46, 70] distills 3D from a 2D diffusion model but faces challenges like expensive optimization, mode collapse, and the Janus problem. More recent methods propose learning the 3D via a two-stage pipeline: multi-view images generation [42, 53, 54, 71] and feed-forward 3D reconstruction [27, 58, 75]. Though yielding promising results, their performance is bounded by the quality of the multi-view generative models, including the violation of strict view consistency [39] and failing to scale up to higher resolution [53]. Recent work has focused on developing more 3D-aware attention operations, such as epipolar attention [28, 61] and row-wise attention [36]. However, we find that enforcing strict view consistency remains challenging when relying solely on attention-based operations.

Image and Video Super-Resolution. Image and video SR aims to improve visual quality by upscaling low-resolution content to high resolution. Research in this field has evolved from focusing on pre-defined single degradations [5, 6, 9, 12, 37, 38, 66, 67, 86, 88] (*e.g.*, bicubic downsampling) to addressing unknown and complex degradations [7, 69, 82, 90] in real-world scenarios. To tackle real-world enhancement, some studies [7, 69, 82, 90] introduce effective degradation pipelines that simulate diverse degradations for data augmentation during training, significantly boosting performance in handling real-world cases. To achieve photorealistic enhancement, recent work has integrated various generative priors to produce detailed textures, including StyleGAN [4, 68, 79], codebook [8, 89], and the latest diffusion models [64, 91]. For instance, StableSR [64] leverages the pretrained image diffusion model, *i.e.*, Stable Diffusion (SD) [48], for image enhancement, while Upscale-A-Video [91] further extends the diffusion model for video upscaling. Video SR networks commonly employ recurrent frame fusion [67, 87], optical flow-guided propagation [5–7, 38] or temporal attention [91] to enhance temporal consistency across adjacent frames. However, due to large spatial misalignments from viewpoint changes, these methods face challenges in establishing long-range correspondences across multi-view images, making them unsuitable for multi-view fusion for 3D. In this study, we focus on exploiting a text-to-image diffusion model to achieve robust 3D enhancement while preserving view consistency.

3D Texture Enhancement. With the rapid advancement of 3D generative models [2, 13, 34, 35, 84], attention is paid to further improve 3D generation quality through a cascade 3D enhancement module. Meta 3D Gen [1, 2] proposes a UV space enhancement model to achieve sharper textures. However, training the UV-specific enhancement model requires spatially continuous UV maps, which are limited in both quantities [14] and qualities [29]. Intex [59] and SyncMVD [40] also employ UV space for generating and enhancing 3D textures. However, these techniques are specifically designed for 3D mesh with UV coordinates, making them unsuitable for other 3D representations like 3DGS [32]. Unique3D [72] and CLAY [84] apply 2D enhancement module RealESRGAN [69] directly to the generated multi-view outputs. Though straightforward, this approach risks compromising 3D consistency across the multi-view results. MagicBoost [78] introduces a 3D refinement pipeline but relies on computationally expensive SDS optimization. SuperGaussian [52] and 3DGS-Enhancer [74] propose to enhance 3D through 2D video generative priors [3, 76]. These pre-trained video models struggle to maintain long-range consistency under large viewpoint variations, making them less effective at fixing texture errors in multi-view generation.

3. Methodology

A common pipeline in current 3D generation involves an *image-to-multiview* stage [65], followed by *multiview-to-3D* [58] generation that converts these multi-view images into a 3D object. However, due to limitations in resolution and view consistency [39], the resulting 3D outputs often lack high-quality textures and detailed geometry. The proposed multi-view enhancement network, 3DENHANCER, aims at improving the quality of 3D representations. Our motivation is that if we can obtain high-quality and view-consistent multi-view images, then the quality of 3D generation can be correspondingly enhanced.

As illustrated in Fig. 2, our framework employs a Diffusion Transformer (DiT) based LDM [10, 44] as the backbone. We incorporate a pose-aware encoder and view-consistent DiT blocks to ensure multi-view consistency, allowing us to leverage the powerful multi-view diffusion models to enhance both coarse multi-view images and 3D models. The enhanced multi-view images can improve the performance of pre-trained feed-forward 3D reconstruction models, *e.g.*, LGM [58], as well as optimize a coarse 3D model through iterative updates.

Preliminary: Multi-view Diffusion Models. LDM [24, 48, 62] is designed to acquire a prior distribution $p_{\boldsymbol{\theta}}(\mathbf{z})$ within the perceptual latent space, whose training data is the latent obtained from the trained VAE encoder $\mathcal{E}_{\boldsymbol{\phi}}$. By training to predict a denoised variant of the noisy input $\mathbf{z}_t = \alpha_t \mathbf{z} + \sigma_t \boldsymbol{\epsilon}$ at each diffusion step t, $\boldsymbol{\epsilon}_\Theta$ gradually learns to denoise from a standard Normal prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$ by solving a reverse SDE [24].

Similarly, multi-view diffusion generation models [54, 77] consider the joint distribution of multi-view images $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, where each set of $\mathcal{X}$ contains RGB renderings $\mathbf{x}_\mathbf{v} \in \mathbb{R}^{H \times W \times 3}$ from the same 3D asset given viewpoints $\mathcal{C} = \{\pi_1, \ldots, \pi_N\}$. The latent diffusion process is identical to diffusing each encoded latent $\mathbf{z} = \mathcal{E}_{\boldsymbol{\phi}}(\mathbf{x})$ independently with the shared noise schedule: $\mathcal{Z}_t = \{\alpha_t \mathbf{z} + \sigma_t \boldsymbol{\epsilon} \mid \mathbf{z} \in \mathcal{Z}\}$. Formally, given the multi-view data $\mathcal{D}_{mv} := \{\mathcal{X}, \mathcal{C}, y\}$, the corresponding diffusion loss is defined as:

$$\mathcal{L}_{MV}(\theta, \mathcal{D}_{mv}) = \mathbb{E}_{\mathcal{Z}, y, \pi, t, \boldsymbol{\epsilon}} \left[\| \epsilon - \boldsymbol{\epsilon}_\Theta(\mathcal{Z}_t; y, \pi, t) \|_2^2 \right], \quad (1)$$

where y is the optional text or image condition.

3.1. Pose-aware Encoder

Given the posed multi-view images $\mathcal{X}$, we add controllable noise to the images as an augmentation to enable controllable refinement, as described later in Sec. 3.3. To further inject camera condition for each view v, we follow the prior work [35, 55, 58, 77], and concatenate Plücker coordinates $\mathbf{r}^i_\mathbf{v} = (\mathbf{d}^i, \mathbf{o}^i \times \mathbf{d}^i) \in \mathbb{R}^6$ with image RGB values $\mathbf{x}^i_\mathbf{v} \in \mathbb{R}^3$ along the channel dimension. Here, $\mathbf{o}^i$ and $\mathbf{d}^i$ are the ray

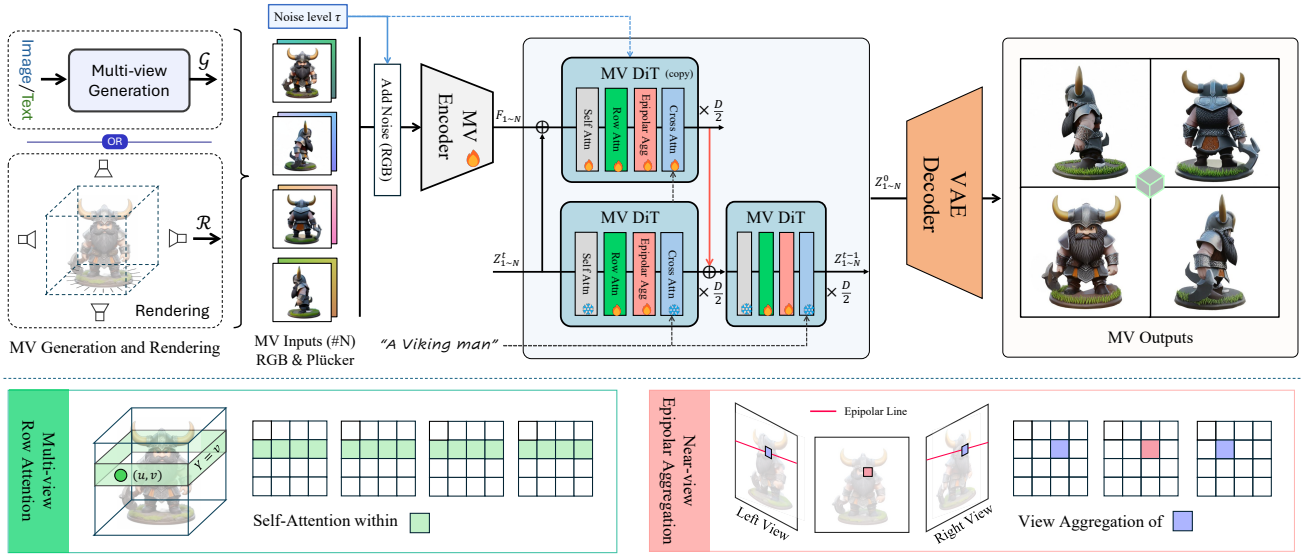


Figure 2. An overview of 3DENHANCER. By harnessing the power of generative priors, 3DENHANCER adapts a text-to-image diffusion model to a multi-view framework aimed at 3D enhancement. It is compatible with multi-view images generated by models like MVDream [54] or those rendered from coarse 3D representations, such as NeRFs [43] and 3DGS [32]. Given low-quality multi-view images along with their corresponding camera poses, 3DENHANCER aggregates multi-view information within a DiT [44] framework using row attention and epipolar aggregation modules, improving visual quality while preserving high consistency across views. Furthermore, the model supports texture-level editing via text prompts and adjustable noise levels, allowing users to correct texture errors and control the enhancement strength.

origin and ray direction for pixel i from view $\mathbf{v}$, and $\times$ denotes the cross product. We then send the concatenated results to a trainable pose-aware multi-view encoder $\mathcal{E}_\psi$, whose outputs are injected into the pre-trained DiT through a learnable copy [83]. The visual illustration of our design is shown in Fig. 2.

3.2. View-Consistent DiT Block

The main challenge of 3D enhancement is achieving precise view consistency across generated 2D multi-view images. Multi-view diffusion methods commonly rely on multi-view attention layers to exchange information across different views, aiming to generate multiview-consistent images. A prevalent approach is extending self-attention to all views, known as dense multi-view attention [42, 54]. While effective, this method significantly raises both computational demands and memory requirements.

To further enhance the effectiveness and efficiency of inter-view aggregation, we introduce two efficient modules in the DiT blocks: multi-view row attention and near-view epipolar aggregation, as shown in Fig. 2.

Multi-view Row Attention. To enhance the noisy input views to higher resolution, *e.g.*, 512×512, efficient multi-view attention is required to facilitate cross-view information fusion. Considering the epipolar constraints [21], the 3D correspondences across multiple views always lie on the epipolar line [28, 61]. Since our diffusion denoising is performed on $16\times$ downsampled features [10], and typical multi-view settings often involve elevation angles around $0°$, we assume that horizontal rows approximate the epipolar line. Therefore, we adopt the special epipolar attention, specifically the multi-view row attention [36], which enables efficient information exchange among multi-view features.

Specifically, the input cameras are chosen to look at the object with their Y axis aligned with the gravity direction and cameras' viewing directions are approximately horizontal (i.e., the pitch angle is generally level, with no significant deviation). This case is visualized in Fig. 2, for a coordinate (u, v) in the attention feature space of one view, the corresponding epipolar line in the attention feature space of other views can be approximated as $Y = v$. This enables the extension of self-attention layers calculated on tokens within the same row across multiple views to learn 3D correspondences. As ablated in Tab. 4, the multi-view row attention can efficiently encourage view consistency with minor memory consumption.

Near-view Epipolar Aggregation. Though multi-view attention can effectively facilitate view consistency, we observe that the attention-only operation still struggles with accurate correspondences across views. To address this issue, we incorporate explicit feature aggregation among neighboring views to ensure multi-view consistency. Specifically, given the output features $\{\mathbf{f_v}\}_{\mathbf{v}=1}^{N}$ from the multi-view row attention layers for each posed multi-view input, we propagate features by finding near-view corre-

spondences with epipolar line constraints. Formally, for the feature map $\mathbf{f_v}$ corresponding to the posed image $\mathbf{x_v}$, we calculate its correspondence map $M_{\mathbf{v},\mathbf{k}}$ with a near-view $\mathbf{x_k}$ as follows:

$$M_{\mathbf{v},\mathbf{k}}[i] = \underset{j,\, j^\top F i=0}{\arg\min}\ D(\mathbf{f_v}[i], \mathbf{f_k}[j]), \tag{2}$$

where D denotes the cosine distance, and $\mathbf{k} \in \{\mathbf{v}-1, \mathbf{v}+1\}$ represents the two nearest neighbor views of the given pose. Here, i and j are indices of the spatial locations in the feature maps, F is the fundamental matrix relating the two views $\mathbf{v}$ and $\mathbf{k}$, and the index j lies on the epipolar line in the view $\mathbf{k}$, subject to the constraint $j^\top F i = 0$. We then obtain the aggregated feature map $\widetilde{\mathbf{f}}_\mathbf{v}$ of the view $\mathbf{v}$ by linearly combining features of correspondences from the two nearest views via:

$$\begin{aligned}\widetilde{\mathbf{f}}_\mathbf{v}[i] = w \cdot \mathbf{f}_{\mathbf{v}-1}[M_{\mathbf{v},\mathbf{v}-1}[i]] \\ + (1-w) \cdot \mathbf{f}_{\mathbf{v}+1}[M_{\mathbf{v},\mathbf{v}+1}[i]],\end{aligned} \tag{3}$$

where w represents the weight to combine the features of the two nearest views. The calculation of w uses a hybrid fusion strategy, which ensures that the weight assignment accounts for both the *physical camera distance* and the *token feature similarity* (see the Appendix Sec. A.3). As the feature aggregation process is non-differentiable, we adopt the straight-through estimator sg[·] in VQVAE [63] to facilitate gradient back-propagation in the token space. Near-view epipolar aggregation explicitly propagates tokens from neighboring views, which greatly improves view consistency. However, due to substantial view changes, the corresponding tokens may not be available, leading to unexpected artifacts during token replacement. To address this, we fuse the feature $\mathbf{f_v}$ of the current view with the feature $\widetilde{\mathbf{f}}_\mathbf{v}$ from near-view epipolar aggregation, with 0.5 averaging. This effectively combines multi-view row attention and near-view epipolar aggregation, thereby enhancing view consistency both implicitly and explicitly.

This approach is similar to token-space editing methods like TokenFlow [18] and DGE [11]. However, we propose a trainable version that considers both geometric and feature similarity for effective feature fusion.

3.3. Multi-view Data Augmentation

Our goal is to train a versatile and robust enhancement model that performs well on low-quality multi-view images from diverse data sources, such as those generated by image-to-3D models or rendered from coarse 3D representations. To achieve this, we carefully design a comprehensive data augmentation pipeline to expand the distribution of distortions in our base training data, bridging the domain gap between training and inference.

Texture Distortion. To emulate the low-quality textures and local inconsistencies found in synthesized multi-view images, we employ a texture degradation pipeline commonly used in 2D enhancement [69, 91]. This pipeline randomly applies downsampling, blurring, noise, and JPEG compression to degrade the image quality. To simulate inconsistencies between views, we adopt different stochastic degradation parameters for each view.

Texture Deformation and Camera Jitter. As in LGM [58], we introduce grid distortion to simulate texture inconsistencies in multi-view images and apply camera jitter augmentation to introduce variations in the conditional camera poses of multi-view inputs.

Color Shift. We also observe color variations in corresponding regions between multi-view images generated by image-to-3D models. By randomly applying color changes to some image patches, we encourage the model to produce results with consistent colors. In addition, renderings from a coarse 3DGS sometimes result in a grayish overlay or ghosting artifacts, akin to a translucent mask. To simulate this effect, we randomly apply a semi-transparent object mask to the image, allowing the model to learn to remove the overlay and improve 3D visual quality.

Noise-level Control. To control the enhancement strength, we apply noise augmentation by adding controllable noise to the input multi-view images. This noise augmentation process is similar to the diffusion process in diffusion models. This approach can further enhance the model's robustness in handling unseen artifacts [91]. Generally, higher noise levels lead to stronger refinement and regeneration. This design allows users to adjust the noise levels, balancing the enhancement and generation, as shown in Fig. 1(b).

3.4. Inference for 3D Enhancement

We present two ways to utilize our 3DENHANCER for 3D enhancement:

- The proposed method can be directly applied to generation results from existing multi-view diffusion models [25, 39, 41, 54], and the enhanced output shall serve as the input to the multi-view 3D reconstruction models [22, 58, 71, 80]. Given the enhanced multi-view inputs with sharper textures and view-consistent geometry, our method can be directly used to improve the quality of existing multi-view to 3D reconstruction frameworks.
- Our method can also be used for *directly* enhancing a coarse 3D model through iterative optimization. Specifically, given an initial coarse 3D reconstruction as $\mathcal{M}$ and a set of viewpoints $\{\pi_\mathbf{v}\}_{\mathbf{v}=1}^N$, we first render the corresponding views $\mathcal{X} = \{\mathbf{x_v}\}_{\mathbf{v}=1}^N$ where $\mathbf{x_v} = \text{Rend}(\mathcal{M}, \pi_\mathbf{v})$ is obtained with the corresponding rendering techniques [32, 43]. Let $\mathcal{X}' = \{\mathbf{x}'_\mathbf{v}\}_{\mathbf{v}=1}^N$ be the enhanced multi-view images, we can then update the 3D

Figure 3. Qualitative comparisons of enhancing multi-view synthesis on the Objaverse synthetic dataset. As can be seen, only 3DENHANCER can correct flowed and missing textures with view consistency. **(Zoom-in for best view)**

model $\mathcal{M}$ by supervising it with $\mathcal{X}'$ as

$$\mathcal{M}' = \arg\min_{\mathcal{M}} \sum_{\mathbf{v}=1}^{N} \mathcal{L}(\mathbf{x}'_{\mathbf{v}}, \text{Rend}(\mathcal{M}, \pi_{\mathbf{v}})). \quad (4)$$

Following previous methods that reconstruct 3D from synthesized 2D images [17, 73], we use a mixture of $\mathcal{L}_1$ and $\mathcal{L}_{\text{LPIPS}}$ [85] for robust optimization. In practice, unlike iterative dataset updates (IDU) [20], we found that inferring the enhanced views $\mathcal{X}'$ once already yields high-quality results. More implementation details and results for this part are provided in the Appendix Sec. D.2.

4. Experiments

4.1. Datasets and Implementation

Datasets. For training, we use the Objaverse dataset [14], specifically leveraging the G-buffer Objaverse [47], which provides diverse and high-quality renderings on Objaverse instances. We construct LQ-HQ view pairs following the augmentation pipeline outlined in Sec. 3.3 and then split the dataset into separate training and testing sets. Overall, approximately 400K objects are used for training.

For evaluation, we use a test set containing 500 objects from different categories within our synthesized Objaverse datasets. For each object, we render four orthogonal views from elevation angles ranging from $-5°$ to $30°$. Besides, we evaluate our model on the zero-shot in-the-wild dataset by selecting images from the GSO dataset [16], the output of generative image diffusion models [48], and web-sourced content. These images are then processed using several novel view synthesis methods [36, 39, 41, 54] to create our in-the-wild testing set, containing a total of 400 instances.

Implementation Details. We employ PixArt-Σ [10], an efficient DiT model, as our backbone. Our model is trained on images with a resolution of 512 x 512. The AdamW optimizer [33] is used with a fixed learning rate of 2e-5. Our training is conducted over 10 days using 8 Nvidia A100-80G GPUs, with a batch size 256. For inference, we employ a DDIM sampler [56] with 20 steps and set the Classifier-Free Guidance (CFG) [23] scale to 4.5. During training, we randomly sample input views with azimuth angles ranging from $0°$ to $360°$ and elevation angles between $-5°$ and $30°$.

Baselines. To assess the effectiveness of our approach, we adopt two image enhancement models, RealESRGAN [69] and StableSR [64], along with two video enhancement models, RealBasicVSR [7] and Upscale-a-Video [91] as our baselines. For a fair comparison, we further finetune all these methods on the Objaverse dataset to minimize potential data domain discrepancies. During inference, since Real-ESRGAN, RealBasicVSR, and Upscale-a-Video by default produce images upscaled by a factor of $\times 4$, we resize their outputs to the a uniform resolution of 512×512 for comparison.

Metrics. We evaluate the effectiveness of our methods on two tasks: multi-view synthesis enhancement and 3D reconstruction improvement. We employ standard metrics including PSNR, SSIM, and LPIPS [85] on our synthetic dataset, along with non-reference metrics including FID

Figure 4. Qualitative comparisons of enhancing multi-view synthesis with RealBasicVSR[7] and Upscale-A-Video[91] on the in-the-wild dataset. Visually inspecting, 3DENHANCER yields sharp and consistent textures with intact semantics, such as the eyes of the girl in the mid row. **(Zoom-in for best view)**

Table 1. Quantitative comparisons of enhancing multi-view synthesis on the Objaverse synthetic dataset, the best and second-best results are marked in red and blue, respectively.

Method	PSNR↑	SSIM↑	LPIPS ↓
Input	26.15	0.9056	0.1257
Real-ESRGAN [69]	26.02	0.9185	0.0877
StableSR [64]	25.12	0.8914	0.1130
RealBasicVSR [7]	26.21	0.9212	0.0888
Upscale-A-Video [91]	25.57	0.8937	0.1153
3DEnhancer (Ours)	27.53	0.9265	0.0626

Table 2. Quantitative comparisons of enhancing multi-view synthesis on the in-the-wild dataset.

Method	MUSIQ↑	FID↓	IS↑
Generated MV	52.77	112.12	7.68 ± 0.86
Real-ESRGAN [69]	72.47	114.25	7.31 ± 0.89
StableSR [64]	70.43	111.53	7.59 ± 0.97
RealBasicVSR [7]	74.07	128.30	7.09 ± 0.87
Upscale-A-Video [91]	71.73	114.81	7.75 ± 0.96
3DEnhancer (Ours)	73.32	108.40	7.93 ± 1.11

[51], Inception Score [49], and MUSIQ [31] on the in-the-wild dataset. For FID computation, we use the rendered images from Objaverse to represent the real distribution. Besides the quantitative metrics, we present comprehensive qualitative results in the paper.

4.2. Comparisons

Enhancing Multi-view Synthesis. The output images from multi-view synthesis models often lack texture details or exhibit inconsistencies across views, as shown in Fig. 1. To demonstrate that 3DENHANCER can correct flawed textures and recover missing textures, we provide quantitative results on both the Objaverse synthetic dataset and the in-the-wild dataset in Tab. 1 and Tab. 2, respectively. Qualitative comparisons on both test sets are presented in Fig. 3 and Fig. 4. As can be seen, our method outperforms others across most metrics. While RealBasicVSR achieves a higher MUSIQ score on the in-the-wild dataset, it fails to generate visually plausible images, as shown in Fig. 4. The image enhancement models RealESRGAN and StableSR can recover textures to some extent in individual views, but they fail to maintain consistency across multiple views. Video enhancement models, such as RealBasicVSR and Upscale-A-Video, also fail to correct texture distortions effectively. For example, both models fail to generate smooth facial textures in the first example shown in Fig. 4. In contrast, our method generates more natural and consistent details across views.

Enhancing 3D Reconstruction. In this section, we present 3D reconstruction comparisons based on rendering views from 3D Gaussians generated by LGM [58]. Quantitative comparisons are shown in Tab. 3. The proposed method outperforms previous approaches in terms of 3D recon-

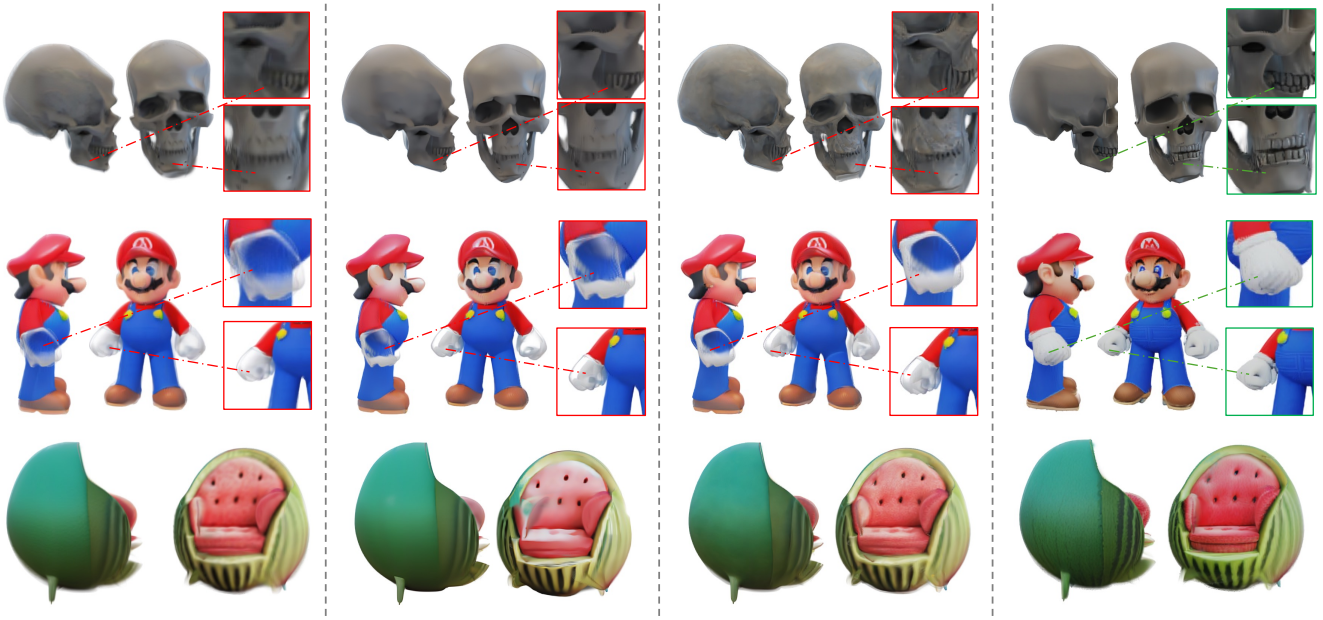


Figure 5. Qualitative comparisons of enhancing 3D reconstruction given generated multi-view images on the in-the-wild dataset. Due to the low quality and view-inconsistency of outputs from multi-view models, the reconstructed 3D models often suffer from issues such as blurry textures, like the skull's teeth in the first row, or artifacts such as ghosting hands in the second Mario example. Existing enhancement methods fail to correct texture artifacts. In contrast, our method produces both geometrically accurate and visually appealing results. **(Zoom-in for best view)**

Table 3. Quantitative comparisons of enhancing 3d reconstruction on the in-the-wild dataset.

Method	MUSIQ↑	FID↓	IS↑
Input	41.04	77.54	8.98 ± 0.65
Real-ESRGAN [69]	65.25	74.29	8.29 ± 0.35
StableSR [64]	65.71	72.98	9.51 ± 0.78
RealBasicVSR [7]	65.70	74.58	7.77 ± 0.48
Upscale-A-Video [91]	64.05	74.12	9.77 ± 0.79
3DEnhancer (Ours)	66.04	71.78	9.96 ± 0.96

Figure 6. Low-resolution 3D Gaussians optimization with 3DENHANCER. **(Zoom-in for best view)**

struction. For qualitative evaluation, we visualize the results of two video enhancement models, RealBasicVSR and Upscale-A-Video. As shown in Fig. 5, these baselines suffer from a lack of multi-view consistency, leading to misalignment, such as the misalignment of the teeth in the first skull example and the ghosting in the example of Mario's hand. In contrast, our model maintains consistency and produces high-quality texture details. We further demonstrate our approach can optimize a coarse differentiable representation. As shown in Fig. 6, our method is capable of refining low-resolution Gaussians [52]. More details and results of refining coarse Gaussians are provided in the Appendix Sec. D.2.

Table 4. Ablation study of cross-view modules.

Exp.	Multi-view Attn.	Epipolar Agg.	PSNR↑	SSIM↑	LPIPS ↓
(a)			25.11	0.9067	0.081
(b)		✓	25.95	0.9147	0.072
(c)	✓		26.92	0.9226	0.0642
(d)	✓	✓	**27.53**	**0.9265**	**0.0626**

4.3. Ablation Study

Effectiveness of Cross-View Modules. To evaluate the effectiveness of our proposed cross-view modules, we ablate two modules: multi-view row attention and near-view epipolar aggregation. As shown in Tab. 4, removing either module results in worse textures between views. The visual comparison in Fig. 7 also validates this observation. Without the multi-view row attention module, the model fails

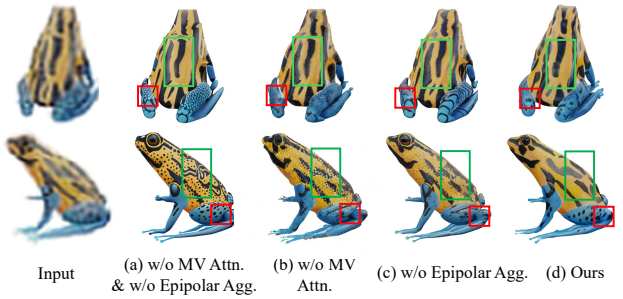


Figure 7. Effectiveness of cross-view modules.

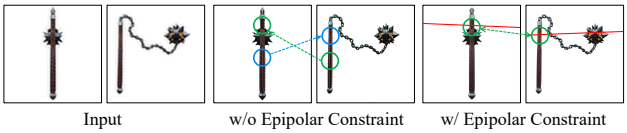


Figure 8. Comparisons of enhancing multiview images with and without epipolar aggregation. The red line denotes the epipolar line corresponding to the circled area, while the dotted arrow indicates the target location of tokens propagated from one view to another. **(Zoom-in for best view)**

to produce smooth textures, as shown in Fig. 7 (b). Without the epipolar aggregation module, reduced texture consistency is observed, as depicted in Fig. 7 (c).

Besides, the epipolar constraint is essential for preventing the model from learning textures from incorrect regions in other views and contributes to the overall consistency. As demonstrated in Fig. 8, without the epipolar constraint, the texture of the top part of the flail is incorrectly aggregated from the grip in the other view, thus resulting in inconsistency across views.

Effectiveness of Noise Level. As shown in Fig 1, our model can generate diverse textures by adjusting noise levels. Low noise levels generally result in outputs with blurred details, while high noise levels produce sharper, more detailed textures. However, high noise levels may also reduce the fidelity of the input images.

5. Conclusion

In conclusion, this work presents a novel 3D enhancement framework that leverages view-consistent latent diffusion model to improve the quality of given coarse multi-view images. Our approach introduces a versatile pipeline combining data augmentation, multi-view attention and epipolar aggregation modules that effectively enforces view consistency and refines textures across multi-view inputs. Extensive experiments and ablation studies demonstrate the superior performance of our method in achieving high-quality, consistent 3D content, significantly outperforming existing alternatives. This framework establishes a flexible and powerful solution for generic 3D enhancement, with broad applications in 3D content generation and editing.

References

[1] Raphael Bensadoun, Yanir Kleiman, Idan Azuri, Omri Harosh, Andrea Vedaldi, Natalia Neverova, and Oran Gafni. Meta 3d texturegen: Fast and consistent texture generation for 3d objects. *arXiv preprint arXiv:2407.02430*. 3

[2] Raphael Bensadoun, Tom Monnier, Yanir Kleiman, Filippos Kokkinos, Yawar Siddiqui, Mahendra Kariya, Omri Harosh, Roman Shapovalov, Benjamin Graham, Emilien Garreau, et al. Meta 3D Gen. *arXiv preprint arXiv:2407.02599*, 2024. 3

[3] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable Video Diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 2, 3

[4] Kelvin C.K. Chan, Xintao Wang, Xiangyu Xu, Jinwei Gu, and Chen Change Loy. GLEAN: Generative latent bank for large-factor image super-resolution. In *CVPR*, 2021. 3

[5] Kelvin C.K. Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. BasicVSR: The search for essential components in video super-resolution and beyond. In *CVPR*, 2021. 3

[6] Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Improving video super-resolution with enhanced propagation and alignment. In *CVPR*, 2022. 3

[7] Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Investigating tradeoffs in real-world video super-resolution. In *CVPR*, 2022. 3, 6, 7, 8, 16, 17, 18

[8] Chaofeng Chen, Shangchen Zhou, Liang Liao, Haoning Wu, Wenxiu Sun, Qiong Yan, and Weisi Lin. Iterative token evaluation and refinement for real-world super-resolution. In *ACM MM*, 2023. 3

[9] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *CVPR*, 2021. 3

[10] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-Σ: Weak-to-strong training of diffusion transformer for 4K text-to-image generation. *arXiv preprint arXiv:2403.04692*, 2024. 3, 4, 6, 14

[11] Minghao Chen, Iro Laina, and Andrea Vedaldi. DGE: Direct gaussian 3D editing by consistent multi-view editing. *arXiv preprint arXiv:2404.18929*, 2024. 2, 5

[12] Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. Activating more pixels in image super-resolution transformer. In *CVPR*, 2023. 3

[13] Yongwei Chen, Yushi Lan, Shangchen Zhou, Tengfei Wang, and XIngang Pan. SAR3D: Autoregressive 3D object generation and understanding via multi-scale 3D VQVAE. *arXiv preprint arXiv:2411.16856*, 2024. 2, 3

[14] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3D objects. *CVPR*, 2023. 2, 3, 6

[15] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte,

Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3D objects. In *NeurIPS*, 2024. 2

[16] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3D scanned household items. In *ICRA*, 2022. 6

[17] Ruiqi Gao*, Aleksander Holynski*, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul P. Srinivasan, Jonathan T. Barron, and Ben Poole*. Cat3d: Create anything in 3d with multi-view diffusion models. *NeurIPS*, 2024. 6, 17

[18] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. TokenFlow: Consistent diffusion features for consistent video editing. *ICLR*, 2024. 2, 5

[19] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 2

[20] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-NeRF2NeRF: Editing 3d scenes with instructions. In *CVPR*, 2023. 6

[21] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 4

[22] Zexin He and Tengfei Wang. OpenLRM: Open-source large reconstruction models. https://github.com/3DTopia/OpenLRM, 2023. 5

[23] Jonathan Ho. Classifier-free diffusion guidance. In *NeurIPS*, 2021. 6, 15

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2, 3

[25] Fangzhou Hong, Zhaoxi Chen, Yushi Lan, Liang Pan, and Ziwei Liu. EVA3D: Compositional 3D human generation from 2d image collections. In *ICLR*, 2022. 5

[26] Fangzhou Hong, Jiaxiang Tang, Ziang Cao, Min Shi, Tong Wu, Zhaoxi Chen, Tengfei Wang, Liang Pan, Dahua Lin, and Ziwei Liu. 3dtopia: Large text-to-3d generation model with hybrid diffusion priors. *arXiv preprint arXiv:2403.02234*, 2024. 15

[27] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. In *ICLR*, 2024. 2

[28] Zehuan Huang, Hao Wen, Junting Dong, Yaohui Wang, Yangguang Li, Xinyuan Chen, Yan-Pei Cao, Ding Liang, Yu Qiao, Bo Dai, et al. EpiDiff: Enhancing multi-view synthesis via localized epipolar-constrained diffusion. In *CVPR*, 2024. 2, 4

[29] Jpcy. Jpcy/xatlas: Mesh parameterization / uv unwrapping library. 3

[30] Heewoo Jun and Alex Nichol. Shap-E: Generating conditional 3D implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 2

[31] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. MUSIQ: Multi-scale image quality transformer. In *ICCV*, 2021. 7

[32] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42(4):1–14, 2023. 2, 3, 4, 5, 17

[33] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[34] Yushi Lan, Shangchen Zhou, Zhaoyang Lyu, Fangzhou Hong, Shuai Yang, Bo Dai, Xingang Pan, and Chen Change Loy. GaussianAnything: Interactive point cloud latent diffusion for 3D generation. 2, 3

[35] Yushi Lan, Fangzhou Hong, Shuai Yang, Shangchen Zhou, Xuyi Meng, Bo Dai, Xingang Pan, and Chen Change Loy. LN3Diff: Scalable latent neural fields diffusion for speedy 3D generation. In *ECCV*, 2024. 3

[36] Peng Li, Yuan Liu, Xiaoxiao Long, Feihu Zhang, Cheng Lin, Mengfei Li, Xingqun Qi, Shanghang Zhang, Wenhan Luo, Ping Tan, et al. Era3D: High-resolution multiview diffusion using efficient row-wise attention. *NeurIPS*, 2024. 2, 4, 6, 15

[37] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using swin transformer. In *ICCV*, 2021. 3

[38] Jingyun Liang, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu Timofte, and Luc Van Gool. Recurrent video restoration transformer with guided deformable attention. In *NeurIPS*, 2022. 3

[39] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3D object. In *CVPR*, 2023. 2, 3, 5, 6

[40] Yuxin Liu, Minshan Xie, Hanyuan Liu, and Tien-Tsin Wong. Text-guided texturing by synchronized multi-view diffusion. *arXiv preprint arXiv:2311.12891*, 2023. 3

[41] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. In *ICLR*, 2024. 5, 6, 15

[42] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3D using cross-domain diffusion. In *CVPR*, 2024. 2, 4

[43] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2, 4, 5

[44] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 3, 4, 14

[45] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *arXiv*, 2023. 14

[46] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. *ICLR*, 2022. 2

[47] Lingteng Qiu, Guanying Chen, Xiaodong Gu, Qi Zuo, Mutian Xu, Yushuang Wu, Weihao Yuan, Zilong Dong, Liefeng Bo, and Xiaoguang Han. Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d. In *CVPR*, 2024. 2, 6, 15

[48] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 3, 6, 14

[49] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016. 7

[50] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 2

[51] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, 2020. Version 0.3.0. 7

[52] Yuan Shen, Duygu Ceylan, Paul Guerrero, Zexiang Xu, Niloy J. Mitra, Shenlong Wang, and Anna Frühstück. SuperGaussian: Repurposing video models for 3D super resolution. In *ECCV*, 2024. 2, 3, 8, 17

[53] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023. 2

[54] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. MVDream: Multi-view diffusion for 3D generation. In *ICLR*, 2024. 2, 3, 4, 5, 6, 14

[55] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *NeurIPS*, 2021. 3

[56] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 6

[57] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 2

[58] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. LGM: Large multi-view gaussian model for high-resolution 3D content creation. In *ECCV*, 2024. 2, 3, 5, 7, 16

[59] Jiaxiang Tang, Ruijie Lu, Xiaokang Chen, Xiang Wen, Gang Zeng, and Ziwei Liu. Intex: Interactive text-to-texture synthesis via unified depth-aware inpainting. *arXiv preprint arXiv:2403.11878*, 2024. 3

[60] Anju Tewari, Otto Fried, Justus Thies, Vincent Sitzmann, S. Lombardi, Z Xu, Tanaba Simon, Matthias Nießner, Edgar Tretschk, L. Liu, Ben Mildenhall, Pranatharthi Srinivasan, R. Pandey, Sergio Orts-Escolano, S. Fanello, M. Guang Guo, Gordon Wetzstein, J y Zhu, Christian Theobalt, Manju Agrawala, Donald B. Goldman, and Michael Zollhöfer. Advances in neural rendering. *Computer Graphics Forum*, 41, 2021. 2

[61] Hung-Yu Tseng, Qinbo Li, Changil Kim, Suhib Alsisan, Jia-Bin Huang, and Johannes Kopf. Consistent view synthesis with pose-guided diffusion models. In *CVPR*, 2023. 2, 4

[62] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *NeurIPS*, 2021. 3

[63] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017. 5

[64] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin C. K. Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. In *IJCV*, 2024. 3, 6, 7, 8, 16, 17, 18

[65] Peng Wang and Yichun Shi. ImageDream: Image-prompt multi-view diffusion for 3D generation. *arXiv preprint arXiv:2312.02201*, 2023. 2, 3, 15

[66] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. ESRGAN: Enhanced super-resolution generative adversarial networks. In *ECCVW*, 2018. 3

[67] Xintao Wang, Kelvin C.K. Chan, Ke Yu, Chao Dong, and Chen Change Loy. EDVR: Video restoration with enhanced deformable convolutional networks. In *CVPRW*, 2019. 3

[68] Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. Towards real-world blind face restoration with generative facial prior. In *CVPR*, 2021. 3

[69] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data. In *ICCVW*, 2021. 2, 3, 5, 6, 7, 8, 16, 18

[70] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3D generation with variational score distillation. In *NeurIPS*, 2023. 2

[71] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. CRM: Single image to 3D textured mesh with convolutional reconstruction model. In *ECCV*, 2024. 2, 5

[72] Kailu Wu, Fangfu Liu, Zhihan Cai, Runjie Yan, Hanyang Wang, Yating Hu, Yueqi Duan, and Kaisheng Ma. Unique3D: High-quality and efficient 3D mesh generation from a single image. *arXiv preprint arXiv:2405.20343*, 2024. 3, 18

[73] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P. Srinivasan, Dor Verbin, Jonathan T. Barron, Ben Poole, and Aleksander Holynski. Reconfusion: 3d reconstruction with diffusion priors. In *CVPR*, 2024. 6

[74] Liu Xi, Zhou Chaoyi, and Huang Siyu. 3DGS-Enhancer: Enhancing unbounded 3D gaussian splatting with view-consistent 2d diffusion priors. *NeurIPS*, 2024. 2, 3

[75] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. InstantMesh: Efficient 3D mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. 2

[76] Yiran Xu, Taesung Park, Richard Zhang, Yang Zhou, Eli Shechtman, Feng Liu, Jia-Bin Huang, and Difan Liu. VideoGigaGAN: Towards detail-rich video super-resolution. *arXiv preprint arXiv:2404.12388*, 2024. 2, 3

[77] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, and Kai Zhang. DMV3D: Denoising multi-view diffusion using 3D large reconstruction model. In *ICLR*, 2024. 3

[78] Fan Yang, Jianfeng Zhang, Yichun Shi, Bowen Chen, Chenxu Zhang, Huichao Zhang, Xiaofeng Yang, Jiashi Feng, and Guosheng Lin. Magic-boost: Boost 3d generation with mutli-view conditioned diffusion. *arXiv preprint arXiv:2404.06429*, 2024. 3

[79] Tao Yang, Peiran Ren, Xuansong Xie, and Lei Zhang. Gan prior embedded network for blind face restoration in the wild. In *CVPR*, 2021. 3

[80] Xu Yinghao, Shi Zifan, Yifan Wang, Chen Hansheng, Yang Ceyuan, Peng Sida, Shen Yujun, and Wetzstein Gordon. GRM: Large gaussian reconstruction model for efficient 3D reconstruction and generation. *arXiv preprint arXiv:2403.14621*, 2024. 5

[81] Biao Zhang, Jiapeng Tang, Matthias Nießner, and Peter Wonka. 3DShape2VecSet: A 3D shape representation for neural fields and generative diffusion models. *ACM TOG*, 42 (4), 2023. 2

[82] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *ICCV*, 2021. 3

[83] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 4, 14, 18

[84] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. CLAY: A controllable large-scale generative model for creating high-quality 3D assets. *ACM TOG*, 2024. 2, 3

[85] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6, 16, 17

[86] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018. 3

[87] Shangchen Zhou, Jiawei Zhang, Jinshan Pan, Haozhe Xie, Wangmeng Zuo, and Jimmy Ren. Spatio-temporal filter adaptive network for video deblurring. In *ICCV*, 2019. 3

[88] Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, and Chen Change Loy. Cross-scale internal graph neural network for image super-resolution. In *NeurIPS*, 2020. 3

[89] Shangchen Zhou, Kelvin CK Chan, Chongyi Li, and Chen Change Loy. Towards robust blind face restoration with codebook lookup transformer. In *NeurIPS*, 2022. 3

[90] Shangchen Zhou, Chongyi Li, and Chen Change Loy. LEDNet: Joint low-light enhancement and deblurring in the dark. In *ECCV*, 2022. 3

[91] Shangchen Zhou, Peiqing Yang, Jianyi Wang, Yihang Luo, and Chen Change Loy. Upscale-A-Video: Temporal-consistent diffusion model for real-world video super-resolution. In *CVPR*, 2024. 3, 5, 6, 7, 8, 16, 17, 18

Appendix

In this appendix, we provide additional discussions and results to supplement the main paper. In Sec. A, we present more architecture and design details of our 3DENHANCER. In Sec. B, we provide detailed information about our training dataset, including the augmentation pipeline and illustrative examples. Sec. C highlights some interesting findings related to inference. Comprehensive results are presented in Sec. D to further demonstrate our performance. Notably, we also include a demo video (Sec. D.4) to showcase rendering results for 3D reconstruction enhancement.

Contents

A. Architecture and Design

A.1. Pose-aware Encoder

Our pose-aware encoder is adapted from the convolutional encoder of LDM [48]. As shown in Fig. 2, the output of the pose-aware encoder serves as the conditioning features for the trainable copies in our ControlNet [83]. The details of its hyperparameters are summarized in Tab. 5. This encoder employs 64 channels and a single residual block to enhance efficiency. Additionally, we incorporate cross-view self-attention [54] into the middle layer of the encoder to improve inter-view consistency. To ensure compatibility with the number of latent channels in the DiT blocks, the output z-channels number is set to 1152. The final convolutional layer in the encoder uses a stride of 2 to match the dimensions of the DiT block latents. All other hyperparameters are kept at default values.

A.2. View-Consistent DiT Block

The view-consistent DiT block is based on the PixArt-Σ [10] architecture. Consistent with PixArt-Σ, we use the T5 large language model as the text encoder for conditional text feature extraction, and the frozen VAE from SDXL [45] to capture the latent features of images. PixArt-Σ consists of 28 Transformer blocks. For the ControlNet [83] implementation, we utilize trainable copies of the first 13 base blocks, augmenting each copied block with zero linear layers before and after it. The output of the i-th trainable copied block is added to the corresponding frozen base i-th block. The multi-view row attention with near-view epipolar aggregation is an additional attention layer that is inserted into both the DiT blocks and the copied ControlNet blocks. This layer is positioned after the self-attention layer, as illustrated in Fig. 2. During training, we train the entire ControlNet blocks and every inserted multi-view row attention layer in the DiT blocks. Detailed hyperparameters for the DiT block and the inserted row attention layers are provided in Tab. 5.

A.3. Weight for Two Nearest Views Aggregation

In Eq. 3, we compute the fusion weight w based on both the physical camera distance and the similarity of token features. First, we consider the geometric distance weight w_d, which reflects the proximity of the camera:

$$w_d = \frac{d_{\mathbf{v},\mathbf{v}+1}}{d_{\mathbf{v},\mathbf{v}-1} + d_{\mathbf{v},\mathbf{v}+1}}, \tag{5}$$

where $d_{\mathbf{v},\mathbf{k}}$ represents the geometric distance between the camera of view v and the camera of view $\mathbf{k} \in \{\mathbf{v}-1, \mathbf{v}+1\}$. To ensure the nearest-view weight calculation also incorporates token feature similarity, we augment the weight token-wise with token similarity:

$$w = \frac{S^i_{\mathbf{v},\mathbf{v}-1} \cdot w_d}{S^i_{\mathbf{v},\mathbf{v}-1} \cdot w_d + (1 - w_d) \cdot S^i_{\mathbf{v},\mathbf{v}+1}}, \tag{6}$$

where $S^i_{\mathbf{v},\mathbf{k}}$ denotes the cosine similarity of the corresponding tokens, *i.e.*, $\mathbf{f}_{\mathbf{v}}[i]$ and $\mathbf{f}_{\mathbf{k}}[M_{\mathbf{v},\mathbf{k}}[i]]$.

Table 5. Hyperparameters for the pose-aware encoder, view-consistent DiT block, and the inserted multi-view row attention layers in our 3DENHANCER. The table follows the hyperparameter table style from [44, 48]. We train our model on images with a resolution of 512×512 using 4 views.

Hyperparameter	*DiT*
Layers	28
Training views shape	$4 \times 512 \times 512 \times 3$
f	8
Patch size	2
Embedding dimension	1024
Hidden size	1152
z-shape	$4 \times 1024 \times 1152$
Head number	16
CA sequence length	300

Hyperparameter	*Pose-aware Encoder*
f	8
Channels	64
Channel multiplier	1, 2, 4, 4
z-channels	1152

Hyperparameter	*Row Attention*
Head number	16
Positional encoding	sine-cosine
Epipolar aggregation	True

B. Dataset

B.1. Dataset

The G-buffer Objaverse dataset [47] contains a broad variety of 3D objects categorized into 10 types: Human-Shaped, Animals, Daily Objects, Furniture, Buildings and Outdoor Objects, Transportation, Plants, Food, and Electronics. To ensure high standards, we exclude any objects labeled as "Poor-quality." We observe that the original captions in G-buffer Objaverse are simple and lack detailed information. Therefore, we adopt captions from 3D-Topia [26], which provide more informative and accurate descriptions for a subset of objects in Objaverse. We update the caption of each object accordingly if it exists in 3D-Topia, resulting in the refinement of approximately 45% of the captions. Additionally, to facilitate CFG [23], we omit the text condition at a rate of 0.2. Such settings enhance the robustness of our method to text conditions with varying levels of detail. For the in-the-wild dataset, we remove backgrounds and center objects as previous works [36, 41, 65]. We uniformly apply a white background to the input views.

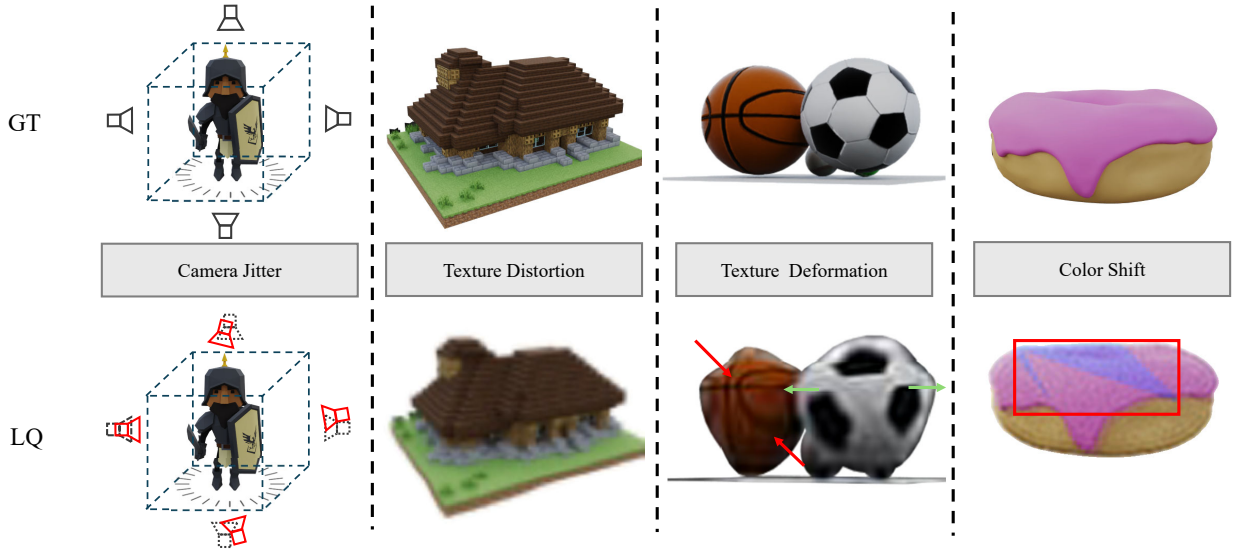


Figure 9. Visualization of several examples from our augmentation pipeline. Thanks to the comprehensive augmentation strategy, our method is able to bridge the domain gap between training and inference.

B.2. Data Augmentation

The visualization of the data augmentation pipeline is shown in Fig. 9. During training, we dynamically generate synthetic training pairs on the fly, and the argumentation is implemented in PyTorch with CUDA acceleration to ensure efficiency. The pipeline incorporates several stochastic augmentation steps, producing diverse training pairs with varying levels of degradation. During augmentation, the input views of the same object are either augmented with the same level of degradation (e.g., the same blur kernel) or with different stochastic augmentations. This strategy encourages the model's ability to learn information across views, particularly from those with fewer degradations. We ensure that the augmentation is confined to the object's masked area with a slight mask dilation. This allows the white background unaffected, which aligns with real-world scenarios of low-quality multi-view images. We also set a probability where no augmentation is applied to the input images, i.e., the low-quality images are identical to the ground truth. In such cases, the model is encouraged to preserve fidelity when the input images are already of high-quality. Details of several augmentation parameters are summarized in Tab. 6. Further implementation details will be provided in our code release.

C. More Details on Inference

C.1. Multi-View Editing

Benefiting from our comprehensive augmentation pipeline and the robust view-consistent DiT Block, we observe an interesting fact: our method is capable of generating detailed and consistent textures even from extremely coarse or corrupted

Table 6. Several augmentation parameters that are used in our augmentation pipeline.

Argumentation type	Parameters
First-order blur prob	0.8
Second-order blur prob	0.3
Blur kernel size range	{7, 9, ..., 21}
Blur standard deviation range	[0.2, 3]
Gaussian noises prob	0.5
Resize range	[0.3, 1.5]
JPEG compression quality factor	[80, 100]

Argumentation type	Parameters
Final sinc filter prob	0.8
Camera jitter prob	0.2
Camera jitter strength range	[0.05, 0.1]
Color shift prob	0.3
Grid distortion prob	0.3
Grid distortion strength range	[0.2, 0.5]
No argumentation prob	0.1

multi-view inputs. This enables our approach to modify multi-view images in two distinct ways: 1. Applying a black mask to the region designated for editing and modifying the text prompt to generate the target multi-view images. 2. Adjusting the inference noise level, where higher noise levels produce more diverse outputs. Using the edited multi-view images, we can subsequently modify the reconstructed 3D representations. An example of editing 3D Gaussians generated by LGM [58] through modifying its multi-view input is shown in Fig. 10.

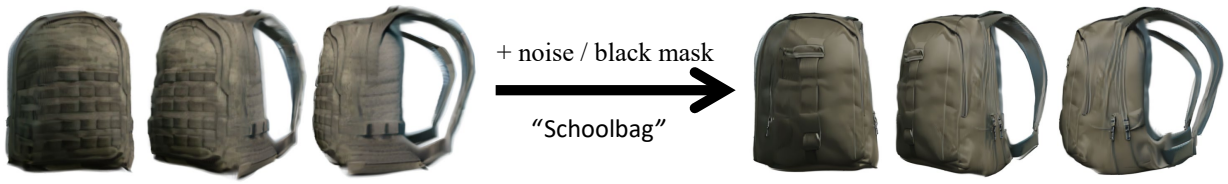


Figure 10. Rendered views of edited 3D Gaussians using our multi-view editing approach. By adding a large noise or a black mask, and leveraging text prompts as guidance, we consistently modify the texture of the bags.

C.2. Color Correction

Previous studies [64, 91] have highlighted that diffusion models often exhibit color shift artifacts, where the global color scheme deviates from the input images. This is different from our color shift augmentation, which introduces localized color changes to specific image regions. However, this augmentation also aims to encourage the model to maintain consistent color reproduction. We observe that integrating a training-free wavelet color correction module [64] can help resolve the global color scheme shift. As reported in Tab. 8, applying wavelet color correction leads to improved fidelity metrics (higher PSNR, SSIM, and lower LPIPS [85]) for the baseline, but it has minimal impact on our results, showing our robustness against global color scheme shifts. However, at extremely high noise levels, such as $\delta = 200$, minor global color shifts may still occur in our method because the noise may impact the original color information. In such cases, wavelet color correction could be beneficial, as illustrated in Fig. 11.

D. More Results

D.1. User Study

To enable a thorough comparison, we conduct a user study to evaluate the enhancement results of multi-view images and 3D reconstructions. For the multi-view image enhancement, each participant is shown 10 sets of randomly selected objects' multi-view images, enhanced by our 3DENHANCER, RealESRGAN [69], StableSR [64], RealBasicVSR [7], and Upscale-a-Video [91]. For the 3D reconstruction enhancement, participants are presented with another 10 360-degree rotating render videos of the 3D Gaussians enhanced by our method, RealBasicVSR [7], and Upscale-a-Video [91]. Their task is to choose the visually superior enhanced results. A total of 20 participants take part in the study. As illustrated in Fig. 12, The results indicate a strong preference for our method over the compared approach. On average, 74% of users preferred our method for enhancing multi-view images, while 78% favored it for enhancing 3D reconstruction. These findings strongly demonstrate the quality and robustness of our approach.

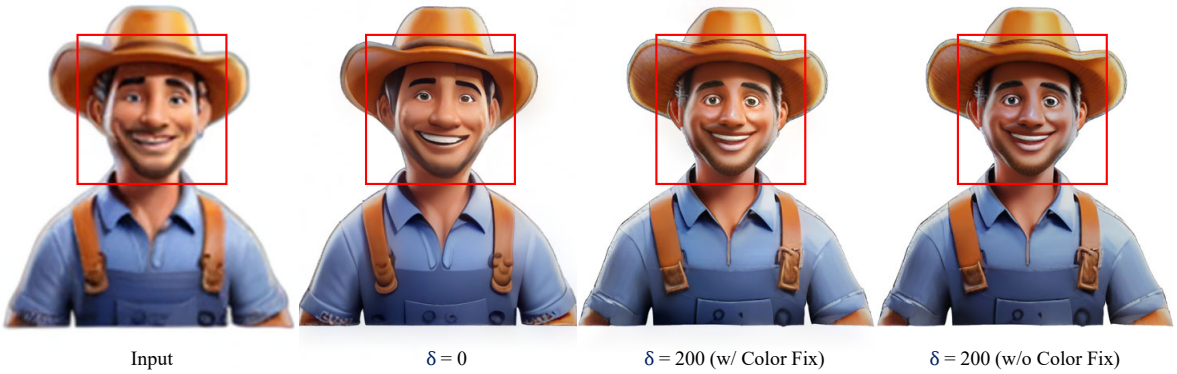


Figure 11. Minor global color scheme shift at high noise levels. When the noise level δ is small, such as $\delta = 0$, our method maintains excellent color fidelity. However, at a higher noise level, such as $\delta = 200$ in the example, the output figure's face appears slightly darker than that of the input. In this case, the wavelet color correction [64] could help mitigate this issue.

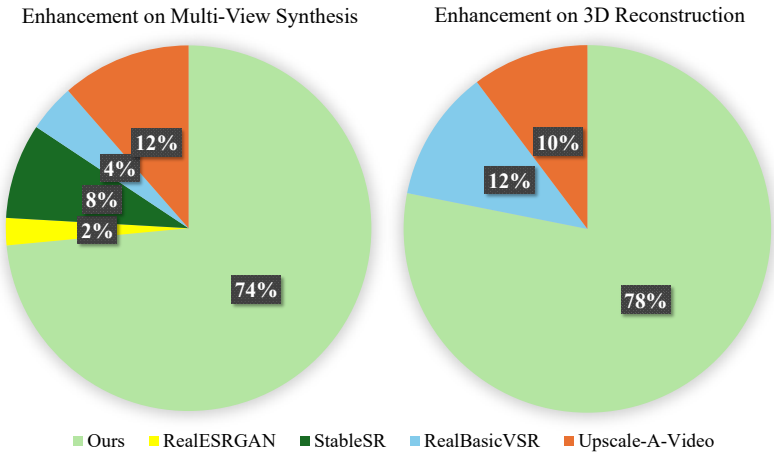


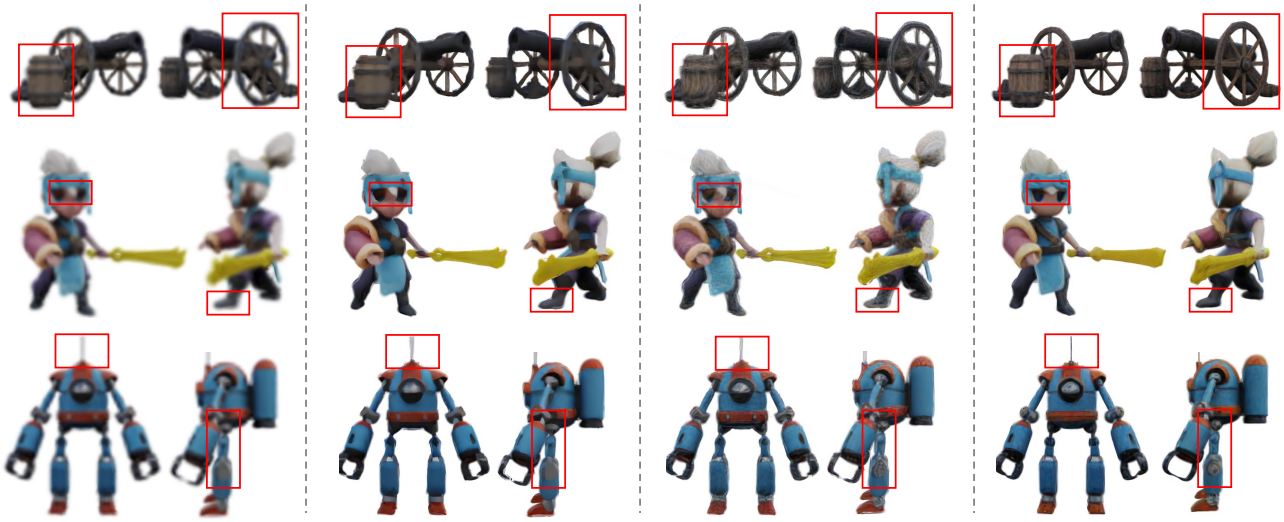
Figure 12. User study results. Human voters consistently prefer our method over other approaches.

D.2. Results of Optimizing 3D Gaussians

3D representations can be rendered from multiple views, this nature allows our method to iteratively optimize a coarse 3D representations. To demonstrate this capability, we adopt Gaussian Splatting [32] as our example due to its high rendering fidelity and efficiency. Specifically, we implement a pipeline to refine coarse 3D Gaussians checkpoints by leveraging our enhanced outputs as pseudo ground truth. We randomly select 20 objects from the Objaverse test dataset for evaluation. Following [52], we fit low-resolution 3D Gaussians using images obtained by bilinearly downsampling the original dataset images by a factor of 8, resulting in a resolution of 64×64 pixels. We use three distinct trajectories for fitting low-resolution Gaussians, refining Gaussians, and evaluation. As proposed in [17], our refinement process also minimizes a combined loss function, including a photometric reconstruction loss and a perceptual loss [85]. The perceptual loss emphasizes high-level semantic similarity between rendered and enhanced images while ignoring inconsistencies in low-level, high-frequency details. To improve regularization during refining, we sample 100 views along a single smooth orbital path, as increasing the number of views has been shown to enhance the refining process [17]. The optimization is conducted over 2000 refinement steps for all methods and takes approximately 130s to refine a single object on one NVIDIA A100 GPU. For comparison, we evaluate our method against two video enhancement models, RealBasicVSR [7] and Upscale-A-Video [91]. Quantitative and qualitative results are presented in Tab. 7 and Fig. 13, respectively. Our results demonstrate detailed and sharp outputs, while other methods exhibit ghosting artifacts and blurry textures. The results highlight the superior performance of our approach in refining coarse 3D representations.

Table 7. Quantitative comparison of optimizing low-resolution Gaussians. The best results are highlighted in **bold**.

Metrics	Low-Resolution Gaussians	RealBasicVSR [7]	Upscale-A-Video[91]	3DENHANCER
PSNR ↑	26.35	27.39	26.20	**27.54**
SSIM ↑	0.9120	0.9216	0.9184	**0.9337**
LPIPS ↓	0.1135	0.0803	0.0928	**0.0756**



Low-Resolution Gaussians RealBasicVSR Upscale-A-Video 3DEnhancer (Ours)

Figure 13. Qualitative comparisons of optimizing low-resolution Gaussians. During optimization, both RealBasicVSR [7] and Upscale-A-Video [91] produce ghosting and blurry textures due to inconsistent outputs. Our 3DENHANCER achieves sharp and clear results.

D.3. More Comparisons

In this section, we introduce another baseline from the multi-view image upscale module in Unique3D [72]. This baseline finetunes ControlNet-Tile [83] to enhance RGB views. While the module can sharpen some textures, it struggles to recover inconsistent or corrupted areas in multi-view images. Our method outperforms Unique3D's MV Upscale both quantitatively and qualitatively. The quantitative comparison between Unique3D's MV Upscale and our method is presented in Tab. 8. Additionally, we provide more visual comparisons of our method with all other baselines, including RealESRGAN [69], StableSR [64], Unique3D's MV Upscale [72], RealBasicVSR [7], and Upscale-a-Video [91]. Fig. 14 and Fig. 15 showcase the visual comparisons of multi-view enhancement on synthetic and in-the-wild datasets, respectively.

Table 8. Quantitative comparison of enhancing multi-view synthesis on the Objaverse synthetic dataset with Unique3D's MV Upscale module. Our method demonstrates clear advantages in restoration fidelity, as measured by PSNR, SSIM, and LPIPS. While applying color correction improves the output of Unique3D's MV Upscale module, it has minimal impact on our results when noise level is set to 0, highlighting our method's robustness against global color scheme shift issues.

Metrics	Unique3D's Upscale	Unique3D's Upscale (+ color fix)	**3DENHANCER**	**3DENHANCER (+ color fix)**
PSNR ↑	25.75	26.18	27.53	27.50
SSIM ↑	0.8989	0.9055	0.9265	0.9258
LPIPS ↓	0.1300	0.1257	0.0626	0.0631

D.4. Video Demo

We also provide a demo video (3DEnhancer-demo.mp4) in our project page, showcasing visual results of 3D reconstruction enhancement.

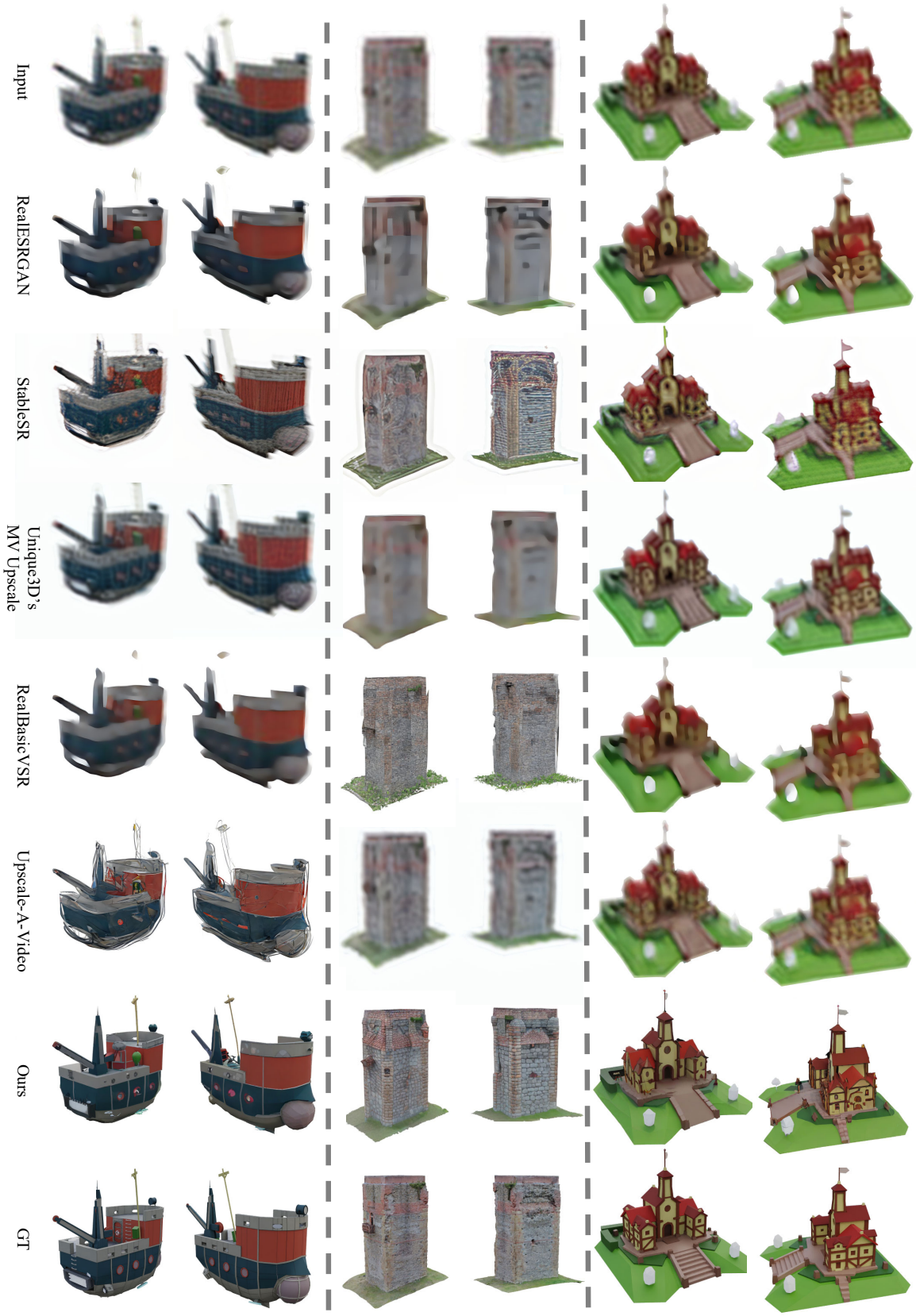


Figure 14. Qualitative comparisons on the Objaverse synthetic dataset. Our 3DENHANCER demonstrates promising improvements, with increased detail and enhanced realism. (**Zoom in for best view.**)

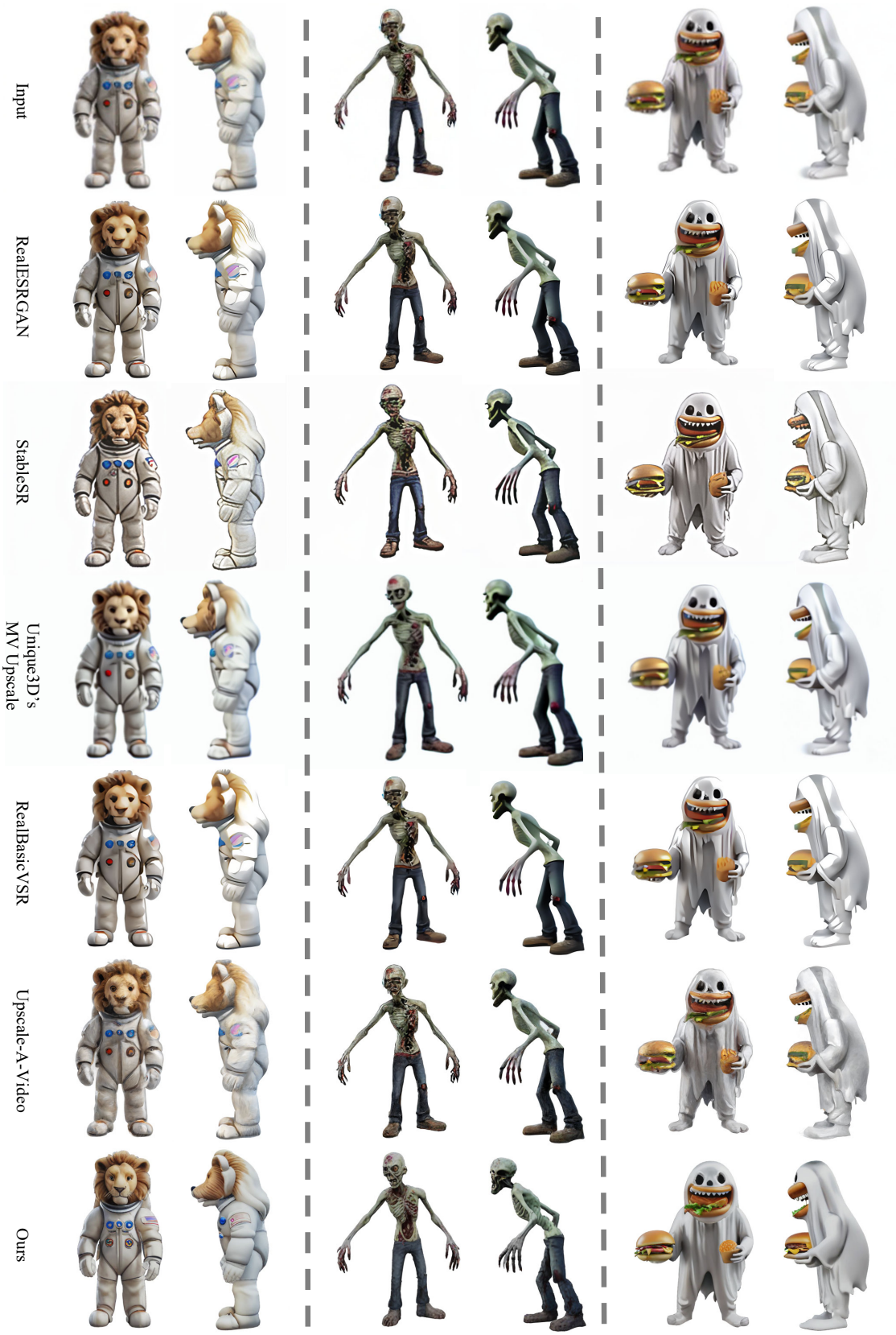


Figure 15. Qualitative comparisons on the in-the-wild dataset. Our 3DENHANCER yields significant improvements, providing enhanced detail and consistent output. (**Zoom in for best view.**)