FocalDreamer: Text-Driven 3D Editing via Focal-Fusion Assembly

Yuhan Li, Yishun Dou, Yue Shi, Yu Lei, Xuanhong Chen, Yi Zhang, Peng Zhou, Bingbing Ni

1Shanghai Jiao Tong University, Shanghai 200240, China
2Huawei
{melodious, nibingbing}@sjtu.edu.cn

Abstract

While text-3D editing has made significant strides in leveraging score distillation sampling, emerging approaches still fall short in delivering separable, precise and consistent outcomes that are vital to content creation. In response, we introduce FocalDreamer, a framework that merges base shape with editable parts according to text prompts for fine-grained editing within desired regions. Specifically, equipped with geometry union and dual-path rendering, FocalDreamer assembles independent 3D parts into a complete object, tailored for convenient instance reuse and part-wise control. We propose geometric focal loss and style consistency regularization, which encourage focal fusion and congruent overall appearance. Furthermore, FocalDreamer generates high-fidelity geometry and PBR textures which are compatible with widely-used graphics engines. Extensive experiments have highlighted the superior editing capabilities of FocalDreamer in both quantitative and qualitative evaluations.

1 Introduction

Art reflects the figments of human imagination and creativity. Recently, the rapid development of neural generative models (Dhariwal and Nichol 2021) has significantly lowered the barriers for humans to engage in artistic creation with just a few words. However, these black-box models also deprive humans of a significant portion of control, which means the generation isn’t often aligned with expectations. In this work, we take a step towards precise editing for 3D creation, enabling networks to naturally expand user’s intentions, rather than controlling the entire generative process.

In the realms of animation, gaming, and the recent advance of virtual augmented reality, 3D models and scenes are commonly constructed as an assembly of semantically distinct base parts, which support the practice of rendering multiple copies of the same part across scenes with different transform matrices, called geometry instancing or instance reuse (Fig. 1). We believe that an ideal 3D editing workflow should possess the following good properties:

- **Separable.** Given a base shape, it should produce structurally separate parts (Li, Niu, and Xu 2020) facilitating for instance reuse and part-wise post-processing, grounded in widespread understanding.
- **Precise.** It should provide fine-grained and local editing, enabling precise control in the desired area (Zhuang et al. 2023), while maintaining other regions untouched.
- **Consistent.** After the editing process, the resultant shape should respect the characteristics of the source shape in harmonious appearance (Xie et al. 2023), while visually adhering to the text specifications.

Emerging approaches in text-3D editing have achieved noteworthy development, yet they often fall short in deliv-

Figure 1: Given the prompt “a butterfly over a tree stump”, our method delivers high-fidelity geometry and photorealistic appearance using PBR materials. Lines (b-c) showcase FocalDreamer’s capability for separable and precise edits.

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Corresponding author: Bingbing Ni.
We introduce the following key contributions to meet our outlined criteria: (1) **Separable**: we propose FocalDreamer, a user-friendly framework that permits intuitive object modifications using text prompts and a rough focal region for the intended edits. Instead of direct modifications to the base shape (e.g., the horse in Fig. 3), a novel editable part (wings in Fig. 3) is generated in the focal region, facilitating instance reuse and precise control. Equipped with geometry union and dual-path rendering, this part is merged with base mesh into a semantically unified shape in a lossless and differentiable manner, then optimized using a powerful text-to-image model to align the prompts and shapes. Furthermore, our decoupled learning of geometry and appearance yields detailed geometry and PBR textures, ensuring compatibility with prominent graphics engines. (2) **Precise**: Users delineate one or several ellipsoid focal regions, in which a spherical editable part initializes, acting as a smooth prior for the geometry network. The geometric focal loss is also introduced, discouraging edits beyond specified regions. (3) **Consistent**: a smooth, coherent surface is essential in certain scenes. Hence, a soft geometry union operator and a style consistency regularization are proposed to ensure a seamless geometric transition and stylistically consistent texture between the learnable part and base shape.

To our knowledge, this is the first component-based editing method with separate learnable parts. Rich experiments and detailed ablation studies highlight the superior editing capabilities of our approach, as shown in Fig. 2.

### 2 Related Work

**Text-guided Image Generation and Editing.** Significant progress in Text-to-Image (T2I) generation with diffusion models (Ho, Jain, and Abbeel 2020) is witnessed in recent years. More recently, with the availability of scalable generator architectures and extremely large-scale image-text paired datasets, they’ve demonstrated impressive performance in high-fidelity and flexible image synthesis (Rombach et al. 2022). Due to their comprehension of complex concepts, diffusion models are also amicable for various editing tasks, such as image inpainting (Lugmayr et al. 2022), image stylization (Zhang et al. 2023). The most relevant field to us among those is inpainting, which provides flexible control of the inpainted content, and a mask to constrain the shape of the inpainted object. SmartBrush (Xie et al. 2023) introduces a precision factor into the masks for multiple-grained controls on inpainting regions.

**Text-to-3D Content Generation.** Driven by the aspiration to produce high-fidelity 3D content using semantic inputs like text prompts, the field of text-to-3D has garnered a significant boost in recent years (Poole et al. 2022). Ear-
lier approaches either align shapes and images in the latent space by CLIP supervision (Radford et al. 2021) to generate 3D geometries (Mohammad Khalid et al. 2022) or synthesize new perspectives (Jain et al. 2022), or they train text-conditioned 3D generative models from the ground up (Li et al. 2023). DreamFusion (Poole et al. 2022) first employs large-scale T2I models with a combination of score distillation sampling to distill the prior, and achieves impressive results. Magic3D (Lin et al. 2023) further improved the quality and performance of generated 3D shapes with a 2-step pipeline. TextMesh (Tsalicoglou et al. 2023) modify the 3D representation to extract detailed mesh. However, all these methods present semantic misalignment between the local content and global text description when editing, leaning towards distorted background and inconsistent results.

3D Content Editing. Semantic-driven 3D scene editing is a much harder task compared with 2D photo editing because of the high demand for multi-view consistency, the scarcity of paired 3D data and its entangled geometry and appearance. Previous approaches either rely on laborious annotation (Kania et al. 2022; Yang et al. 2022), only support object deformation or translation (Tschernezki et al. 2022; Kobayashi, Matsumoto, and Sitzmann 2022), or only perform global style transfer (Chen et al. 2022; Chiang et al. 2022; Fan et al. 2022; Huang et al. 2022) without strong semantic meaning. Recently, thanks to the development of score distillation sampling technique, text-guided editing has emerged as a promising direction with great potential. SKED (Mikaeli et al. 2023) possesses the capability to edit 3D scenes with multi-view sketches. Latent-NeRF (Metzer et al. 2023) and Fantasia3D (Chen et al. 2023) realize sketch-shape guidance by relaxed geometric constraints. Instruct-NeRF2NeRF (Haque et al. 2023) can edit an existing NeRF scene by iterative dataset update. However, it manipulates the entire space, and the preservation of undesired regions is absent. Vox-E (Sella et al. 2023) allows local edits on an existing NeRF, but it suffers from subpar editing quality and noticeable noise as shown in Section 4, because of coupling geometry and textures. Most related to our work, DreamEditor (Zhuang et al. 2023) locally edits a mesh-based neural field. However, it doesn’t achieve separable editing which is vital for instance reuse and part-wise control. Moreover, DreamEditor cannot change the number of vertices, supporting only minor shape insertion and replacement of objects of the same type (e.g., a horse to a deer). In contrast, our work not only brings about reasonable and noticeable geometric changes but also generates realistic appearances.

3 Method

As illustrated in Fig. 3, a complete object is conceptualized as a composition of base shape and learnable parts, wherein both of them possess their own geometry and texture, tailored for convenient instance reuse and part-wise control. Furthermore, a two-stage training strategy is adopted to sequentially learn the geometry and texture of the editable shape, to avoid the potential interference that can occur when geometry and texture learning are intertwined. For instance, in the case of zebra modeling, geometric protrusions might be learned instead of the desired black stripes. Such a disentangled representation not only stabilizes the training process but also yields high-fidelity geometry and textures, especially when compared to popular text-to-3D models.

3.1 Preliminary

Score Distillation Sampling. Score distillation sampling (SDS) is a way to distill the priors hidden in large T2I models for 3D generation proposed by DreamFusion (Poole et al. 2022). DreamFusion represents 3D scenes as a series of learnable parameters $\theta$. Utilizing a differentiable renderer, it converts the 3D scenes into 2D image sets $x$. Subsequently, it employs large-scale models $f$ to optimize the parameters of the 3D scenes with a score function as follows:

$$\nabla_\theta L_{SDS}(f, x) = \mathbb{E}_{t, \epsilon} \left[ w(t)(\hat{z}_\theta(x; t, \epsilon) - \epsilon) \frac{\partial \hat{z}}{\partial \theta} \right],$$

Fig. 3: An overview of FocalDreamer. (a) During geometry learning, given a base shape, we first initialize an ellipsoid as editable geometry within each focal region. Then we render the normal map of merged shape as shape encoding of pre-trained T2I models, to optimize the editable geometry according to prompts. (b) During appearance learning, resultant shape is rendered in a dual-path manner with base and editable textures. The outcomes are then blended by Pixel-wise Discriminative Mask for a unified appearance. (c) Several regularizations are introduced to improve the editing quality, including $L_{GF}$, $L_{CA}$, and $L_{SC}$. The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)
where \( w(t) \) controls the weight of SDS guidance depending on noise level \( t \), \( \hat{e}_d(z_t; y, t) \) and \( \epsilon \) are the predicted noise and actual noise, respectively. \( y \) is the condition.

**DMTet.** DMTet (Munkberg et al. 2022) is a hybrid representation that has two components, i.e., a deformable tetrahedral grid and a differentiable Marching Tetrahedral (MT) layer. The Signed Distance Function (SDF) values and the position offsets of deformable tetrahedral vertices are learnable, followed by the MT layer to extract meshes.

**3.2 Geometry Editing**

**Focal Region.** The starting point of our algorithm is a base shape (\( \Psi_b \) for geometry and \( \Gamma_h \) for texture) to be edited, which can be the reconstruction from images, crafted shapes by artists (Munkberg et al. 2022), and even the novel shapes from the generative method (Chen et al. 2023). Then the base model is modified by compositing with a new learnable texture, followed by the MT layer to extract meshes.

**Geometry Learning and Fusion.** We adopt DMTet as our 3D scene representation optimized by the prior knowledge distilled from pre-trained T2I model. More specifically, keeping the base shape \( \Psi_b(v_i) \) frozen, we parameterize the SDF values (inner is positive) of editable parts using MLP \( \Psi_e(v_i) \) for each vertex \( v_i \) within the tetrahedral grid. Subsequently, a soft geometry union (Quilez and Jeremias 2018) is performed between \( \Psi_b(v_i) \) and \( \Psi_e(v_i) \), resulting in \( \Psi_u(v_i) \) for a smooth junction:

\[
\Psi_u(v_i) = \max \{ \Psi_b(v_i), \Psi_e(v_i) \} + 0.1 \times \frac{h^2}{k},
\]

where \( h = \max \{ (k - |\Psi_b(v_i) - \Psi_e(v_i)|), 0 \} \),

where \( k \) determines the extent of the soft merge and is set to 0.15 by default. After geometry fusion, a differentiable MT layer transforms \( \Psi_u(v_i) \) and the vertex offset \( \Delta v_i \) into a triangular surface mesh \( M \). Finally, the rendered normal map \( n \) and the object mask \( o \) extracted from the mesh \( M \) are fed into pre-trained T2I models with SDS loss to update \( \Psi_e \):

\[
\nabla \Psi_e L_{SDS}(\phi, \tilde{n}) = \mathbb{E}_{t,x} \left[ w(t) \hat{e}_d(z_t; y, t) - \epsilon \partial \tilde{n} \frac{\partial \tilde{n}}{\partial \bar{n}} \right],
\]

where \( \phi \) parameterized pre-train T2I model, \( \tilde{n} \) represents the augmentation of \( n \) concatenated with \( o \), \( z^\phi \) is latent encoding of \( \hat{n} \). We observed using normal map \( n \) promotes the expression of geometric details and training stability (Chen et al. 2023). This improvement from \( n \) is partly attributed to disentangling the geometry from the intertwining of texture, and its sufficient expressiveness to depict complex geometric details.

**Geometric Concentration.** One of the main criteria for a proficient 3D editing algorithm is its ability to retain the geometry and color of the base object throughout the editing process. However, the aforementioned pipeline cannot ensure locality in editing. We have observed global changes and a loss of characteristics from the base shape (Fig. 7). To counteract it, we introduce distance-aware geometric focal loss \( L_{GF} \). During each iteration, a certain number of points \( p_i \in \mathbb{R}^3 \) are sampled outside the user-specified focal region \( \Omega_i \), with their SDF values \( \Psi_e(p_i) \) and their distances \( d_i \) to the focal region \( \Omega_i \). The objective of \( L_{GF} \) is punishing the editable shape when it produces topological structures \( (\Psi_e(p_i) > 0) \) outside \( \Omega_i \). Moreover, the closer \( p_i \) is to the target region, the less the penalty, for this distance-aware setting permits geometry to overrun beyond the rough focal region slightly. The geometric focal loss is defined as:

\[
L_{GF} = \mathbb{E}_{p_i \in \Omega_i'} \left[ (1 - e^{-\frac{d_i^2}{\sigma_1}}) \cdot \tanh\left( \frac{\max \{ \Psi_e(p_i) + \xi, 0 \} }{\sigma_2} \right) \right],
\]

where \( \sigma_1 = 0.05 \) and \( \sigma_2 = 0.01 \) control how sensitive the loss is, i.e., lower, lower \( \sigma_3 \) values tighten the constraint on the optimization such that only the editable region is modified strictly. The hyperparameter \( \xi \) is a small positive threshold to prevent topological structures from minor positive SDF values. For computational efficiency, we sample query points on the tetrahedral vertex \( v_i \), and pre-compute their distance \( d_i \) to \( \Omega_i \) before the geometry generation process begins.

**Collision Avoidance.** Another essential criterion is to respect the purity of the editing results, i.e., the editable shape should not overlap with the base shape, as they are semantically independent and distinct parts. We enforce it by penalizing the query points \( p_i \) that reside both within the learnable shape and the base shape with the collision avoidance loss:

\[
L_{CA} = \mathbb{E}_{p_i} \left[ \max \{ \Psi_b(p_i), 0 \} \cdot \max \{ \Psi_e(p_i), 0 \} \right].
\]

Intuitively, this reduces the likelihood of overlap between the editable shape and the original mesh, resulting in cleaner editing outcomes. For computational efficiency, we sample query points at \( v_i \) as the same as geometric focal Loss.

**3.3 Appearance Editing**

**Dual-path Physically Based Rendering.** After the optimization of the geometry network, the resultant mesh \( \mathcal{M} \) is obtained from the soft fusion and MT layer. Following Physically Based Rendering (PBR) material model, we use hash-grid-based texture neural fields \( \Gamma \) for \( \mathcal{M} \) to produce the diffuse term \( k_d \), the roughness and metallic term \( k_{rm} \), and the normal term \( k_n \) as \( (k_d, k_{rm}, k_n) = (\Gamma(p_i)) \). In order to retain the appearance of the base shape untouched, a naive and straightforward idea would be to initialize the learnable texture neural fields \( \Gamma_e \) with the base texture fields.
“a human skull wearing a pair of dark glasses”

“Flash Gordon wears the red velvet cape with a golden trim over his shoulder, highly detailed”

“a labrador wears a crown with sapphires”

“a deer standing on two separate wooden skateboards”

Figure 5: Comparison with SOTA image editing methods. The gray areas in input images indicate the inpainting region. We observed that 2D editing methods exhibit viewpoint inconsistent, and their quality varies with viewpoints.

<table>
<thead>
<tr>
<th>Method</th>
<th>CLIP$_{sim}$</th>
<th>CLIP$_{dir}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fantasia3D*</td>
<td>0.284</td>
<td>0.0180</td>
</tr>
<tr>
<td>Vox-E-Global</td>
<td>0.299</td>
<td>0.0204</td>
</tr>
<tr>
<td>Vox-E</td>
<td>0.293</td>
<td>0.0178</td>
</tr>
<tr>
<td>FocalDreamer (ours)</td>
<td><strong>0.329</strong></td>
<td><strong>0.0519</strong></td>
</tr>
</tbody>
</table>

Table 1: Quantitative evaluation results across 15 scenes.

$\Gamma_b$ derived from the base shape reconstruction, then the entire shape’s appearance is modeled by $\Gamma_e$ exclusively. However, this simple pipeline has two shortcomings: 1) As the number of iterations increases, it suffers from sub-optimal convergence and loss of the original material (in Fig. 7). In essence, the texture of the base shape isn’t adequately retained due to the overly strong knowledge supervision from T2I models. 2) Although learnable parts have independent semantics, such as “the wings”, their texture cannot be extracted alone. This impediment makes the reuse and driving of materials for these editable parts unfeasible.

To tackle this issue, we re-design the rendering pipeline in a dual-path manner. Central to this redesign is a Pixel-wise Discriminative Mask (PDM) generated in the rasterization process, which discerns whether each pixel comes from the face of the base mesh or the editable mesh. As depicted in Fig 3, throughout the dual-path rendering process, both parts are rendered based on their own neural texture fields, and the outcomes are then blended by PDM, which is called texture composition, culminating in a unified merged view. Similarly, the merged view is inputted into the T2I model for texture optimization with SDS loss. By truncating the gradient towards $\Gamma_b$, the texture of the base shape is precisely preserved, while the editable shape has its independent trainable texture $\Gamma_e$. Dual-path rendering balances the preservation of the base shape structure with flexible part-wise control, as well as the seamless integration of both parts.

**Style Consistency.** In some instances, local changes are anticipated to be realized seamlessly, as well as in a harmoniously coordinated style, as shown in Fig. 7. This problem is modeled as follows: let $M_e \in \mathbb{R}^3$ be a closed subspace to represent the editable parts with boundary $\partial M_e$. Let $f^*$ be a known mapping function defined over $\mathbb{R}^3$ minus the interior of $M_e$ to be preserved, and let $f$ be the unknown function defined over the interior of $M_e$. A classical interpolant $f$ is defined as the solution (Pérez, Gangnet, and Blake 2003):

$$\min \int \int_{M_e} |\nabla f|^2 \text{ with } f|_{\partial M_e} = f^*|_{\partial M_e}.$$  

(8)

We propose two consistency regularization items to imi-
\begin{table}[h]
\centering
\begin{tabular}{ |c|c|c|c|c|c|c| }
\hline
\(\mathcal{L}_G\) & \(\mathcal{L}_CA\) & \(\mathcal{L}_{SC}\) & Dual-path Render & CLIP\(_{sim}\) & CLIP\(_{dir}\) \\
\hline
✓ & ✓ & ✓ & ✓ & 0.312* & 0.0402* \\
✓ & ✓ & ✓ & ✓ & 0.310* & 0.0495* \\
✓ & ✓ & ✓ & ✓ & 0.316 & 0.0433 \\
✓ & ✓ & ✓ & ✓ & 0.329 & 0.0517 \\
✓ & ✓ & ✓ & ✓ & 0.313 & 0.0401 \\
x & ✓ & ✓ & ✓ & 0.329 & 0.0519 \\
\hline
\end{tabular}
\caption{Ablation study. Since not all scenes require style consistency, we report the editings require \(\mathcal{L}_{SC}\) with *.}
\end{table}

4 Experiments

4.1 Experimental Setups

Implementation Details. We use the Stable Diffusion implementation by HuggingFace Diffusers for SDS, and adopt DMTet to learn geometry and texture separately with NVDiffRast as a differentiable renderer. FocalDreamer usually takes less than 30 minutes (3000 steps) for geometry and 20 minutes (2000 steps) for texture to converge on 4 Nvidia RTX 3090 GPUs, where we use AdamW optimizer with the respective learning rates of \(1 \times 10^{-3}\) and \(1 \times 10^{-2}\). UV edge padding techniques are utilized to remove the seams in the texture maps. More details are provided in the appendix.

Synthetic Object Dataset. We assemble the dataset with 15 high-quality meshes found on the internet. We paired each object in our dataset with a detailed edit prompt to showcase our approach’s ability to perform expressive and meaningful changes to both geometry and texture. 

Evaluation Criteria. Following Vox-E, we report auxiliary quantitative metrics on our dataset: (1) CLIP Similarity (CLIP\(_{sim}\)) measures the alignment of the performed 3D edits with the text descriptions, and (2) CLIP Direction Similarity (CLIP\(_{dir}\)) evaluates the edits with the editing directions from the input to edit results, by measuring the directional CLIP similarity between changes of text and 3D shapes, first introduced by (Gal et al. 2022).

Baselines. We compare FocalDreamer with three baselines. (1) Fantasia3D*: as claimed in Fantasia3D, it is able to generate shapes initialized with a low-quality customized 3D mesh. In order to additionally endow it with preservation of texture from base shape, the texture field \(\Gamma(p_i)\) is supervised by base texture with reconstruction loss on the base mesh surface, as one of the baselines. (2) Vox-E (Sella et al. 2023): to show our superior editing within desired regions, SOTA editing work Vox-E is also compared. To the best of our knowledge, Vox-E is the only open-source method that directly performs text-guided localized edits for 3D objects. (3) Vox-E-Global: Vox-E also supports global editing to better align with the prompts without constraining from base shape. More details are provided in the appendix.

4.2 Qualitative Results

The qualitative comparison with 3D editing baselines is shown in Fig. 4 over our dataset. As illustrated in the figure, Fantasia3D* results in an appearance vastly different from the base mesh, even with the texture reconstruction loss, because the whole shape is re-optimized according to prompts. While Vox-E-Global occasionally produces edits that align with prompts, it suffers from subpar editing quality and noticeable outliers. Vox-E demonstrates a limited capacity to filter out undesired changes and noise based on Vox-E-Global, since it heavily relies on a keyword, such as cape or glasses. Vox-E sometimes misidentifies the focal regions, i.e., placing glasses on the top of the skull. In contrast to them, our editings align perfectly with the prompts while faithfully preserving the details of base mesh, achieving precise and meaningful changes to both geometry and texture.

2D Image Editing Comparisons. We demonstrate that 2D image editing methods cannot effectively handle 3D object editing tasks, because 2D editing does not yield satisfactory view-consistent results. We sample renderings from three different viewpoints and apply SOTA image editing methods, namely Instruct Pix2Pix (IP2P) (Brooks, Holynski, and Efros 2023) and ControlNet-inpainting (ControlNet) (Zhang and Agrawala 2023). We input the same prompts in Fig. 2 for FocalDreamer, ControlNet and IP2P. As depicted in Fig. 5, the quality of editing by 2D methods drops significantly from less canonical views (e.g., the turtle’s left view), and they severely lack view-consistency.

4.3 Quantitative Results

We perform a quantitative evaluation in Tab. 1 on our dataset. To perform a fair comparison, all metrics are calculated with renderings from the same 100 views across different methods. As illustrated in the table, FocalDreamer achieves noticeably higher CLIP\(_{dir}\). This is attributed to its capability to accurately execute the desired editing direction, primarily due to the geometric concentration. Additionally, our editing fidelity (CLIP\(_{sim}\)) stands out as the best, stemming from the enhanced part-wise details brought by the separable framework and decoupled learning.

User Study. While CLIP mainly evaluates the matching degree of rendered views and text prompts, it fails to assess the extent to which the base shape is properly preserved. We conduct user studies with 65 participants to evaluate different methods based on user preferences across 15 scenes. We ask the participants to give a preference score (range from 1 to 10) in terms of prompt relevance and base shape preservation for 5 random views per scene from anonymized methods’ generation. As shown in Fig. 6, we report the distribu-
4.4 Ablation Study

We conduct the ablation study both qualitatively and quantitatively. By setting $\mathcal{L}_{GF}$, $\mathcal{L}_{CA}$ and $\mathcal{L}_{SC}$ to zero respectively, we investigated the effects of our proposed Geometric Concentration, Collision Avoidance, and Style Consistency strategies. To validate the dual-path rendering, we employ the single rendering outlined in Section 3.3. Specifically, it involves rendering the entire shape with a learnable texture $\Gamma_e$, which is initialized with the base texture $\Gamma_b$.

As illustrated in Fig. 7 and Tab. 2, $\mathcal{L}_{GF}$ significantly constrains geometric alterations outside the focal region, resulting in localized edits. $\mathcal{L}_{CA}$ effectively prevents undesirable geometric overlap within the base mesh, especially at the junction like the root of wings and capes. Since $\mathcal{L}_{CA}$ predominantly affects the purity of the editable part and has minimal impact on the overall appearance, its quantitative metrics closely align with the full model. In the absence of dual-path rendering, the base mesh texture experiences unintended alterations due to the update of the whole texture network during appearance learning. Moreover, editing with $\mathcal{L}_{SC}$ exhibits a harmonious overall style and nature transition in certain instances, but it is not universally required (e.g., a butterfly over a tree stump). In Tab. 2, we use * to denote scenes that require $\mathcal{L}_{SC}$ for a fair comparison.

Progressive Editing. Our method can be used as a sequential editor for users’ requirements, and progressively edits the base mesh. In Fig. 9, we exhibit a two-step editing by first generating two wings on the horse, followed by adding a horn.

Geometry Union Sensitivity. We also demonstrate the smoothness of the junction between the editable part and base mesh with various $k$ (Eq. 3 and 4) in Fig. 8. It is evident that larger $k$ leads to a more natural but pronounced transition region. We set $k = 0.15$ for a moderate transition.
Acknowledgements

This work was supported by National Science Foundation of China (U20B2072, 61976137). This work was also partly supported by SJTU Medical Engineering Cross Research Grant YG2021ZD18.

References


