

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

— Supplementary Materials —

Yihang Luo Shangchen Zhou[†] Yushi Lan Xingang Pan Chen Change Loy[†]

S-Lab, Nanyang Technological University

<https://yihangluo.com/projects/3DEnhancer>

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. More results and comparisons are presented in Sec. **D** to further demonstrate our performance. We also include a demo video (Sec. **D.6**) to showcase rendering results for 3D reconstruction enhancement.

Contents

A Architecture and Design	2
A.1 Pose-aware Encoder	2
A.2 View-Consistent DiT Block	2
A.3 Weight for Two Nearest Views Aggregation	2
B Dataset	3
B.1 Dataset	3
B.2 Data Augmentation	3
C More Details on Inference	3
C.1 Multi-View Editing	3
C.2 Color Correction	4
D More Results	5
D.1 User Study	5
D.2 Results of Optimizing 3D Gaussians	6
D.3 Results of Generalization to Real-World Objects	6
D.4 Results of Further Fine-tuning Upscale-A-Video	7
D.5 More Comparisons	8
D.6 Video Demo	8

[†]Corresponding authors.

A. Architecture and Design

A.1. Pose-aware Encoder

Our pose-aware encoder is adapted from the convolutional encoder of LDM [12]. As shown in Fig. 2, the output of the pose-aware encoder serves as the conditioning features for the trainable copies in our ControlNet [21]. The details of its hyperparameters are summarized in Tab. A. This encoder employs 64 channels and a single residual block to enhance efficiency. Additionally, we incorporate cross-view self-attention [14] 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 [2] 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 [10] to capture the latent features of images. PixArt- Σ consists of 28 Transformer blocks. For the ControlNet [21] 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. A.

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:

$$w_d = \frac{d_{\mathbf{v}, \mathbf{v}+1}}{d_{\mathbf{v}, \mathbf{v}-1} + d_{\mathbf{v}, \mathbf{v}+1}}, \quad (\text{A})$$

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:

$$w = \frac{S_{\mathbf{v}, \mathbf{v}-1}^i \cdot w_d}{S_{\mathbf{v}, \mathbf{v}-1}^i \cdot w_d + (1 - w_d) \cdot S_{\mathbf{v}, \mathbf{v}+1}^i}, \quad (\text{B})$$

where $S_{\mathbf{v}, \mathbf{k}}^i$ 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 A. 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 [9, 12]. We train our model on images with a resolution of 512×512 using 4 views.

Hyperparameter	<i>DiT</i>	Hyperparameter	<i>Pose-aware Encoder</i>
Layers	28	f	8
Training views shape	$4 \times 512 \times 512 \times 3$	Channels	64
f	8	Channel multiplier	1, 2, 4, 4
Patch size	2	z -channels	1152
Embedding dimension	1024		
Hidden size	1152	Hyperparameter	<i>Row Attention</i>
z -shape	$4 \times 1024 \times 1152$	Head number	16
Head number	16	Positional encoding	sine-cosine
CA sequence length	300	Epipolar aggregation	True

B. Dataset

B.1. Dataset

The G-buffer Objaverse dataset [11] 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 [5], 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 [4], 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 [7, 8, 17]. We uniformly apply a white background to the input views.

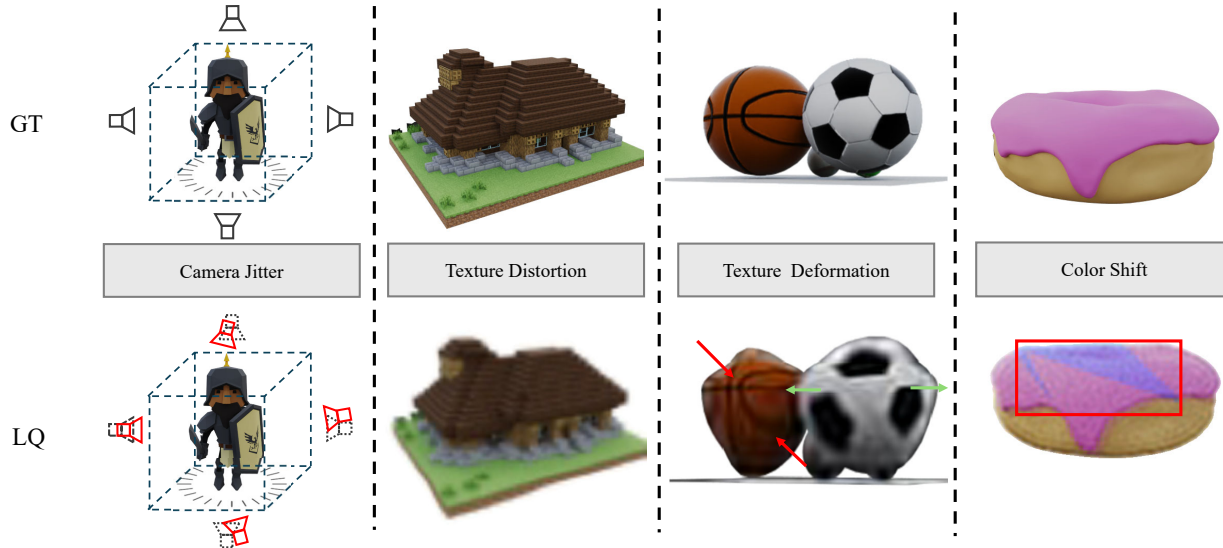


Figure A. Visualization of several examples from our augmentation pipeline. Thanks to the comprehensive augmentation strategy, our method is able to bridge the domain gap between training and inference.

B.2. Data Augmentation

The visualization of the data augmentation pipeline is shown in Fig. A. During training, we dynamically generate synthetic training pairs on the fly, and the augmentation 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. B. 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 B. Several augmentation parameters that are used in our augmentation pipeline.

Argumentation type	Parameters	Argumentation type	Parameters
First-order blur prob	0.8	Final sinc filter prob	0.8
Second-order blur prob	0.3	Camera jitter prob	0.2
Blur kernel size range	{7, 9, ..., 21}	Camera jitter strength range	[0.05, 0.1]
Blur standard deviation range	[0.2, 3]	Color shift prob	0.3
Gaussian noises prob	0.5	Grid distortion prob	0.3
Resize range	[0.3, 1.5]	Grid distortion strength range	[0.2, 0.5]
JPEG compression quality factor	[80, 100]	No argumentation prob	0.1

multi-view inputs. As shown in Fig. B, our method effectively handles various challenging cases, including multi-views with (a) *extremely blurred textures*, (b) *masked or missing parts*, and (c) *significant noise*.

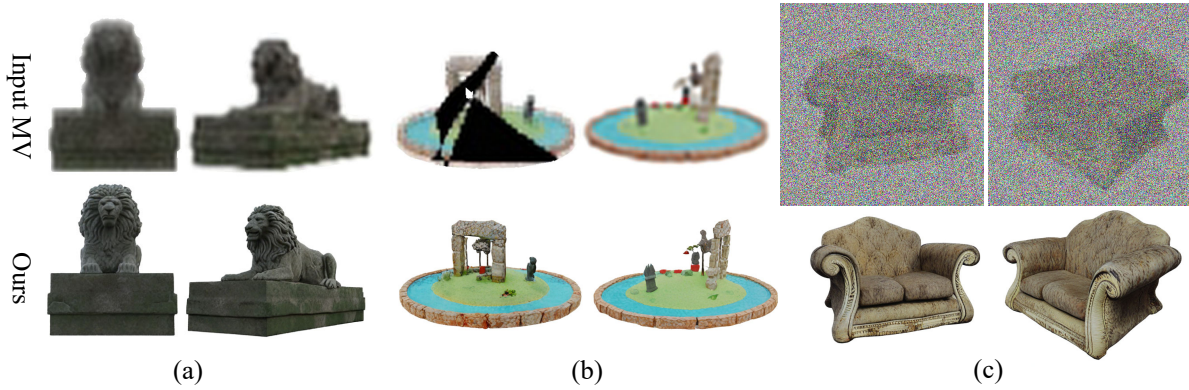


Figure B. Examples of handling extremely coarse inputs with 3DENHANCER.

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 [15] through modifying its multi-view input is shown in Fig. C.

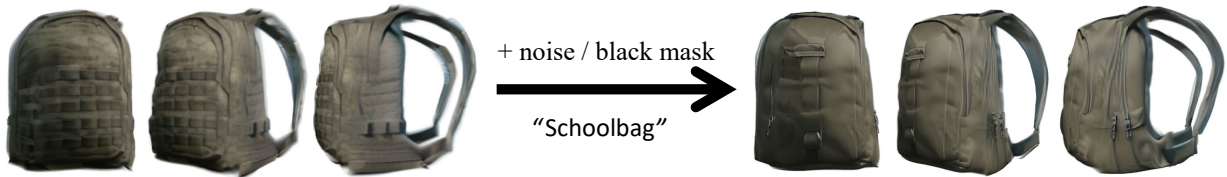


Figure C. Rendered views of edited 3D Gaussians using our multi-view editing approach. By adding a large noise or a black mask, and leveraging text prompts as guidance, we consistently modify the texture of the bags.

C.2. Color Correction

Previous studies [16, 24] 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 [16] can help resolve the global color scheme shift. As reported in Tab. E, applying wavelet color correction leads to improved fidelity metrics (higher PSNR, SSIM, and lower LPIPS [22]) 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. D.

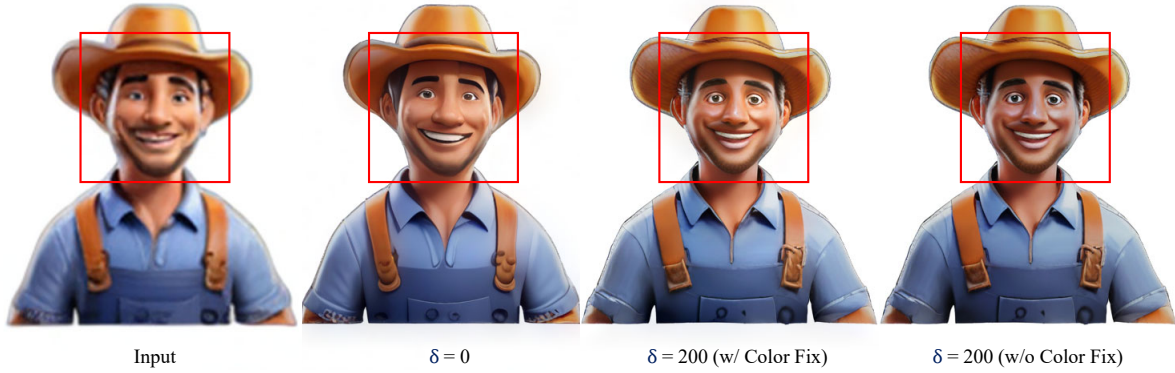


Figure D. Minor global color scheme shift at high noise levels. When the noise level δ is small, such as $\delta = 0$, our method maintains excellent color fidelity. However, at a higher noise level, such as $\delta = 200$ in the example, the output figure’s face appears slightly darker than that of the input. In this case, the wavelet color correction [16] could help mitigate this issue.

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 [18], StableSR [16], RealBasicVSR [1], and Upscale-a-Video [24]. 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 [1], and Upscale-a-Video [24]. Their task is to choose the visually superior enhanced results. A total of 20 participants take part in the study. As illustrated in Fig. E, 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.

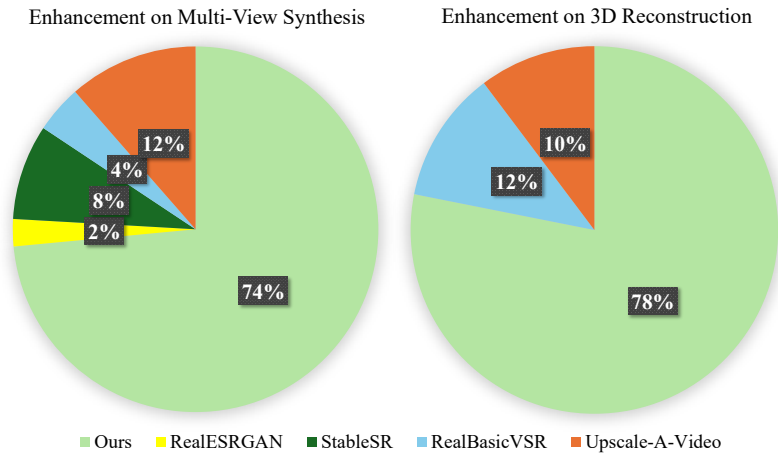


Figure E. User study results. Human voters consistently prefer our method over other approaches.

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 [6] 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 [13], 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 [3], our refinement process also minimizes a combined loss function, including a photometric reconstruction loss and a perceptual loss [22]. 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 [3]. 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 [1] and Upscale-A-Video [24]. Quantitative and qualitative results are presented in Tab. C and Fig. F, 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 C. Quantitative comparisons of optimizing low-resolution Gaussians. The best results are highlighted in **bold**.

Metrics	Low-Resolution Gaussians	RealBasicVSR [1]	Upscale-A-Video[24]	3DENHANCER
PSNR \uparrow	26.35	27.39	26.20	27.54
SSIM \uparrow	0.9120	0.9216	0.9184	0.9337
LPIPS \downarrow	0.1135	0.0803	0.0928	0.0756

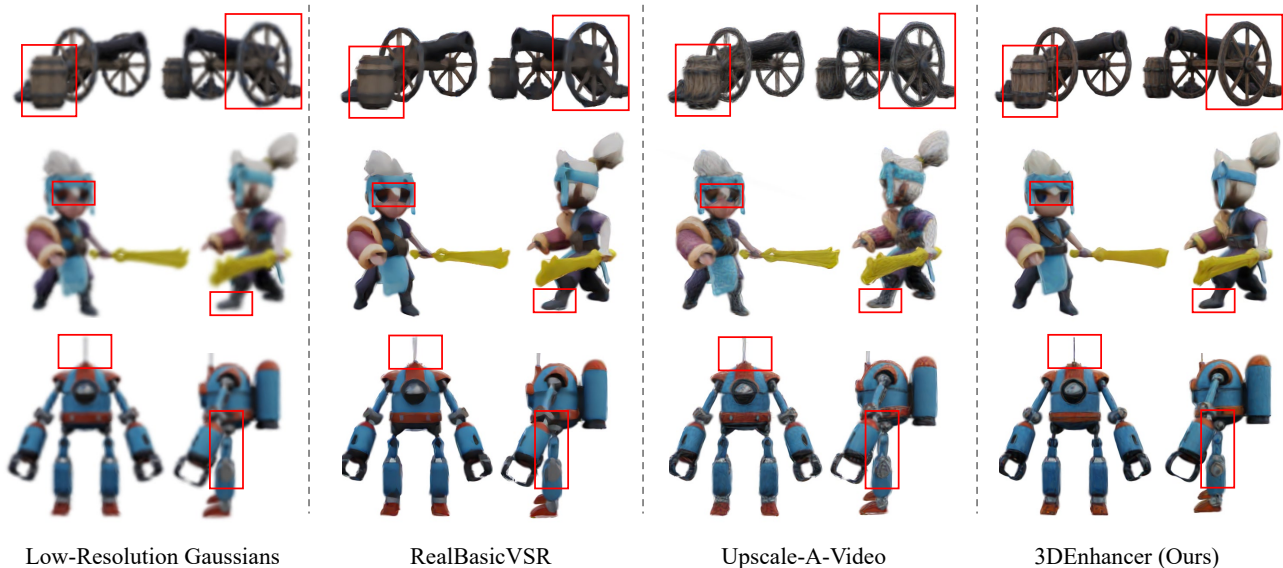


Figure F. Qualitative comparisons of optimizing low-resolution Gaussians. During optimization, both RealBasicVSR [1] and Upscale-A-Video [24] produce ghosting and blurry textures due to inconsistent outputs. Our 3DENHANCER achieves sharp and clear results.

D.3. Results of Generalization to Real-World Objects

We test our model on the constructed OmniObject3D dataset [20], which provides realistic 3D object scans, and also on complex, richly textured objects from Polycam. Backgrounds are removed as needed using BiRefNet [23]. As shown in Fig. G, our model effectively enhances real-world objects.



Figure G. Examples of handling complex real-world objects with 3DENHANCER. Our method generates rich textures on realistic objects.

D.4. Results of Further Fine-tuning Upscale-A-Video

Our work aims to provide a *generic* framework for 3D object enhancement, supporting enhancing (I) sparse multi-view images from large angles for multi-view reconstruction networks (*e.g.*, LGM [15]), and (II) coarse 3D model via per-instance optimization. Existing video diffusion models, *e.g.*, Upscale-A-Video [24], mainly rely on temporal attention for consistency. They are designed for handling adjacent video frames with minimal spatial variations, without considering camera pose. Thus, they struggle to establish multi-view correspondences in case (I), where input views vary significantly, leading to suboptimal results. Additionally, due to the huge GPU memory cost of video diffusion models, they also cannot handle dense 360° views simultaneously, typically working with short video sequences (*e.g.*, 8 frames for Upscale-A-Video), limiting their performance in case (II) as well. Thus, this study is crucial to explore new and effective modules of *sparse* multi-view attention for 3D enhancement, using a pose-aware encoder and an epipolar aggregation mechanism, which together achieve superior results in both (I) and (II) (see Tabs. 1-3 and Figs. 3 and 4). All baseline methods in the main paper are fine-tuned on the Objaverse dataset. We further fine-tune Upscale-A-Video with our proposed data augmentation. The results in the Tab. D and Fig. H show that our method still outperforms the video-based Upscale-A-Video, further supporting our discussion here.

Table D. Quantitative comparisons with fine-tuned Upscale-A-Video (UAV) on synthetic Objaverse multi-view images and Low-Resolution (LR) Gaussians.

Method	<i>Synthetic Objaverse</i>			<i>Low-Resolution Gaussians</i>		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
UAV (main paper)	25.57	0.8937	0.1153	26.20	0.9184	0.0928
UAV (further fine-tuned)	26.14(+0.57)	0.9086(+0.0149)	0.0996(-0.0157)	26.69 (+0.49)	0.9197 (+0.0013)	0.0850 (-0.0078)
Ours	27.53	0.9265	0.0626	27.54	0.9337	0.0756

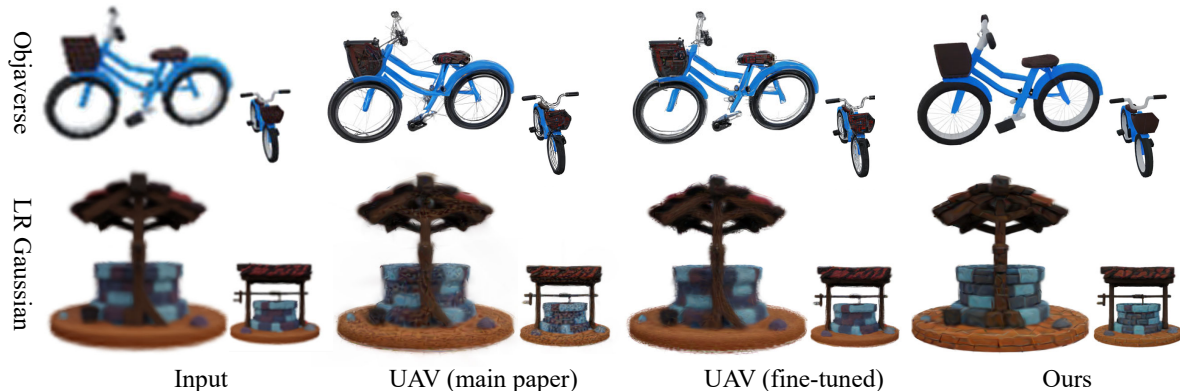


Figure H. Qualitative comparisons with fine-tuned Upscale-A-Video (UAV) on synthetic Objaverse multi-view images and Low-Resolution (LR) Gaussians. With additional fine-tuning using our augmentations, Upscale-A-Video reduces inconsistent artifacts outside the object area. Our method still shows superior generative capabilities.

D.5. More Comparisons

In this section, we introduce another baseline from the multi-view image upscale module in Unique3D [19]. This baseline fine-tunes ControlNet-Tile [21] 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. E. Additionally, we provide more visual comparisons of our method with all other baselines, including RealESRGAN [18], StableSR [16], Unique3D’s MV Upscale [19], RealBasicVSR [1], and Upscale-a-Video [24]. Fig. I and Fig. J showcase the visual comparisons of multi-view enhancement on synthetic and in-the-wild datasets, respectively.

Table E. 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 \uparrow	25.75	26.18	27.53	27.50
SSIM \uparrow	0.8989	0.9055	0.9265	0.9258
LPIPS \downarrow	0.1300	0.1257	0.0626	0.0631

D.6. Video Demo

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

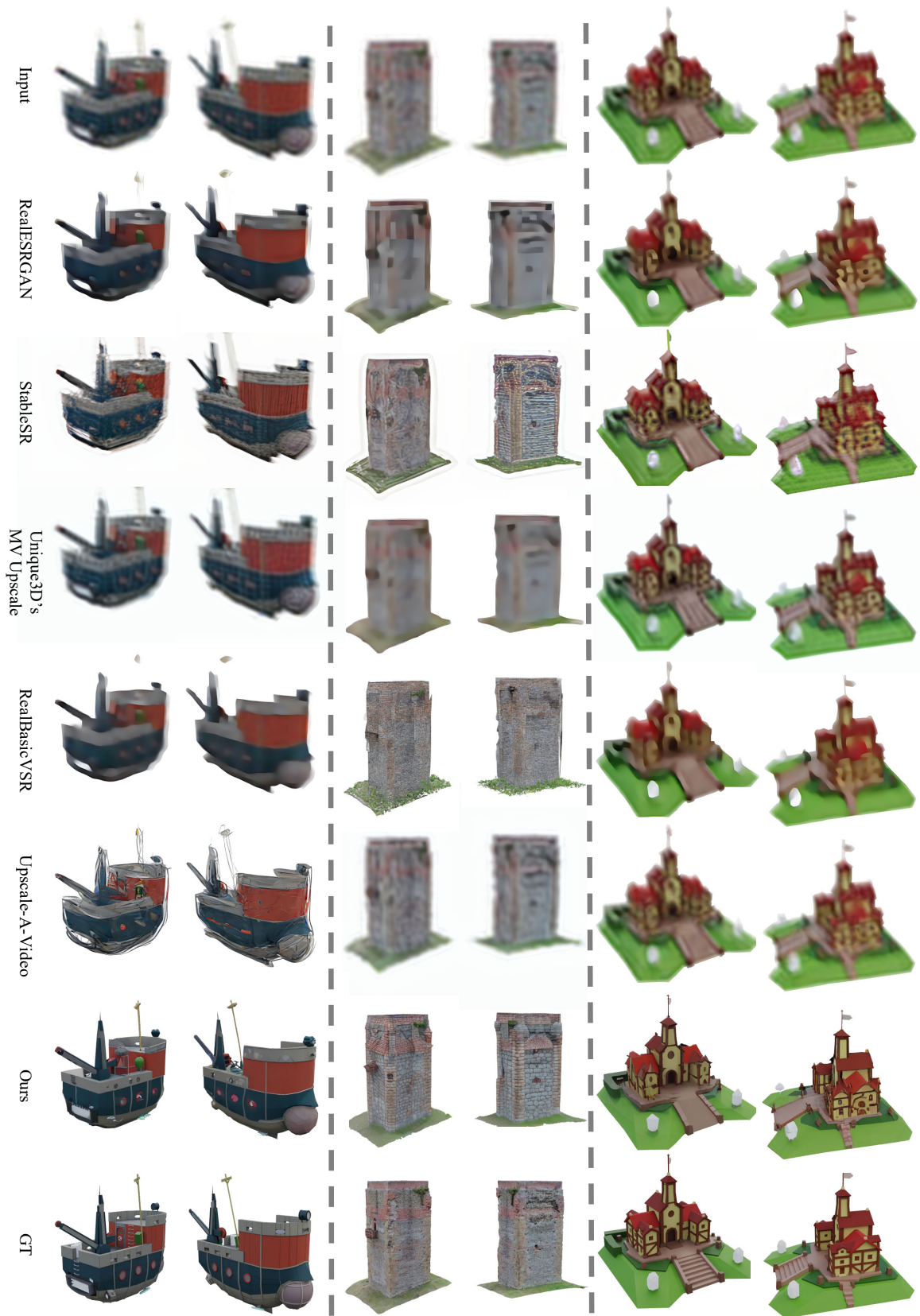


Figure I. Qualitative comparisons on the Objaverse synthetic dataset. Our 3DENHANCER demonstrates promising improvements, with increased detail and enhanced realism. (Zoom in for best view.)

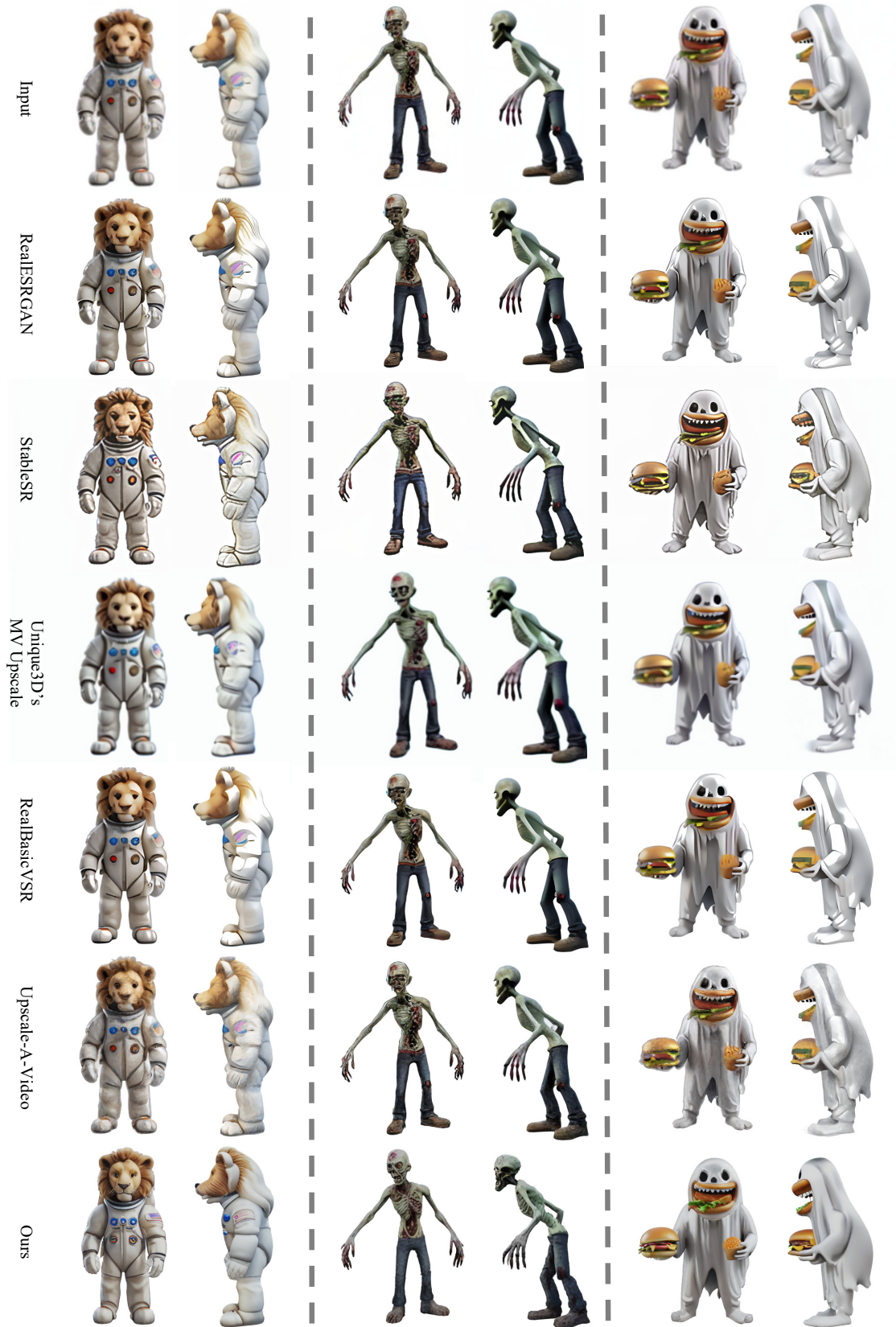


Figure J. Qualitative comparisons on the in-the-wild dataset. Our 3DENHANCER yields significant improvements, providing enhanced detail and consistent output. (**Zoom in for best view.**)

References

- [1] Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Investigating tradeoffs in real-world video super-resolution. In *CVPR*, 2022. 5, 6, 8
- [2] 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. 2
- [3] 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
- [4] Jonathan Ho. Classifier-free diffusion guidance. In *NeurIPS*, 2021. 3
- [5] 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. 3
- [6] 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. 6
- [7] 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. 3
- [8] 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. 3
- [9] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 2
- [10] 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. 2
- [11] 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. 3
- [12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2
- [13] 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. 6
- [14] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. MVDream: Multi-view diffusion for 3D generation. In *ICLR*, 2024. 2
- [15] 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. 4, 7
- [16] 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. 4, 5, 8
- [17] Peng Wang and Yichun Shi. ImageDream: Image-prompt multi-view diffusion for 3D generation. *arXiv preprint arXiv:2312.02201*, 2023. 3
- [18] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data. In *ICCVW*, 2021. 5, 8
- [19] 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. 8
- [20] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Liang Pan Jiawei Ren, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *CVPR*, 2023. 6
- [21] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 2, 8
- [22] 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. 4, 6
- [23] Peng Zheng, Dehong Gao, Deng-Ping Fan, Li Liu, Jorma Laaksonen, Wanli Ouyang, and Nicu Sebe. Bilateral reference for high-resolution dichotomous image segmentation. *CAAI Artificial Intelligence Research*, 3:9150038, 2024. 6
- [24] 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. 4, 5, 6, 7, 8