LatentEditor: Text Driven Local Editing of 3D Scenes

Umar Khalid*,[1], Hasan Iqbal*,[2], Nazmul Karim*,[1], Muhammad Tayyab[1], Jing Hua[2], and Chen Chen[1]

[1] Center for Research in Computer Vision, University of Central Florida, Orlando, FL, USA

[2] Department of Computer Science, Wayne State University, Detroit, MI, USA

Abstract. While neural fields have made significant strides in view synthesis and scene reconstruction, editing them poses a formidable challenge due to their implicit encoding of geometry and texture information from multi-view inputs. In this paper, we introduce LatentEditor, an innovative framework designed to empower users with the ability to perform precise and locally controlled editing of neural fields using text prompts. Leveraging denoising diffusion models, we successfully embed real-world scenes into the latent space, resulting in a faster and more adaptable NeRF backbone for editing compared to traditional methods. To enhance editing precision, we introduce a delta score to calculate the 2D mask in the latent space that serves as a guide for local modifications while preserving irrelevant regions. Our novel pixel-level scoring approach harnesses the power of InstructPix2Pix (IP2P) to discern the disparity between IP2P conditional and unconditional noise predictions in the latent space. The edited latents conditioned on the 2D masks are then iteratively updated in the training set to achieve 3D local editing. Our approach achieves faster editing speeds and superior output quality compared to existing 3D editing models, bridging the gap between textual instructions and high-quality 3D scene editing in latent space. We show the superiority of our approach on four benchmark 3D datasets, LLFF [26], IN2N [8], NeRFStudio [44] and NeRF-Art [47]. Project Page: https://latenteditor.github.io/

1 Introduction

The advent of neural radiance fields (NeRF) [27], neural implicit functions [49], and subsequent innovations [20,28,48] has revolutionized 3D scene reconstruction and novel view synthesis. These neural fields, leveraging multi-view images and volume/surface rendering mechanisms, have enabled neural networks to implicitly represent both geometry and texture of scenes, offering a more reliable and user-friendly alternative to the intricate matching and complex post-processing in traditional methods [19, 54].

* * Equal Contribution

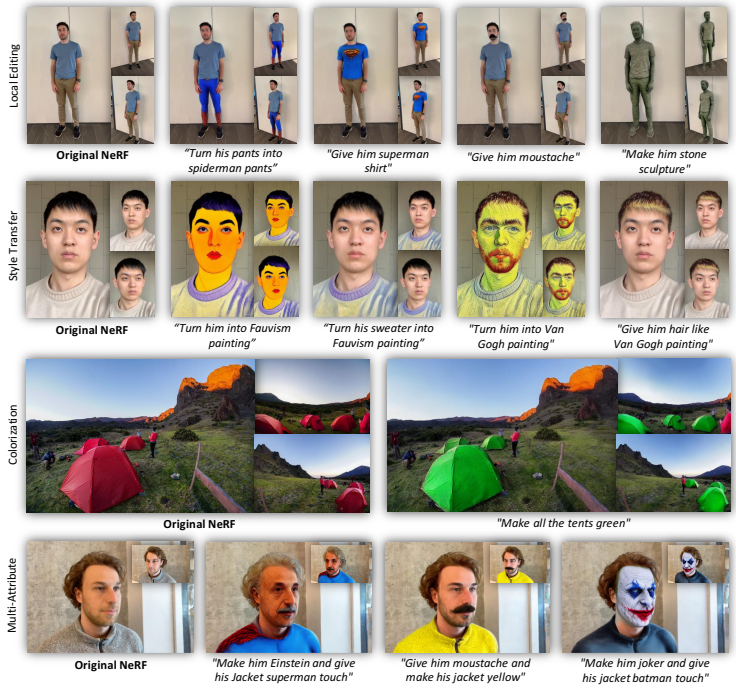

Fig. 1: Our proposed method ***LatentEditor*** enables text-based NeRF editing (e.g. color, attributes, style, etc.) by instilling both spatial and 3D awareness into image diffusion models. It can be observed in all the results that the background is intact with the color change or style transfer textual prompts.

While neural fields streamline the digital representation of real-world scenes, editing these fields poses significant challenges. Unlike conventional pixel-space edits [1, 2, 10, 29, 37], editing a NeRF-represented scene requires synchronized editing of multi-views under different perspectives to maintain their consistency, a complex and error-prone task due to the high-dimensional neural network features encoding of shape and texture information. Prior approaches have explored various editing aspects [3, 50, 51], yet they require extensive user interaction. More recent developments, like InstructNeRF2NeRF (IN2N) [8], employ text and image-conditioned models for 3D object creation and editing. However, IN2N struggles with localizing edits.

To tackle this issue, we propose **LatentEditor** that seeks to enhance local editing by predicting the editing area from the editing prompt inside the stable diffusion [38] denoising process. Our approach hinges on the critical insight that maintaining consistency in the diffusion feature space is viable and essential for achieving coherent edits in the 3D scene. To this end, we introduce a novel approach to assign delta scores to latent pixels in our designed delta module by contrasting noise predictions from provided instructions against unconditional

noise prediction to constrain the local editing. However, projecting these delta scores back to the RGB pixel space introduces inconsistencies. As there is a lack of direct alignment between latent and image pixels, a slight inconsistency in the latent space can affect the NeRF training significantly. Therefore, we propose to train NeRF directly within the latent space to enable local latent editing. Our approach, informed by the principles of shape-guided 3D generation in latent space [24], incorporates a refining module that consists of a residual adapter with self-attention to ensure consistency between rendered latent features and the scene's original latent. During inference, LatentEditor generates a latent representation for specific poses, convertible into RGB images through a Stable Diffusion [38] decoder. The efficacy of LatentEditor is evident in Figure 1, showcasing its capability for precise and minimal local edits in 3D NeRF scenes. The primary **contributions** of this paper are as follows:

- We introduce an efficient text-driven 3D NeRF local editing framework that operates solely based on text prompts, eliminating the need for additional controls. This innovation marks a significant advancement in text-driven 3D scene editing.
- Our unique delta module, utilizing the InstructPixtoPix [2] backbone, enables a novel mechanism for local editing. Guided solely by editing prompts, it efficiently calculates 2D masks in the latent space, automatically constraining targeted modifications in precise locations.
- In our efficient NeRF editing method, NeRF operates directly in its latent space, reducing computational costs significantly. In addition, our designed delta module further limits the required number of editing iterations, resulting in up to a 5-fold reduction in editing time compared to the baseline IN2N [8].
- A newly introduced refining module, featuring a trainable adapter with residual and self-attention mechanism, ensures enhanced consistency in the integration of latent masks within latent NeRF training. This adapter is key in aligning rendered latent features with the scene's original latents, resolving unwanted inconsistencies.

Extensive experiments on four 3D datasets and practical applications demonstrate our framework's capability to achieve spatially and semantically consistent performance and precise multi-attribute local editing in 3D scenes.

2 Related Work

Recent advancements in denoising diffusion probabilistic models [5,10,11,13,42] have enabled high-quality image generation from complex text cues [13,37–39]. These models have been refined for text-driven image editing, with significant contributions by [1,4,9,14]. Notably, SDEdit [23] and DiffEdit [4] have pioneered using denoising diffusion in image editing, despite limitations in preserving original image details or handling complex captions. These advancements paved the way for novel 3D scene synthesis, particularly with text-to-image models like CLIP [34] in DreamField [12] and DreamFusion [33].

Table 1: Comparative analysis of prior methods. Our approach (*LatentEditor*) stands out by not requiring any guidance and showcasing a broader capability for editing, particularly in achieving local edits without reliance on pre-trained segmentation models.

Methods	Guidance			Editing Capacity			
	Pre-Computed Masks	Bounding Box	GAN Guidance	Text Driven Style Transfer		Multi-Attribute Editing	Local Editing
Blend-NeRF [15]	✓	✗	✗	✗	✗	✗	✓
Blended-NeRF [6]	✗	✓	✗	✓	✗	✗	✓
DreamEditor [55]	✓	✗	✗	✓	✓	✗	✓
Control-4D [41]	✗	✗	✓	✓	✗	✗	✗
NeRF-Art [47]	✗	✗	✗	✓	✓	✗	✗
Instruct-N2N [8]	✗	✗	✗	✓	✓	✗	✗
LatentEditor (Ours)	✗	✗	✗	✓	✓	✓	✓

Starting with DreamFusion [33], subsequent methods [21,24,35] demonstrated noteworthy outcomes using diffusion-based priors. However, these methods are limited to generating novel 3D scenes, making them unsuitable for our focus on NeRF-editing, which modifies existing 3D scenes based on provided conditions.

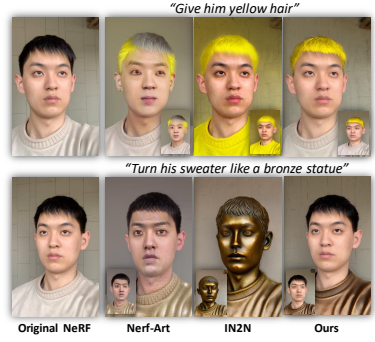

Fig. 2: Local Editing Challenge: Comparative analysis of local editing capabilities between our *LatentEditor* method and other text-driven 3D NeRF editing approaches, specifically IN2N [8] and NeRF-Art [47]. Our approach preserves the background seamlessly under both editing prompts.

Compared to novel object generation, NeRF editing is less explored due to its inherent complexity. Initial efforts focused on color, geometric, and style modifications [7, 17, 22, 46, 53]. The integration of text-to-image diffusion models is a recent trend in NeRF editing, with methodologies like Score Distillation Sampling in DreamFusion [33], and regularization approaches in Vox-e [40] and NeRF-Art [47]. Notably, Instruct-NeRF2NeRF (IN2N) [8] utilized 2D image translation models for NeRF editing based on text prompts but faced over-editing issues. DreamEditor [55] addressed this with a mesh-based approach for focused edits using pre-computed masks. Similarly, Blended-NeRF [6] and Blend-NeRF [15] incorporated additional cues like bounding boxes for localized adjustments. Our approach innovates by integrating latent space NeRF training with a unique delta module leveraging diffusion models to generate editing masks. This enables precise local editing without extra guidance. The comparative analysis (Table 1 and Figure 2) showcases our framework's superior local editing compared to existing text-driven NeRF methods.

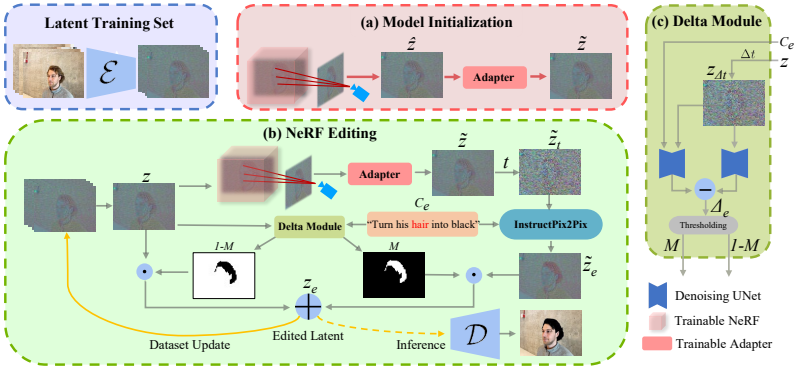

Fig. 3: Overall pipeline of LatentEditor. (a) We initialize the NeRF model within the latent domain, guided by the latent features of the original dataset. Our refinement adapter mitigates the misalignment in the latent space and encompasses a trainable adapter with residual and self-attention mechanisms. (b) Upon initialization, LatentEditor iteratively refines the model within the latent space for a predetermined number of iterations, while consistently updating the training set with the edited latents, Z_e. (c) The Delta Module is adept at interpreting prompts and produces the mask for targeted editing. An RGB image can be obtained by feeding the edited latent to the stable diffusion (SD) [38] decoder $\mathcal{D}$ whereas $\mathcal{E}$ represents SD [38] encoder.

3 Proposed Framework

Our objective is to introduce a text-prompt-driven approach that enables efficient and effective local editing of real-world 3D scenes. Our method, as illustrated in Figure 3, commences by optimizing a Neural Radiance Field (NeRF) within the latent space delineated by Stable Diffusion [38], thus anchoring the scene representation using latent feature vectors (see Figure 3(a)).

For scene editing, we present an innovative approach to facilitate localized editing using unconditional edits within the framework of the IP2P [2] model (see Figure 3(b)). Such unconditional edits help compute latent masks within our distinctive delta module as illustrated in Figure 3(c). Original latents are edited through the guidance of the masks and prompts.

3.1 Background

InstructPix2Pix. IP2P [2] edits an input image I based on a textual editing instruction C_e. Leveraging latent diffusion techniques [38] and a Variational Autoencoder (VAE) with an encoder $\mathcal{E}$ and decoder $\mathcal{D}$, IP2P processes a noisy image latent z_t. It predicts a less noisy version z_0 by approximating the noise vector $\hat{\varepsilon}$ using a U-Net ε_θ:

$$\hat{\varepsilon} = \varepsilon_\theta(z_t, I, C_e), \tag{1}$$

where $t \in T$ represents the noise level. IP2P is trained under various conditions, enabling both conditional and unconditional denoising.

Neural Radiance Fields NeRF [27] uses Multi-Layer Perceptrons (MLPs) to estimate the density σ and color $\mathbf{c}$ for a given 3D voxel $\mathbf{p} = (p_x, p_y, p_z)$ and view direction $\mathbf{v}$. After transforming $\mathbf{p}$ and $\mathbf{v}$ into high-frequency vectors via positional encoding $\phi(\cdot)$, NeRF produces

$$(\mathbf{c}, \sigma) = F_\theta^c(\phi(\mathbf{p}), \phi(\mathbf{v})). \tag{2}$$

NeRF calculates the world-space ray $\mathbf{r}(\tau) = \mathbf{c} + \tau\mathbf{v}$ per image pixel and minimizes the difference between rendered and captured pixel colors through the loss $\mathcal{L}(C(\mathbf{r}), \hat{C}(\mathbf{r}))$.

InstructNeRF2NeRF IN2N [8] fine-tunes reconstructed NeRF models using editing instructions to create modified scenes. It employs an iterative Dataset-Update (DU) strategy, where dataset images are successively replaced with post-editing versions. This process enables the gradual integration of diffusion priors into the 3D scene. IN2N leverages an image-conditioned diffusion model, IP2P [2], to facilitate these edits.

3.2 Method

We outline our latent training pipeline before delving into the details of the latent editing framework as an application stemming from latent training.

3.3 Latent Training.

Given a dataset of multi-view images $\mathbf{I} \in \mathbb{R}^{N\times W\times H\times 3}$, we encode these images using an encoder, $\mathcal{E}$, to obtain latent features $z^n = \mathcal{E}(I^n) \in \mathbb{R}^{W'\times H'\times 4}$ for $W' < W$ and $H' < H$. These latent feature maps $\mathbf{Z} := \{z^n\}_{n=1}^N$ serve as labels for initial LatentEditor NeRF training. We redefine the volume rendering integral as follows:

$$\hat{Z}(\mathbf{r}) = \int_{\tau_n}^{\tau_f} \mathbf{T}(\tau)\sigma(\mathbf{r}(\tau))\mathbf{z}(\mathbf{r}(\tau, \mathbf{d}))\, d\tau, \tag{3}$$

where $\mathbf{T}(\tau)$ is the accumulated transmittance, $\sigma(\mathbf{r}(\tau))$ is the density, and $\mathbf{z}(\mathbf{r}(\tau), \mathbf{d})$ is the radiance emitted at $\mathbf{r}(\tau)$.

The reconstruction loss, as the difference between estimated and actual pixel latent values, is defined by:

$$\mathcal{L}_r = \sum_{\mathbf{r}\in\mathcal{R}} \|\hat{Z}(\mathbf{r}) - Z(\mathbf{r})\|^2, \tag{4}$$

The NeRF model F_θ^z is trained to predict both latent features $\mathbf{z}$ and density σ from encoded positions and directions:

$$(\mathbf{z}, \sigma) = F_\theta^z(\phi(\mathbf{p}), \phi(\mathbf{v})). \tag{5}$$

Refinement Adapter with Self-Attention. Our refinement module, addressing misalignment in latent space, includes a trainable adapter for real-world 3D

scene editing. It performs the following residual and self-attention operations on an input tensor $\hat{z} \in \mathbb{R}^{4\times h'\times w'}$:

$$z_{\text{attention}} = \text{SelfAttention}(\text{ConvDown}(\hat{z})) \tag{6}$$

$$\tilde{z} = \hat{z} + \text{ConvUp}(z_{\text{attention}}) \tag{7}$$

The refinement loss for pixel latent vector $\tilde{Z}^i$ from refined feature map $\tilde{z}^i$ is:

$$\mathcal{L}_f = \sum_{\mathbf{r}\in\mathcal{R}} \left\| \tilde{Z}^i(\mathbf{r}) - Z^i(\mathbf{r}) \right\|^2, \tag{8}$$

where $\tilde{z}^i = F_\Theta(\hat{z}^i)$.

Camera Parameters Alignment in the Latent-space. Estimating Camera parameters from Structure-from-Motion (SfM) techniques is a standard preprocessing step for NeRF training for a scene. However, directly applying these parameters, estimated in pixel space, for NeRF training in the latent space, leads to subpar renderings marked by blurriness. While the refinement adapter mitigates this issue to a certain degree, achieving high-quality 3D reconstructions of objects and vast scenes still requires precise camera parameters for optimal outcomes. Scalar adjustments alone are insufficient for correcting camera parameter disparities which stem from the varied impact of camera parameters on image projection, influenced by the unit differences between focal length (in pixels) and translation (in world units) [31]. Facing these challenges, we propose a streamlined preconditioning strategy in the latent space for optimizing camera parameters inspired by [31]. This approach, grounded in analyzing the camera projection function's sensitivity to changes, employs a whitening transform computed via a proxy problem. This technique separates correlated parameters and balances their influence, simplifying the joint optimization process.

In our investigation of camera parameterizations, we underscore the critical impact of parameter choice on reconstruction quality. By considering camera projection as a function $\mathcal{P}^m(\phi) : \mathbb{R}^k \to \mathbb{R}^{2m}$ over m latnet points $\{Z_j\}_{j=1}^m$, we aim to discern the influence of camera parameters on the scene's reconstructed representation. To accurately capture the nuances of parameter effects, we examine the Jacobian matrix at a particular point ϕ^0 in the parameter space:

$$\left.\frac{\partial \mathcal{P}^m}{\partial \phi}\right|_{\phi=\phi^0} = J_{\mathcal{P}} \in \mathbb{R}^{2m\times k}, \tag{9}$$

where $\Sigma_{\mathcal{P}} = J_{\mathcal{P}}^\top J_{\mathcal{P}} \in \mathbb{R}^{k\times k}$ highlights the motion magnitude and correlation between camera parameters. We introduce a preconditioning matrix $\mathcal{M}$ to equalize the effect of different camera parameters, ensuring that the covariance matrix $\Sigma_{\mathcal{P}'} = J_{\mathcal{P}'}^\top J_{\mathcal{P}'}$ transforms to the identity matrix $\mathcal{I}_k$. To meet this condition, we employ Cholesky Decomposition [16]. Cholesky Decomposition provides a lower triangular matrix $\mathcal{L}$ such that: $\Sigma_{\mathcal{P}} = \mathcal{L}\mathcal{L}^\top$. Then, the preconditioning matrix is formulated as:

$$\mathcal{M}^{-1} = \mathcal{L}^{-1}, \tag{10}$$

where $\mathcal{L}^{-1}$ is the inverse of the lower triangular matrix obtained from Cholesky Decomposition of the covariance matrix $\Sigma_{\mathcal{P}}$. This approach aims to normalize the influence of camera parameters during optimization, improving the conditioning of the optimization problem. To implement camera parameterizations, we adopt residuals $\delta\phi_i$ applied to initial parameters ϕ_i^0, employing FocalPose's joint pose and focal length parameterization designed for object-centric views. This setup includes principal point and radial lens distortion parameters. Originating from FocalPose's [32] iterative pose estimation framework, we modify this parameterization to a residual form relative to initial estimates, enhancing adaptability and precision in parameter adjustments.

Regularizing Camera Parameters. To regularize camera parameters and prevent them from deviating significantly from their initial values during training, we introduce a regularization term to the loss function. This regularization term penalizes large deviations from the initial SfM parameter estimates, encouraging the optimization to favor solutions close to the initial values. The total loss function is defined as: The total training loss including refinement and reconstruction losses:

$$\mathcal{L}_T = \lambda_r \mathcal{L}_r + \lambda_f \mathcal{L}_f + \lambda_p \mathcal{L}_{\text{reg}}, \tag{11}$$

We concurrently update NeRF (F_θ) and the refinement adapter (F_Θ) by minimizing $\mathcal{L}_T$ for various view reconstructions while we set $\lambda_p = 0$ during editing where we don't optimize the camera parameters. The regularization term can be expressed using the squared Euclidean distance (L2 norm) as follows:

$$\mathcal{L}_{\text{reg}} = \sum_i \|\theta_i - \theta_{i,0}\|^2, \tag{12}$$

with θ_i denoting the current value of the camera parameter, $\theta_{i,0}$ its initial value, and the summation extending over all camera parameters. The regularization weight λ controls the strength of the regularization, balancing the fidelity to the data against the preservation of the initial camera estimates.

3.4 Latent Editing: An Application of Latent-NeRF

After initializing the NeRF model in the latent domain with features $\mathbf{Z} = \{z^n\}_{n=1}^N$, we employ the InstructPix2Pix (IP2P) framework [2] to align its parameters with the textual cue C_e. The original latent variables $\mathbf{Z}$ are systematically replaced with edited versions $\mathbf{Z_e} = \{z_e^n\}_{n=1}^N$ at an editing rate υ, enabling progressive transformation to reflect desired edits. For viewpoint K and editing iteration s, we obtain a render, $\tilde{z}^n$ from F_θ^z and generate edited latents z_e^n using our novel local editing technique.

Prompt Aware Pixel Scoring. We design a delta module by modulating the IP2P [2] diffusion process that aims to guide the generation of z_e^n using a generated mask. Starting with noise addition to latent z^n up to timestep Δt, we obtain the noisy latent $z_{\Delta t}^n$ as:

$$z_{\Delta t}^n = \sqrt{\beta_{\Delta t}} z^n + \sqrt{1 - \beta_{\Delta t}}\varepsilon, \tag{13}$$

where $\varepsilon \sim \mathcal{N}(0,1)$, and β_t is the noise scheduling factor at timestep t.

IP2P's score estimation encompasses conditional and unconditional editing:

$$\begin{aligned}\tilde{\varepsilon}_\theta(z_t, I, C_e) = \varepsilon_\theta(z_t, \varnothing_I, \varnothing_e) \\ + s_I\big(\varepsilon_\theta(z_t, I, \varnothing_e) - \varepsilon_\theta(z_t, \varnothing_I, \varnothing_e)\big) \\ + s_T\big(\varepsilon_\theta(z_t, I, C_e) - \varepsilon_\theta(z_t, I, \varnothing_e)\big).\end{aligned} \tag{14}$$

We calculate the delta scores, Δ_ε, using two noise predictions:

$$\Delta_\varepsilon = |\varepsilon_\theta(z^n_{\Delta t}, I, C_e) - \varepsilon_\theta(z^n_{\Delta t}, I, \varnothing_e)| \tag{15}$$

where $z^n_{\Delta t}$ is calculated using Equation 13, and Δt is a hyperparameter in our method.

The higher values of the delta scores, $\Delta_\varepsilon \in \mathbb{R}^{W' \times H' \times 4}$ indicate the region to be edited. Hence, a binary mask, $M \in \mathbb{R}^{W' \times H' \times 4}$ can be generated by applying a threshold μ on Δ_ε as following,

$$M = \begin{cases} 1 & \text{if } \Delta_\varepsilon \geq \mu \\ 0 & \text{otherwise} \end{cases}$$

Given that M calculated above is not unifrom across different view, we adopt a zero-shot point tracker [36] to achieve consistent mask creation across all views. The process initiates by selecting query points within the first video frame's ground truth mask. These query points are determined through K-Medoids [30] sampling, which selects cluster centers as query points from the K-Medoids clustering. This strategy ensures extensive representation of different object parts and increases robustness against noise and anomalies. Since M is generated in the latent space, we will perform the local editing by refining the NeRF on edited latents as discussed in the following section.

Local Editing with Latent Features. Once the delta module outputs the mask M, a noisy version $\tilde{z}^n_t$ of the current render is generated, where t now has a random value in a specific range $[t_{min}, t_{max}]$. The edited latent $\tilde{z}^n_{t-1,e}$ at time-step $t-1$, is computed using DDIM [42]:

$$\tilde{z}^n_{t-1,e} = \sqrt{\beta_{t-1}}\left(\frac{\tilde{z}^n_t - \sqrt{1-\beta_t}\tilde{\varepsilon}_t}{\sqrt{\beta_t}}\right) + \sqrt{1-\beta_{t-1}}\tilde{\varepsilon}_t. \tag{16}$$

where $\tilde{\varepsilon}_t = \tilde{\varepsilon}_\theta(\tilde{z}_t, I, C_e)$ is the predicted noise. Iteratively applying these denoising stages yields the edited latent z^n_e.

The final prediction $\tilde{z^n}_e$ after complete denoising merges with z^n using the mask M:

$$z^n_e = \tilde{z}^n_e \odot M + (1-M) \odot z^n \tag{17}$$

Ultimately, for each n in the set $\{1, \ldots, N\}$, the locally edited latent z^n_e is substituted for the original z^n in an iterative fashion. This method ensures that pixels

outside the edit mask M remain unaltered, confining edits to the intended regions.

Optimizing NeRF Editing. We don't incorporate any additional preservation or density blending losses within our methodology. The loss defined in Equation 11 suffices for NeRF editing with DU, where the ground truth is continuously updated with edited latents. Our approach avoids the extensive hyperparameter tuning required by other NeRF editing methods [6, 15, 25].

Multi-attribute Editing. We also expand our method to tackle a more challenging multi-attribute editing task in NeRF. Our approach can be seamlessly integrated with any pre-trained grounding framework, such as GLIP [18]. However, given that the grounding mechanism is inherently embedded within our delta module, we performed prompt engineering for LLM [45].

For instance, employing our designed prompt, LLM [45] processes the input "Make his hair red and give him a blue jacket," yielding the output {["Make his hair red", "Give him a blue jacket"], 2}. Our delta module is designed to be aware of LLM output, generating M^1 and M^2 based on C_e^1 and C_e^2, respectively which further controls multi-edit in a single scene.

To achieve multi-attribute editing, we leverage pre-trained large language model (LLM) Llama 2 [45]. To achieve our multi-attribute editing, we feed in the given prompt C_e to the LLM, and the LLM output is engineered in a way that our proposed delta-module can generate multiple masks.

Here is the designed prompt used in the study:

```
### Contexts
Break the following editing prompt into multiple parts with "
    and" as the key indicator of partition. Produce editing
    prompts based on the given input.
### Input
Make his hair red and give him blue jacket.
### Response
```

4 Experimental Analysis

Implementation Details. In our experiments, we adopt the implementation strategy of IN2N [8], specifically setting the interval $[t_{\min}, t_{\max}] = [0.02, 0.98]$ and defining $\Delta t = 0.75$. Model initialization on the original scene is performed for 30,000 iterations, ensuring a robust baseline representation. The editing process then commences, with the number of iterations tailored to the number of training views ranging from 2,000 iterations to 4,000. Detailed descriptions of these settings are available in the *supplementary material*.

Baselines. For a comprehensive evaluation of LatentEditor's performance, we compared it against state-of-the-art (SOTA) models across four datasets: (i) IN2N [8], (ii) NeRF-Art [47], (iii) LLFF [26] and (iv) NeRFstudio Dataset [44]. We included various NeRF editing frameworks in our analysis, such as IN2N [8], NeRF-Art [47], Control-4D [41], and DreamEditor [55]. However, due to space

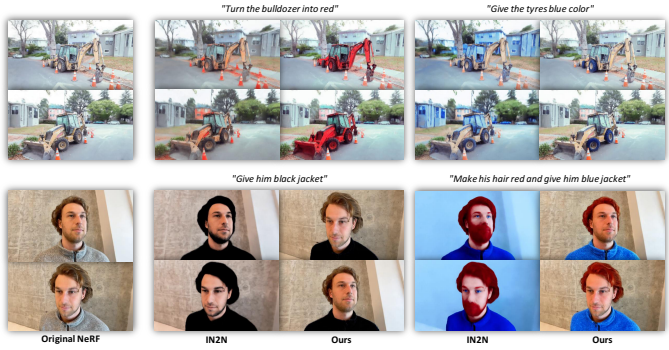

Fig. 4: Qualitative Results. The visual results of our approach, when contrasted with the baseline IN2N [8] across two distinct scenes, distinctly demonstrate that *LatentEditor* excels in accurately pinpointing the pertinent region, executing faithful text edits, and averting undesired alterations. These achievements prove challenging for the baseline method [8] to replicate effectively as IN2N [8] also changes the background objects' color to blue given that the editing prompt, "Make his hair red and give him blue jacket" only wants the jacket color to be changed.

constraints, we primarily emphasize qualitative comparisons with IN2N [8], the current benchmark in text-driven NeRF editing.

4.1 Results

Qualitative. Our method's unified editing capability is distinctly showcased in Figure 1.Furthermore, Figure 2 demonstrates LatentEditor's enhanced local editing prowess when juxtaposed with SOTA IN2N [8] approach. A notable distinction is observed in Figure 4, where our method adeptly adheres to the prompt *"Give him Black Jacket"*, unlike IN2N [8] which erroneously alters the hair color. Although style transfer results from both LatentEditor and IN2N [8] are comparable, as seen in Figure 5, due to their shared IP2P [2] backbone, LatentEditor exhibits more refined control.

The versatility of LatentEditor in managing complex prompts and multi-attribute edits is further highlighted in Figure 4, underscoring its robustness in diverse editing scenarios. We also assess our technique in **object removal**, aiming to seamlessly erase objects from 3D scenes, which might result in voids due to missing data. This process involves detecting and excising areas using the 2D mask, and then repairing these "invisible regions" in the original 2D frames with LAMA [43]. As shown in

Table 2: Quantitative evaluation of scene edits in terms of text alignment and frame consistency in CLIP space.

Method	CLIP Text-Image Direction Similarity↑	CLIP Direction Consistency↑	Edit PSNR↑
NeRF-Art [47]	0.2617	0.9188	21.04
Control4D [41]	0.2378	0.9263	19.85
DreamEditor [55]	0.2474	0.9312	20.67
IN2N [8]	0.2649	0.9358	24.07
Ours	**0.2661**	**0.9387**	**25.15**

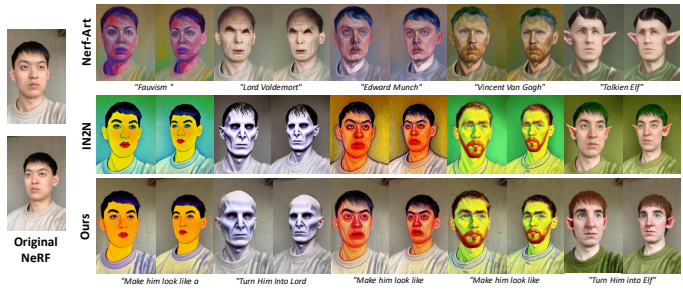

Fig. 5: Style Transfer Comparison. We present a visual representation for stylization editing, comparing our results with those obtained using NeRF-Art [47] and IN2N [8]. It can be observed that LatentEditor keeps the background intact while transferring the style of an object.

Fig. 6: Object Removal. Our method outperforms Gaussian Grouping [52] in removing 3D objects across various scenes.

Figure 6, LatentEditor excels in removing objects, delivering superior spatial accuracy and view consistency compared to Gaussian Grouping [52].

Quantitative. Quantitative evaluations, as detailed in Table 2, 200 edits across 20 scenes from the above-mentioned four datasets. Our method outperformed baselines in CLIP similarity scores and CLIP direction consistency, as averaged over multiple views rendered from NeRF.

User Study. To evaluate the subjective nature of scene editing, we conducted a user study comparing our method with SOTA alternatives. The study garnered a total of 500 votes across three key metrics: 3D consistency, preservation of original scene content, and adherence to text descriptions. As depicted in Figure 7, our method has been predominantly favored across these metrics. Further details on the quantitative evaluation

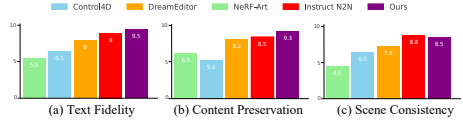

Fig. 7: User Study. LatentEditor achieves the highest text fidelity and content preservation scores.

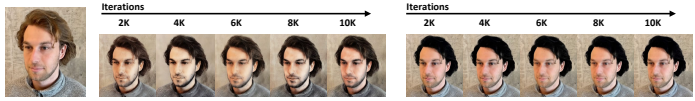

Fig. 8: Comparing *LatentEditor* to IN2N in terms of a reduced computational cost. Our approach achieves the desired editing results, "Turn his hair black", in approximately 2000 iterations, whereas IN2N continues to face challenges even after 10,000 iterations.

criteria and implementation of this user study are available in the *supplementary material*.

5 Ablations

Editing Rate vs Training Iterations. In this ablation study, we evaluate the computational efficiency of our LatentEditor approach against IN2N [8], particularly in terms of training iterations required to achieve a targeted editing performance. In Figure 8, using the editing prompt *"Turn his hair black"*, we demonstrate that LatentEditor significantly outperforms IN2N [8] in computational cost.

Specifically, our method achieves the desired editing results with an approximately five to ten-fold reduction in training iterations. LatentEditor requires only about 2000 iterations to reach the editing rate benchmark, while IN2N [8] still faces challenges even after 10,000 iterations. Despite LatentEditor involves multiple denoising stages per editing step, which increases the cost per step, it achieves the desired performance with significantly fewer iterations. As a result, it is **5-8** times faster in NeRF editing compared to IN2N [8].

Refinement Coefficient Sensitivity. To understand the impact of the refinement module, we conduct an ablation study on the refinement coefficient where this coefficient is set to zero. The results, as illustrated in Figure 9, reveal that omitting the refinement leads to noticeable inconsistencies and artifacts in the NeRF scene. These findings highlight the importance of the refinement loss in achieving high-quality, consistent NeRF scenes and validate its inclusion in our loss formulation.

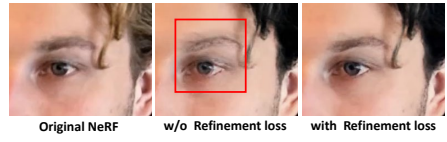

Fig. 9: Qualitative comparison of an edited image with and without refinement loss against the editing prompt, *"turn his hair black"*. The red box indicates the noticeable artifacts without the refinement module.

6 Limitations

Our method's efficacy is contingent on the capabilities of the pre-trained IP2P model [2], which presents certain limitations. This is particularly evident in

cases where IP2P's inherent weaknesses are pronounced. For instance, in Figure 10, the prompt *"Turn the bear into an orange bear"* exemplifies such a scenario. While IN2N [8] introduces random coloring throughout the scene, failing to generate the desired NeRF, our method, though demonstrating more controlled editing, does not completely succeed in turning the bear orange. The underlying limitation stems from IP2P's challenges in accurately interpreting and executing specific editing instructions like this. Our approach, being model-agnostic, can benefit from future enhancements in instruction-conditioned diffusion models, potentially overcoming these current constraints in localized edits.

7 Conclusion

Fig. 10: Our method grapples with instructions like "Turn the bear into an orange bear" due to IP2P's limitations, while LatentEditor's model-agnostic approach offers promise for addressing such issues through enhanced instruction-conditioned diffusion models.

In conclusion, LatentEditor marks a significant advancement in the field of neural field editing. We tackled the inherent challenges in editing neural fields, which arise from their implicit encoding of geometry and texture, by introducing a novel framework capable of precise, controlled editing via text prompts. By embedding real-world scenes into latent space using denoising diffusion models, our framework offers a faster and more adaptable NeRF backbone for text-driven editing. The introduction of the delta score, which calculates 2D masks in latent space for precise editing while preserving untargeted areas, is a key innovation. LatentEditor not only simplifies the process of 3D scene editing with textual instructions but also enhances the quality of the results, marking a new direction in 3D content creation and modification.

8 Acknowledgement

This work was partially supported by the NSF under Grant Numbers OAC-1910469 and OAC-2311245.

References

1. Avrahami, O., Lischinski, D., Fried, O.: Blended diffusion for text-driven editing of natural images. In: CVPR 2022. pp. 18208–18218 (2022)
2. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18392–18402 (2023)
3. Chen, J., Zhang, Y., Kang, D., Zhe, X., Bao, L., Jia, X., Lu, H.: Animatable neural radiance fields from monocular rgb videos. arXiv preprint arXiv:2106.13629 (2021)
4. Couairon, G., Verbeek, J., Schwenk, H., Cord, M.: Diffedit: Diffusion-based semantic image editing with mask guidance. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=3lge0p5o-M-
5. Frakes, E., Khalid, U., Chen, C.: Efficient and consistent zero-shot video generation with diffusion models. In: Kehtarnavaz, N., Shirvaikar, M.V. (eds.) Real-Time Image Processing and Deep Learning 2024. vol. 13034, p. 1303407. International Society for Optics and Photonics, SPIE (2024). https://doi.org/10.1117/12.3013575, https://doi.org/10.1117/12.3013575
6. Gordon, O., Avrahami, O., Lischinski, D.: Blended-nerf: Zero-shot object generation and blending in existing neural radiance fields. arXiv preprint arXiv:2306.12760 (2023)
7. Gu, J., Liu, L., Wang, P., Theobalt, C.: Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. arXiv preprint arXiv:2110.08985 (2021)
8. Haque, A., Tancik, M., Efros, A.A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19740–19750 (2023)
9. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022)
10. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems **33**, 6840–6851 (2020)
11. Iqbal, H., Khalid, U., Chen, C., Hua, J.: Unsupervised anomaly detection in medical images using masked diffusion model. In: International Workshop on Machine Learning in Medical Imaging. pp. 372–381. Springer (2023)
12. Jain, A., Mildenhall, B., Barron, J.T., Abbeel, P., Poole, B.: Zero-shot text-guided object generation with dream fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 867–876 (2022)
13. Karim, N., Khalid, U., Iqbal, H., Hua, J., Chen, C.: Free-editor: Zero-shot text-driven 3d scene editing. arXiv preprint arXiv:2312.13663 (2023)
14. Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6007–6017 (2023)
15. Kim, H., Lee, G., Choi, Y., Kim, J.H., Zhu, J.Y.: 3d-aware blending with generative nerfs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22906–22918 (2023)
16. Krishnamoorthy, A., Menon, D.: Matrix inversion using cholesky decomposition. In: 2013 signal processing: Algorithms, architectures, arrangements, and applications (SPA). pp. 70–72. IEEE (2013)

17. Kuang, Z., Luan, F., Bi, S., Shu, Z., Wetzstein, G., Sunkavalli, K.: Palettenerf: Palette-based appearance editing of neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20691–20700 (2023)
18. Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.N., et al.: Grounded language-image pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10965–10975 (2022)
19. Liu, H.K., Shen, I., Chen, B.Y., et al.: Nerf-in: Free-form nerf inpainting with rgb-d priors. arXiv preprint arXiv:2206.04901 (2022)
20. Liu, L., Gu, J., Zaw Lin, K., et al.: Neural sparse voxel fields. NeurIPS 2020 **33**, 15651–15663 (2020)
21. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9298–9309 (2023)
22. Liu, S., Zhang, X., Zhang, Z., Zhang, R., Zhu, J.Y., Russell, B.: Editing conditional radiance fields. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5773–5783 (2021)
23. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Guided image synthesis and editing with stochastic differential equations. In: International Conference on Learning Representations (2021)
24. Metzer, G., Richardson, E., Patashnik, O., Giryes, R., Cohen-Or, D.: Latent-nerf for shape-guided generation of 3d shapes and textures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12663–12673 (2023)
25. Mikaeili, A., Perel, O., Safaee, M., Cohen-Or, D., Mahdavi-Amiri, A.: Sked: Sketch-guided text-based 3d editing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14607–14619 (2023)
26. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (TOG) **38**(4), 1–14 (2019)
27. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
28. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) **41**(4), 1–15 (2022)
29. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021)
30. Park, H.S., Jun, C.H.: A simple and fast algorithm for k-medoids clustering. Expert systems with applications **36**(2), 3336–3341 (2009)
31. Park, K., Henzler, P., Mildenhall, B., Barron, J.T., Martin-Brualla, R.: Camp: Camera preconditioning for neural radiance fields. ACM Transactions on Graphics (TOG) **42**(6), 1–11 (2023)
32. Ponimatkin, G., Labbé, Y., Russell, B., Aubry, M., Sivic, J.: Focal length and object pose estimation via render and compare. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3825–3834 (2022)

33. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=FjNys5c7VyY
34. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
35. Raj, A., Kaza, S., Poole, B., Niemeyer, M., Ruiz, N., Mildenhall, B., Zada, S., Aberman, K., Rubinstein, M., Barron, J., et al.: Dreambooth3d: Subject-driven text-to-3d generation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2349–2359 (2023)
36. Rajič, F., Ke, L., Tai, Y.W., Tang, C.K., Danelljan, M., Yu, F.: Segment anything meets point tracking. arXiv preprint arXiv:2307.01197 (2023)
37. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022)
38. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR 2022. pp. 10684–10695 (2022)
39. Saharia, C., Chan, W., Saxena, S.e.a.: Photorealistic text-to-image diffusion models with deep language understanding. NeurIPS 2022 **35**, 36479–36494 (2022)
40. Sella, E., Fiebelman, G., Hedman, P., Averbuch-Elor, H.: Vox-e: Text-guided voxel editing of 3d objects. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 430–440 (2023)
41. Shao, R., Sun, J., Peng, C., Zheng, Z., Zhou, B., Zhang, H., Liu, Y.: Control4d: Efficient 4d portrait editing with text. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4556–4567 (2024)
42. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
43. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 2149–2159 (2022)
44. Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A., et al.: Nerfstudio: A modular framework for neural radiance field development. In: ACM SIGGRAPH 2023 Conference Proceedings. pp. 1–12 (2023)
45. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
46. Wang, C., Chai, M., He, M., et al.: Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In: CVPR 2022. pp. 3835–3844 (2022)
47. Wang, C., Jiang, R., Chai, M., He, M., Chen, D., Liao, J.: Nerf-art: Text-driven neural radiance fields stylization. IEEE Transactions on Visualization and Computer Graphics (2023)
48. Wang, C., Wu, X., Guo, Y.C., et al.: Nerf-sr: High quality neural radiance fields using supersampling. In: ACM MM 2022. pp. 6445–6454 (2022)
49. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. arXiv preprint arXiv:2106.10689 (2021)

50. Xiang, F., Xu, Z., Hasan, M., et al.: Neutex: Neural texture mapping for volumetric neural rendering. In: CVPR 2021. pp. 7119–7128 (2021)
51. Yang, B., Bao, C., Zeng, J.e.a.: Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. In: ECCV 2022. pp. 597–614. Springer (2022)
52. Ye, M., Danelljan, M., Yu, F., Ke, L.: Gaussian grouping: Segment and edit anything in 3d scenes. arXiv preprint arXiv:2312.00732 (2023)
53. Zhang, K., Kolkin, N., Bi, S., Luan, F., Xu, Z., Shechtman, E., Snavely, N.: Arf: Artistic radiance fields. In: European Conference on Computer Vision. pp. 717–733. Springer (2022)
54. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: Nerf++: Analyzing and improving neural radiance fields. arXiv preprint arXiv:2010.07492 (2020)
55. Zhuang, J., Wang, C., Lin, L., Liu, L., Li, G.: Dreameditor: Text-driven 3d scene editing with neural fields. In: SIGGRAPH Asia 2023 Conference Papers. pp. 1–10 (2023)