Image-Guided Geometric Stylization of 3D Meshes

Changwoon Choi*[1] Hyunsoo Lee*[1] Clément Jambon[2] Yael Vinker[2] Young Min Kim†[1]
[1]Seoul National University [2]Massachusetts Institute of Technology (MIT)

Figure 1. Our method enables stylization of 3D meshes driven by image style references. The target stylized meshes retain the coarse structure and semantics of the input mesh while incorporating internal geometry derived from the style reference.

Abstract

Recent generative models can create visually plausible 3D representations of objects. However, the generation process often allows for implicit control signals, such as contextual descriptions, and rarely supports bold geometric distortions beyond existing data distributions. We propose a geometric stylization framework that deforms a 3D mesh, allowing it to express the style of an image. While style is inherently ambiguous, we utilize pre-trained diffusion models to extract an abstract representation of the provided image. Our coarse-to-fine stylization pipeline can drastically deform the input 3D model to express a diverse range of geometric variations while retaining the valid topology of the original mesh and part-level semantics. We also propose an approximate VAE encoder that provides efficient and reliable gradients from mesh renderings. Extensive experiments demonstrate that our method can create stylized 3D meshes that reflect unique geometric features of the pictured assets, such as expressive poses and silhouettes, thereby supporting the creation of distinctive artistic 3D creations. Project page: https://changwoonchoi.github.io/GeoStyle

1. Introduction

Despite impressive advancements in 3D generative models, it is not trivial to support artistic creation of 3D models with existing AI tools. The artistic style is beyond what one can achieve by interpolating or composing existing data distributions, and the unique component embedded within the imagined creation cannot be communicated intuitively. Previous attempts at stylization focus on altering the local appearances of the given global structure. Image stylization works distinguish style from content, maintaining the overall layout of the original image while altering only local patch statistics [9]. Similarly, 3D stylization methods incorporate text descriptions to guide local geometric variations on the surface [6, 40], or produce specific geometric characteristics with handcrafted regularizations [22, 32, 33].

We expand the notion of style beyond high-frequency textures to embrace geometric features of various scales as components of a unique style. For example, in Fig. 1, the distinctive silhouette of Bourgeois's spider or the rigid structural characteristics of a fire hydrant cannot be described by local texture. Such a diverse range of variations requires a holistic analysis, whose geometric characteristics are challenging to describe unambiguously with an input text, as shown in Fig. 2. We utilize reference images as a means of explicit description to inspire 3D stylization intended by users. With the power of image diffusion models,

*Equal contribution.
†Young Min Kim is the corresponding author.

Figure 2. Text-based deformation often struggles to capture and transfer geometric style, whereas our image-guided framework can transfer intended aesthetic from rich and specific visual cues.

the image clue can be transformed into a style-specific optimizer for free, which can guide geometric stylization.

Instead of generating stylized geometry from scratch, we formulate geometric stylization as the task of deforming a user-provided mesh model. The deformation maintains the original manifold topology despite a significant change in structure. While the popular choice of volumetric representations can generate visually plausible 3D shapes with a differentiable rendering pipeline, these representations often lack a rigorous topological structure. In contrast, we can start from a valid mesh topology and maintain compatibility with valuable assets, such as UV maps and motion rigs, as well as rich geometric processing pipelines for smoothing, upsampling, and re-meshing.

Our stylization framework requires the extraction of geometric style and significant yet valid deformation of the given mesh. We define the stylistic component as an abstract feature of a pre-trained large-scale diffusion model [44] and extract the style of the reference image as LoRA weights [15]. Then, Score Distillation Sampling (SDS) [45] drives mesh deformation to align with the reference style. We propose using an approximate VAE encoder of the latent-based diffusion model, which is crucial for stabilizing the optimization in practice. Our deformation pipeline first encourages semantically coherent deformation per part at a coarse level, followed by finer deviation via Jacobian optimization [1]. We optionally preserve symmetry, which maintains internal consistency. Together, these components form a general and practical framework for transferring high-level geometric style from 2D images to 3D meshes, paving the way for intuitive, reference-driven 3D content creation.

2. Related Work

Style transfer. Style transfer is the task of applying the visual style of one input to the content of another, producing a stylized output that preserves the original content while adopting the target style. The seminal work by Gatys et al. [9] and its follow-ups [4, 11, 23, 25–27, 29, 31, 38, 47] formulated style transfer as an iterative optimization problem that minimizes content and style losses defined on deep features [52]. Another line of works [2, 17, 30, 43, 51] employs feed-forward networks that learn to approximate the optimization-based process with a neural network, enabling efficient stylization. While these approaches show promising results, they primarily focus on matching patch statistics and transferring high-frequency appearance attributes such as color, texture and brushstroke patterns.

Image style transfer has been naturally extended to 3D style transfer, aiming to stylize a 3D scene so that its rendered appearance matches the visual characteristics of a reference image. Recent works have explored style transfer across various 3D representations, including point clouds [16, 41], meshes [14], Neural Radiance Fields [5, 18, 42, 57, 58] and 3D Gaussian Splatting [35, 56]. Similar to image style transfer, most 3D style transfer approaches mainly focus on transferring high-frequency surface textures while overlooking the role of geometry in conveying style. Only a few works have explored geometric style transfer; however, they are restricted to the image domain [37, 54] or limited to specific object categories [54, 55].

3D mesh stylization by deformation. In contrast to texture-based style transfer on 3D meshes, some approaches stylize meshes by deforming them. Early works [19, 34] and its follow-ups [10] optimize positions of mesh vertices by minimizing image-space style loss [9, 24] with differentiable rendering technique. Hertz et al. [12] transfer geometric textures from reference meshes to source meshes. However, they are limited to transferring high-frequency geometric details and only achieve local vertex displacements. Another line of work [22, 32, 33] stylize meshes with large-scale deformation by minimizing hand-crafted heuristic regularization terms, which are limited to representing specific styles. Recent works [6, 8, 20] utilize large image models, including CLIP [46] and pixel-space diffusion model [3], for mesh deformation. By leveraging powerful large image models which have high-level and semantic understanding of shapes, the methods successfully deform the source mesh into various concepts or styles. However, the inherent ambiguity of text descriptions often makes it difficult to precisely control the stylization or reflect the user's stylistic intent.

3. Method

Given a source mesh and reference style images, our method deforms the mesh to exhibit the geometric style depicted in the input images. An overview of our pipeline is illustrated in Fig. 3. We first extract the abstracted style component from input images as a LoRA weight of latent diffusion model. Then we deform the mesh by minimizing the SDS loss from a style-infused latent diffusion model with an efficient low-rank approximation of the encoder. The deformation adapts our novel coarse-to-fine strategy,

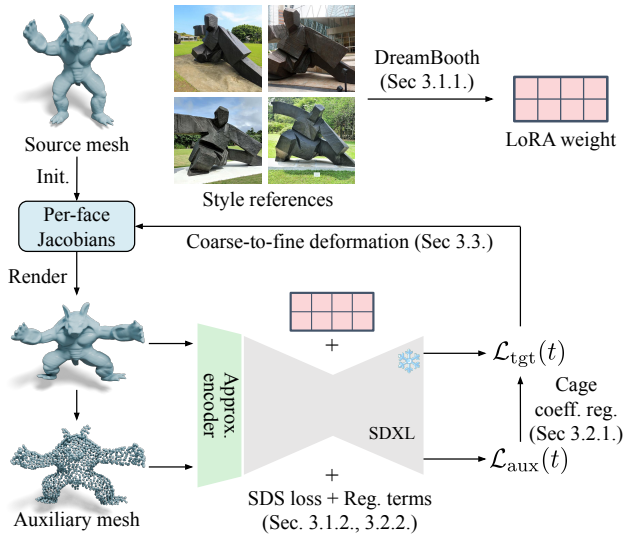

Figure 3. Method overview. We first extract geometric style from reference images by training LoRA using a DreamBooth-style objective (Sec. 3.1.1). We optimize per-face Jacobians to transfer geometric style to source mesh. The optimization is guided by SDS loss (Sec. 3.1.2). To handle both large structural deformation and fine-grained details, we adapt a coarse-to-fine deformation (Sec. 3.3), along with the cage-based regularization (Sec. 3.2.1) and optional symmetry regularization (Sec. 3.2.2).

which effectively preserves content identity and local features without deteriorating the mesh structure. In addition to per-face optimization of the mesh to minimize the proposed loss, we introduce an auxiliary representation with coarse samples, and allow large-scale changes with part-level regularization.

3.1. Style Matching

The primary subtask in geometric stylization is to extract the *style* encoded in the reference images and to optimize the source mesh. Although 'style' is difficult to define explicitly, we leverage the generative priors of pre-trained diffusion models to extract a coherent abstraction of it.

3.1.1. Image Style Extraction

DreamBooth [50] provides a critical mechanism for this – by fine-tuning a pretrained diffusion model on a small set of reference images, it encourages the model to capture the distinctive attributes that define the subject's appearance:

$$\mathcal{L}_{\text{DreamBooth}} = \mathbb{E}_{\mathbf{x},\mathbf{c},\epsilon,t}[w_t \|\hat{\mathbf{x}}_\theta(\alpha_t \mathbf{x} + \sigma_t \epsilon, \mathbf{c}) - \mathbf{x}\|_2^2], \quad (1)$$

where $t \sim \text{Uniform}([0,1])$, $\mathbf{x}$ denotes a reference image, $\mathbf{c}$ is the conditioning signal such as a text prompt, and α_t, σ_t, w_t are the noise scheduling parameters of the diffusion model [13, 53]. Importantly, the attributes captured through this process extend beyond surface-level texture or fine-scale appearance: DreamBooth has been shown to preserve coherent shape, proportions, and other structural traits that form the global identity of the subject. This aligns with our goal for *geometric stylization*, which incorporates both local details and global shape priors. We further reduce the computational overhead by restricting the parameter updates to low-rank adapters (LoRA) [15] inserted into the diffusion model's U-Net [49]. Once it is extracted from the input, the compact abstraction serves as the stylization target that the geometry should match.

3.1.2. SDS Loss with an Approximated VAE Encoder

We deform the input mesh based on the Score Distillation Sampling (SDS) loss that leverages a pre-trained diffusion model [48]. Following DreamFusion [45], the SDS loss is derived by omitting the Jacobian term of the U-Net [49] in the gradient of the original diffusion training loss:

$$\mathcal{L}_{\text{Diff}} = \mathbb{E}_{\mathbf{c},\epsilon,t}[w_t \|\epsilon_\theta(\mathbf{z}_t, \mathbf{c}, t) - \epsilon\|_2^2], \quad (2)$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, ϵ_θ represents the noise prediction network of the diffusion model, and $\mathbf{z}_t$ is the noised latent of ground-truth data. This formulation provides an update direction that follows the score function of the diffusion model toward high-density regions of the data distribution, without requiring backpropagation through the pretrained diffusion model.

To deform the source mesh, we adopt the per-face Jacobian parameterization introduced by Neural Jacobian Fields [1]. More concretely, we avoid directly optimizing vertex positions, which has been shown to be unstable and prone to producing local, fragmented deformations [8]. Instead, we represent the deformation using per-face Jacobians $\mathbf{J}_i \in \mathbb{R}^{3\times3}$, which enables more coherent and larger-scale shape changes. The deformed vertex positions are then recovered from the deformation map ϕ^* by solving the following Poisson equation:

$$\phi^* = \min_\phi \sum_i |t_i| \|\nabla_i \phi - \mathbf{J}_i\|^2. \quad (3)$$

With this parameterization, the gradient of the SDS loss with respect to the per-face Jacobians $\{\mathbf{J}_i\}$ is expressed as

$$\nabla_{\mathbf{J}_i} \mathcal{L}_{\text{SDS}} = \mathbb{E}_{\mathbf{c},\epsilon,t}\left[w_t(\epsilon_\theta(\mathbf{z}_t, \mathbf{c}, t) - \epsilon) \frac{\partial \mathbf{z}_t}{\partial \mathbf{J}_i} \right], \quad (4)$$

where $\mathbf{z}_t$ is the noisy latent obtained from the rendered mesh image, computed with per-face Jacobian $\mathbf{J}_i \in \mathbb{R}^{3\times3}$ of a triangular mesh.

Approximated VAE encoder. The geometric stylization process extracts features from mesh renderings and propagates gradients to refine the underlying geometry. Although latent diffusion models such as Stable Diffusion XL

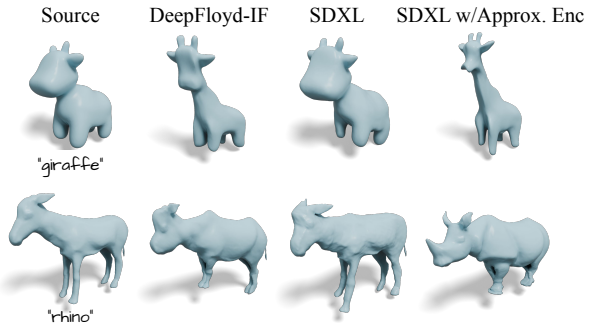

Figure 4. Effect of model choices on text-guided mesh deformation. For each source mesh, we apply SDS [45] guidance using (1) DeepFloyd-IF [3], (2) SDXL [44] with the original VAE [21], and (3) SDXL with our approximated encoder. Our setup successfully transfers the intended semantic shape, while the alternatives either distort geometry or fail to deform the global structure.

(SDXL) [44] provide strong generative priors, their large VAE encoder–decoder architecture can make gradient propagation less effective for geometry optimization. Previous work has shown that approximating components of the VAE in latent diffusion models can significantly accelerate 3D reconstruction tasks [39]. Building on this insight, we develop an efficient approximation of the VAE encoder tailored to our setting, enabling fast and stable encoding of rendered images into the latent space during optimization.

Let $\mathbf{x} \in \mathbb{R}^{3\times H\times W}$ be a rendered image of the source mesh and $\mathbf{z} \in \mathbb{R}^{4\times H\times W}$ be its corresponding latent encoded from the SDXL VAE. We then compute a matrix $\mathbf{A} \in \mathbb{R}^{4\times 4}$ such that $\mathbf{z} \simeq \mathbf{A}\bar{\mathbf{x}}$, where $\bar{\mathbf{x}} = [\mathbf{x}; \mathbf{1}]$ is obtained by concatenating a one matrix along the channel dimension, allowing to model affine components. Given N rendered image-latent pairs $\{(\mathbf{x}_i, \mathbf{z}_i)\}_{i=1}^{N}$, we fit $\mathbf{A}^*$ via least squares:

$$\mathbf{A}^* = \arg\min_{\mathbf{A}} \sum_{i=1}^{N} \|\mathbf{z}_i - \mathbf{A}\bar{\mathbf{x}}_i\|_2^2, \tag{5}$$

where we use $N = 500$ in practice.

We emphasize that this approximation strategy is crucial, as even text-guided mesh deformation fails without it. To justify our choice of using SDXL with an approximated VAE encoder, we compare three setups on text-based mesh deformation: (1) DeepFloyd-IF [3] – the pixel diffusion model used in MeshUp [20], (2) SDXL with its native VAE, and (3) SDXL with our approximated encoder. As shown in Fig. 4, the naïve SDXL setup struggles to deform the global shape meaningfully, and the DeepFloyd-IF variant often shows suboptimal results. In contrast, our setup yields semantically aligned deformations, demonstrating its effectiveness. Therefore, we adopt SDXL with the approximated encoder for all subsequent experiments.

3.2. Regularization with Identity Preservation

By allowing deformation to match the style of the input images, we can transform the mesh into the desired style. While the Jacobian field in Eq. (3) can progressively update the surface details, extreme deformation can deteriorate the overall structure when the reference image is significantly different from the initial geometry, as shown in the second row of Fig. 8. We propose a coarse-to-fine strategy, enabling large-scale changes at an early stage. At a coarse level, we define a cage loss $\mathcal{L}_{\text{cage}}$, which creates locally coherent structures and preserves the semantics of the geometry (Sec. 3.2.1). We assume that the part-level decomposition of the mesh can provide clues for relative semantics, facilitating content preservation, and define coarse-level deformation using cages per part. Then we gradually increase the relative contribution of the Jacobian optimization $\mathcal{L}_{\text{SDS}}$, which refines fine-scale details. Optionally, we detect the reflective symmetry of the input mesh and preserve it (Sec. 3.2.2).

3.2.1. Auxiliary Mesh and Cage-Guided Deformation

In addition to per-face Jacobians, cage-guided deformation coherently moves the large-scale semantic structures. To disregard the detailed triangulation, we process the coarse-level changes on an auxiliary mesh composed of spheres extracted from the vertex samples of the original mesh. Additionally, we extract semantic mesh parts using PartField [36]. Then we fit Oriented Bounding Boxes (OBBs) aligned with the part segmentation, denoted as $\{\mathcal{C}_l\}_{l=1}^{L}$.

The coarse deformation is parameterized by scale s_l, rotation R_l, and translation T_l of each OBB C_l. At each optimization step, we first update the OBB parameters $\{s_l, R_l, T_l\}_{l=1}^{L}$ using the SDS loss calculated with the rendered view of the auxiliary mesh, denoted as $\mathcal{L}_{\text{SDS}}^{\text{aux}}$. The centers of the auxiliary spheres are directly updated by applying the optimized cage transformations. Concretely, a sphere center with initial coordinate $\mathbf{p} = (x, y, z)$ is translated into $\mathbf{p}' = (x', y', z')$ as $\mathbf{p}' = s_l\mathbf{R}_l\mathbf{p} + \mathbf{T}_l$.

The part-wise transform is transferred from the auxiliary mesh to the deformation of the source mesh, guided by cages. We define cage coefficients $\mathbf{W}_i = [w_{i1}, ..., w_{i8}]^\top$ that satisfy

$$\mathbf{v}_i = \sum_{j=1}^{8} w_{ij}\mathbf{c}_{lj}, \tag{6}$$

where these coefficients indicate how much influence the j-th OBB corner $\mathbf{c}_{lj}$ has on the position of mesh vertex $\mathbf{v}_i$. Note that they satisfy $\sum_{j=1}^{8} w_{ij} = 1$. We regularize cage coefficients of the target mesh to follow those of the auxiliary mesh by minimizing the following loss function:

$$\mathcal{L}_{\text{cage}} = \frac{1}{L}\sum_{l=1}^{L}\frac{1}{|\mathcal{C}_l|}\sum_{\mathbf{v}_i\in\mathcal{C}_l}\|\mathbf{W}_i^{\text{aux}} - \mathbf{W}_i^{\text{tgt}}\|_2^2, \tag{7}$$

where $|\mathcal{C}_l|$ is the number of vertices in part $\mathcal{C}_l$, and $\mathbf{W}_i^{\text{aux}}$, $\mathbf{W}_i^{\text{tgt}}$ are cage coefficients calculated using the updated OBBs of the auxiliary mesh and target mesh, respectively. The auxiliary mesh and cage coefficient regularization enable stable and semantically coherent deformations under SDS optimization.

3.2.2. Symmetry Regularization

We allow users to optionally enforce symmetry during optimization if the source mesh exhibits internal symmetry. We detect reflectional symmetry based on the vertices of the source mesh. We apply PCA to the vertices $\{\mathbf{v}_i\}_{i=1}^{V}$ and obtain the principal axes $\{\mathbf{a}_k\}_{k=1}^{3}$. For each axis $\mathbf{a}_k$, we define a reflection plane $\mathbf{\Pi}_k$ with normal $\mathbf{a}_k$ passing through the centroid $\bar{\mathbf{v}} = \frac{1}{V}\sum_i \mathbf{v}_i$. Then each vertex $\mathbf{v}_i$ is mirrored across $\mathbf{\Pi}_k$ onto $\mathbf{v}_{\text{mir}(i,k)}$, and its nearest vertex $\mathbf{v}_{j(i,k)}$ is identified. We consider $\mathbf{\Pi}_k$ to be a valid symmetry plane when the following two conditions hold:

$$\|\mathbf{v}_{\text{mir}(i,k)} - \mathbf{v}_{j(i,k)}\|_2 < \tau_1, \forall i \quad \text{and} \quad \sum_i \|\mathbf{v}_{\text{mir}(i,k)} - \mathbf{v}_{j(i,k)}\|_2 < \tau_2, \tag{8}$$

where τ_1 and τ_2 are threshold values. We denote the set containing the symmetric pairs $\{(i, j(i,k))\}$ as $\mathcal{P}_k$.

Once the symmetry is detected, we introduce a symmetry loss consisting of two regularization terms. For each symmetric pair $(i, j(i,k)) \in \mathcal{P}_k$, we force the midpoint of the symmetric pair to lie on a common plane $\tilde{\mathbf{\Pi}}_k$ (which can be changed from the initial symmetric plane). The first loss term penalizes the deviation of midpoints from this plane:

$$\mathcal{L}_{\text{mid}} = \sum_k \frac{1}{|\mathcal{P}_k|} \sum_{(i,j(i))\in\mathcal{P}_k} |\tilde{\mathbf{n}}_k^\top (\mathbf{m}_{i,k} - \bar{\mathbf{v}})|^2. \tag{9}$$

Using the calculated midpoints $\mathbf{m}_{i,k} = \frac{1}{2}(\mathbf{v}_i + \mathbf{v}_{j(i,k)})$, we calculate a normal vector $\tilde{\mathbf{n}}_k$ of common plane $\tilde{\mathbf{\Pi}}_k$ to all midpoints by performing SVD on their covariance matrix. The second term encourages the direction vector between the symmetric pair, $\mathbf{d}_{i,k} = \mathbf{v}_i - \mathbf{v}_{j(i,k)}$, to be orthogonal to $\tilde{\mathbf{n}}_k$ by minimizing

$$\mathcal{L}_{\text{dir}} = \sum_k \frac{1}{|\mathcal{P}_k|} \sum_{(i,j(i))\in\mathcal{P}_k} (1 - |\tilde{\mathbf{n}}_k^\top \hat{\mathbf{d}}_{i,k}|), \tag{10}$$

where $\hat{\mathbf{d}}_{i,k} = \mathbf{d}_{i,k}/\|\mathbf{d}_{i,k}\|$. The complete symmetry loss is given by: $\mathcal{L}_{\text{sym}} = \mathcal{L}_{\text{mid}} + \mathcal{L}_{\text{dir}}$. Note that symmetry loss is also defined for the auxiliary mesh by treating the centers of spheres as $\{\mathbf{v}_i\}$, and is denoted as $\mathcal{L}_{\text{sym}}^{\text{aux}}$.

3.3. Coarse-to-Fine Deformation Pipeline

In the coarse stage, we optimize the scale, rotation, and translation parameters of the cages of the auxiliary mesh using the following loss function:

$$\mathcal{L}_{\text{aux}} = \lambda_1 \mathcal{L}_{\text{SDS}}^{\text{aux}} + \lambda_2 \mathcal{L}_{\text{sym}}^{\text{aux}}, \tag{11}$$

where λ_1 and λ_2 are hyperparameters. With the optimized cages, we calculate $\mathcal{L}_{\text{cage}}$ by following Eq. (7). Then, the target mesh is optimized with Eq. (12):

$$\mathcal{L}_{\text{tgt}}(t) = \lambda_3 \mathcal{L}_{\text{SDS}} + \lambda_4 \mathcal{L}_{\text{reg}} + \lambda_5 \mathcal{L}_{\text{sym}} + \lambda_6(t) \mathcal{L}_{\text{cage}}, \tag{12}$$

where $t \in (0, N_1]$ denotes the optimization iteration, and $\lambda_3, \lambda_4, \lambda_5$ are constant hyperparameters. We linearly decay $\lambda_6(t)$ from its initial value λ_6 as follows:

$$\lambda_6(t) = \lambda_6 \left(1 - 0.99t/N_1\right). \tag{13}$$

Here, $\mathcal{L}_{\text{reg}}$ is a Jacobian regularization term introduced in TextDeformer [8] that prevents the deformed mesh from deviating excessively from the source mesh by encouraging $\{\mathbf{J}_i\}$ to follow the identity matrix. In the fine stage, we do not regularize cage coefficients and minimize:

$$\mathcal{L}_{\text{tgt}} = \lambda_3 \mathcal{L}_{\text{SDS}} + \lambda_4 \mathcal{L}_{\text{reg}} + \lambda_5 \mathcal{L}_{\text{sym}}, \tag{14}$$

where $t \in (N_1, N_2]$. This stage applies fine-grained adjustments to capture the geometric style of the reference image while preserving the large-scale translations established in the coarse stage.

4. Experiments

In this section, we first demonstrate the superiority of our method over baselines through a user study and qualitative comparisons. We further highlight the importance of the proposed components via ablation studies. Finally, we show that our approach flexibly incorporates additional conditioning signals such as texts and user-selected parts.

4.1. Implementation Details

We train LoRA [15] module of rank 16 with DreamBooth [50] using 4-12 reference images. The source meshes used in our experiments typically contain 2k-20k vertices. During optimization, we render meshes with differentiable rasterizer [28]. Users can adaptively select the number of semantic parts segmented by PartField [36] for cage coefficient regularization. Details of the experimental setup are provided in the Appendix A.

4.2. Comparative Evaluation

Quantitative evaluation. We compare our method against four baselines: Paparazzi [34], Neural 3D Mesh Renderer [19], MeshUp [20], Text2Mesh [40], and TextDeformer [8]. Since our task focuses on geometric stylization from image references rather than text prompts, we adapt Text2Mesh and TextDeformer by replacing their CLIP [46] text embeddings with CLIP image embeddings of the reference style references. We compute the loss with the same set of reference images as those used during LoRA [15] training, and we use the averaged loss value for optimization. For MeshUp, we leverage the Textual Inversion [7]

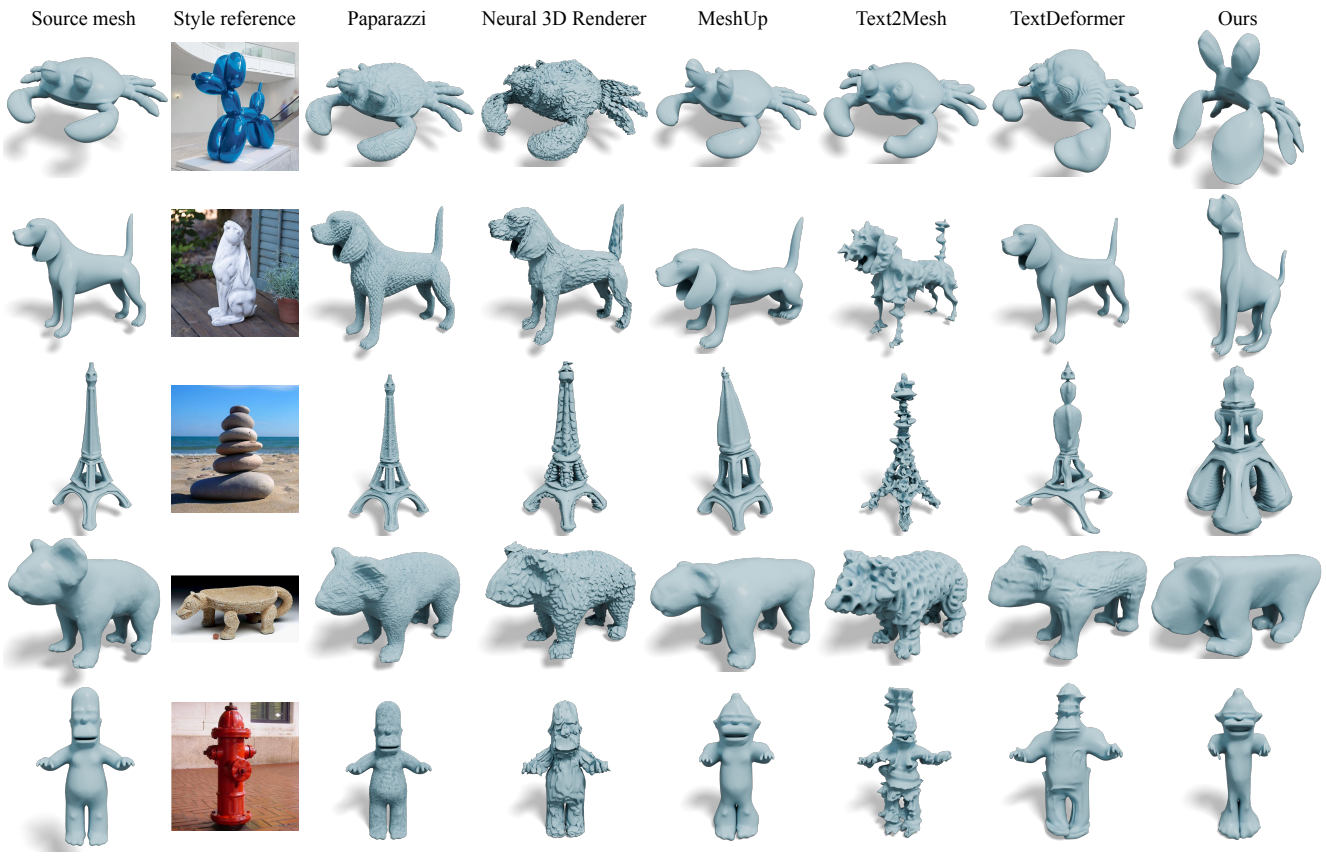

Figure 5. Qualitative comparison of the proposed method against baselines [8, 19, 20, 34, 40]. Our method achieves expressive geometric deformation, accurately reflecting both coarse structure and fine detail from the style reference while preserving the identity of source mesh, whereas baselines struggle to capture the intended geometry.

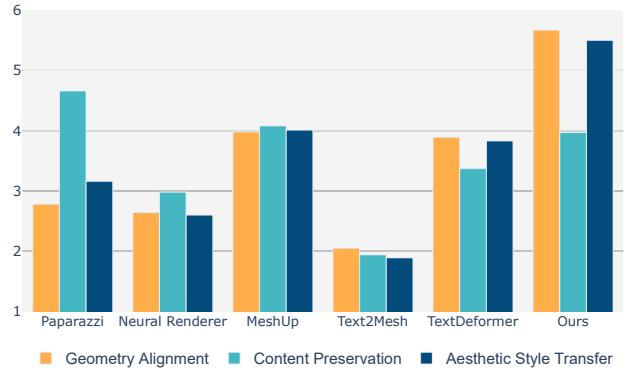

Figure 6. User study results. We examine each method using three criteria, including the measurement of geometry alignment, content preservation, and aesthetic style transfer.

technique to obtain an optimized token, then use it with the pretrained DeepFloyd-IF [3] for deformation. Then we quantitatively evaluate the proposed method with baselines through a perceptual user study. A total of 32 participants were gathered, and each participant was requested to rank the outputs of each method for 8 samples based on three criteria: (1) how well the geometry aligns with the style reference, (2) how faithfully the content of the source mesh is preserved, and (3) how effectively the aesthetic style is transferred. For every sample, we converted ranks into scores in descending order. We present the details of user study in Appendix B. As shown in Fig. 6, our method achieves the best perceptual ranking in terms of geometric alignment and aesthetic style transfer. We note that baselines may score higher in content preservation since some of them struggle to produce geometric structural changes, resulting in meshes that remain close to the source mesh without reflecting the desired deformation.

Qualitative evaluation. We visualize the qualitative comparisons in Fig. 5. Paparazzi and Neural 3D Mesh Renderer primarily perform texture-oriented style transfer, and therefore struggle to induce a desired geometric deformation, resulting in only local shape variations. Text2Mesh tends to produce noisy artifacts due to its direct optimization over vertex coordinates and color rather than structured

Figure 7. Qualitative geometric stylization results using diverse source mesh and style references. Our method effectively transfers the geometric style from the reference image while preserving the overall semantics of the source mesh.

per-face Jacobians. TextDeformer, which uses Jacobian-based deformation, still fails to capture complex geometric styles, highlighting the limited capability of CLIP-based guidance. MeshUp, while using the SDS [45] loss, relies on a pixel-level diffusion model rather than SDXL [44] and does not incorporate the sophisticated techniques used in our method, thus still yielding suboptimal results. In contrast, our method effectively transforms the global structure, pose, and geometry of the style reference while maintaining the semantic content of the source mesh. We visualize additional results in Figs. 1 and 7 and Appendix C.

4.3. Ablation Study

We validate the effectiveness of the proposed components through ablation studies, with qualitative results in Fig. 8. Firstly, we examine the role of the approximated VAE [21] encoder of our image-guided geometric deformation. As shown in the first row, using the original SDXL VAE [44] fails to produce plausible transforms and tends to remain close to the source mesh, indicating that the approximated VAE encoder is crucial for inducing meaningful shape deformation. Second, we evaluate the impact of cage coefficient regularization. As visualized in the second row, without this regularization, the deformed meshes often exhibit distorted geometry and fail to accurately reflect the geometric style encoded from the reference image. These results show that the auxiliary mesh-based regularization not only facilitates large geometry transforms, but also stabilizes the overall deformation process. Lastly, we analyze the symmetry loss. As shown in the final row, enforcing symmetry is beneficial when the source mesh and the intended translation inherently possess symmetric structures. This optional constraint allows users to achieve more visually consistent deformations in such scenarios.

4.4. Additional Results

Geometric stylization with text conditions. We further demonstrate that our method can be flexibly combined with additional conditions. First, we use text prompts as control signals, showing that our method enables simultaneous content manipulation and style transfer. In this setting, the source meshes are deformed based on both text instructions and style references. For instance, as illustrated in the first row of Fig. 9, the proposed method transforms the source mesh into a giraffe while transferring the overall style of the given sculpture image.

Localized deformations. We further demonstrate that the proposed method can perform localized geometric stylization. In this setting, users specify one or more regions of the source mesh to be deformed. In practice, we adopt the part segmentation defined by PartField [36], and visualize the selected regions as point sets in Fig. 10. During optimization, we compute the target loss $\mathcal{L}_{\text{tgt}}$ and backpropa-

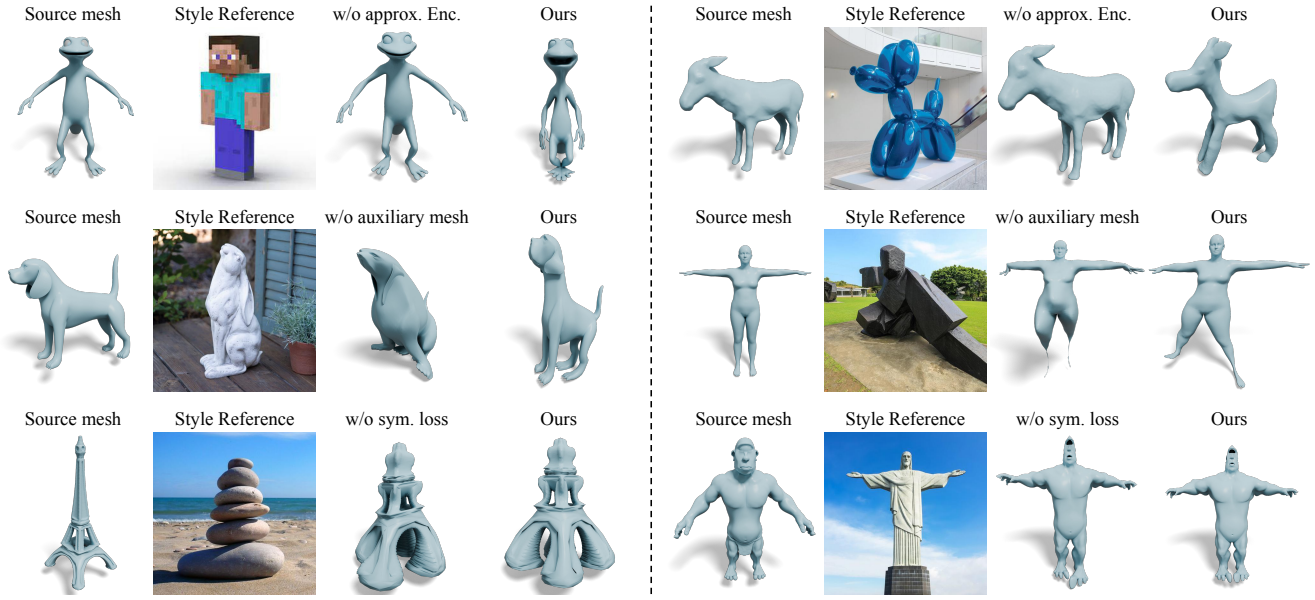

Figure 8. Ablation study results. We analyze the effects of the approximated VAE encoder (1st row), cage-coefficient regularization (2nd) and symmetry loss (3rd). The qualitative results show that each component is essential for producing high-fidelity stylization result.

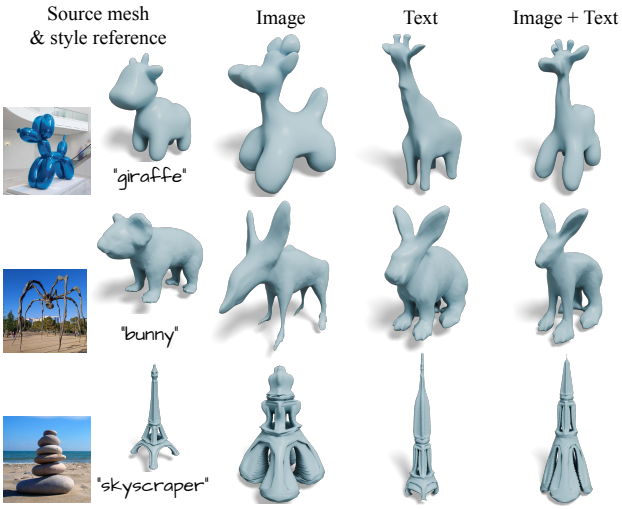

Figure 9. Geometric stylization results with text conditioning. Our method effectively incorporates textual prompts alongside style reference images. The resulting meshes align with the text-described contents while consistently maintaining the geometric style encoded in the style reference.

gate gradients only through the Jacobians corresponding to the selected parts. As shown, the proposed method enables flexible and targeted style transfer, allowing stylization of either the entire mesh or localized areas while preserving the geometry elsewhere.

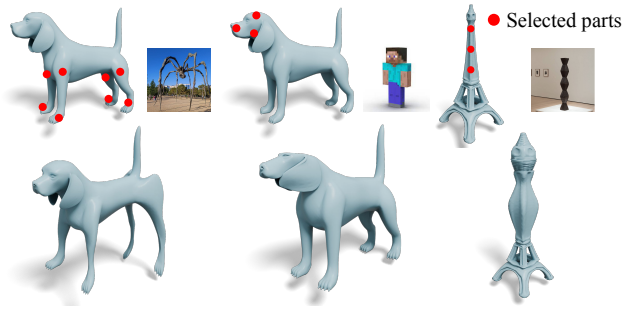

Figure 10. Our approach can transfer geometric style only to user-selected regions while preserving the original shape elsewhere.

5. Conclusion

In this work, we propose a novel framework for geometric stylization of 3D meshes. Unlike prior approaches that primarily focus on texture-oriented stylization or text-based mesh deformation, our method emphasizes geometric style and derives it directly from reference images. We extract style information by training LoRA on a set of style images via DreamBooth, and apply it to the source mesh through a Jacobian-based deformation guided by SDS loss. To further improve the expressiveness of the deformation and computational efficiency, we adopt Stable Diffusion XL with an approximated VAE encoder. We also introduce both cage coefficient regularization and a symmetry loss to enable large-scale geometry manipulations while preserving the inherent structural symmetries. Experimental results demonstrate that our method produces semantically consistent geometric stylizations, outperforming existing baselines.

Acknowledgements

This work was supported by the National Research Foundation of Korea(NRF) grant (No. RS-2026-25485899) and the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant (RS-2025-25442338, AI star Fellowship Support Program(Seoul National Univ.)) funded by the Korea government(MSIT).

References

[1] Noam Aigerman, Kunal Gupta, Vladimir G Kim, Siddhartha Chaudhuri, Jun Saito, and Thibault Groueix. Neural jacobian fields: Learning intrinsic mappings of arbitrary meshes. *SIGGRAPH*, 2022. 2, 3

[2] Jie An, Siyu Huang, Yibing Song, Dejing Dou, Wei Liu, and Jiebo Luo. Artflow: Unbiased image style transfer via reversible neural flows. In *CVPR*, 2021. 2

[3] DeepFloyd Lab at StabilityAI. DeepFloyd IF: a novel state-of-the-art open-source text-to-image model with a high degree of photorealism and language understanding. https://www.deepfloyd.ai/deepfloyd-if, 2023. 2, 4, 6

[4] Tian Qi Chen and Mark Schmidt. Fast patch-based style transfer of arbitrary style. *arXiv:1612.04337*, 2016. 2

[5] Pei-Ze Chiang, Meng-Shiun Tsai, Hung-Yu Tseng, Wei-Sheng Lai, and Wei-Chen Chiu. Stylizing 3d scene via implicit representation and hypernetwork. In *WACV*, 2022. 2

[6] Nam Anh Dinh, Itai Lang, Hyunwoo Kim, Oded Stein, and Rana Hanocka. Geometry in style: 3d stylization via surface normal deformation. In *CVPR*, 2025. 1, 2

[7] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv:2208.01618*, 2022. 5

[8] William Gao, Noam Aigerman, Thibault Groueix, Vova Kim, and Rana Hanocka. Textdeformer: Geometry manipulation using text guidance. In *SIGGRAPH*, 2023. 2, 3, 5, 6

[9] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. 1, 2

[10] Guilherme Gomes Haetinger, Jingwei Tang, Raphael Ortiz, Paul Kanyuk, and Vinicius Azevedo. Controllable neural style transfer for dynamic meshes. In *SIGGRAPH*, 2024. 2

[11] Shuyang Gu, Congliang Chen, Jing Liao, and Lu Yuan. Arbitrary style transfer with deep feature reshuffle. In *CVPR*, 2018. 2

[12] Amir Hertz, Rana Hanocka, Raja Giryes, and Daniel Cohen-Or. Deep geometric texture synthesis. *ACM TOG*, 2020. 2

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

[14] Lukas Höllein, Justin Johnson, and Matthias Nießner. Stylemesh: Style transfer for indoor 3d scene reconstructions. In *CVPR*, 2022. 2

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 2022. 2, 3, 5

[16] Hsin-Ping Huang, Hung-Yu Tseng, Saurabh Saini, Maneesh Singh, and Ming-Hsuan Yang. Learning to stylize novel views. In *ICCV*, 2021. 2

[17] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 2

[18] Hyunyoung Jung, Seonghyeon Nam, Nikolaos Sarafianos, Sungjoo Yoo, Alexander Sorkine-Hornung, and Rakesh Ranjan. Geometry transfer for stylizing radiance fields. In *CVPR*, 2024. 2

[19] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *CVPR*, 2018. 2, 5, 6

[20] Hyunwoo Kim, Itai Lang, Noam Aigerman, Thibault Groueix, Vladimir G Kim, and Rana Hanocka. Meshup: Multi-target mesh deformation via blended score distillation. In *3DV*, 2025. 2, 4, 5, 6

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv:1312.6114*, 2013. 4, 7

[22] Maximilian Kohlbrenner, Ugo Finnendahl, Tobias Djuren, and Marc Alexa. Gauss Stylization: Interactive Artistic Mesh Modeling based on Preferred Surface Normals. *Computer Graphics Forum*, 2021. 1, 2

[23] Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. Style transfer by relaxed optimal transport and self-similarity. In *CVPR*, 2019. 2

[24] Nicholas Kolkin, Michal Kucera, Sylvain Paris, Daniel Sykora, Eli Shechtman, and Greg Shakhnarovich. Neural neighbor style transfer. *arXiv:2203.13215*, 2022. 2

[25] Dmytro Kotovenko, Artsiom Sanakoyeu, Sabine Lang, and Bjorn Ommer. Content and style disentanglement for artistic style transfer. In *ICCV*, 2019. 2

[26] Dmytro Kotovenko, Artsiom Sanakoyeu, Pingchuan Ma, Sabine Lang, and Bjorn Ommer. A content transformation block for image style transfer. In *CVPR*, 2019.

[27] Dmytro Kotovenko, Matthias Wright, Arthur Heimbrecht, and Bjorn Ommer. Rethinking style transfer: From pixels to parameterized brushstrokes. In *CVPR*, 2021. 2

[28] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM TOG*, 2020. 5

[29] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *CVPR*, 2016. 2

[30] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. *NIPS*, 2017. 2

[31] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. *arXiv:1705.01088*, 2017. 2

[32] Hsueh-Ti Derek Liu and Alec Jacobson. Cubic stylization. *ACM TOG*, 2019. 1, 2

[33] Hsueh-Ti Derek Liu and Alec Jacobson. Normal-driven spherical shape analogies. In *Computer Graphics Forum*, 2021. 1, 2

[34] Hsueh-Ti Derek Liu, Michael Tao, and Alec Jacobson. Paparazzi: surface editing by way of multi-view image processing. *ACM TOG*, 2018. 2, 5, 6

[35] Kunhao Liu, Fangneng Zhan, Muyu Xu, Christian Theobalt, Ling Shao, and Shijian Lu. Stylegaussian: Instant 3d style transfer with gaussian splatting. In *SIGGRAPH Asia Technical Communications*. 2024. 2

[36] Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. Partfield: Learning 3d feature fields for part segmentation and beyond. *ICCV*, 2025. 4, 5, 7

[37] Xiao-Chang Liu, Xuan-Yi Li, Ming-Ming Cheng, and Peter Hall. Geometric style transfer. *arXiv:2007.05471*, 2020. 2

[38] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. In *ECCV*, 2018. 2

[39] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *CVPR*, 2023. 4

[40] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *CVPR*, 2022. 1, 5, 6

[41] Fangzhou Mu, Jian Wang, Yicheng Wu, and Yin Li. 3d photo stylization: Learning to generate stylized novel views from a single image. In *CVPR*, 2022. 2

[42] Hong-Wing Pang, Binh-Son Hua, and Sai-Kit Yeung. Locally stylized neural radiance fields. In *ICCV*, 2023. 2

[43] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *CVPR*, 2019. 2

[44] 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. *arXiv:2307.01952*, 2023. 2, 4, 7

[45] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *ICLR*, 2023. 2, 3, 4, 7

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2, 5

[47] Eric Risser, Pierre Wilmot, and Connelly Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses. *arXiv:1701.08893*, 2017. 2

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

[49] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015. 3

[50] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2023. 3, 5

[51] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *CVPR*, 2018. 2

[52] K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[53] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, 2021. 3

[54] Jordan Yaniv, Yael Newman, and Ariel Shamir. The face of art: landmark detection and geometric style in portraits. *ACM TOG*, 2019. 2

[55] Kangxue Yin, Jun Gao, Maria Shugrina, Sameh Khamis, and Sanja Fidler. 3dstylenet: Creating 3d shapes with geometric and texture style variations. In *ICCV*, 2021. 2

[56] Dingxi Zhang, Yu-Jie Yuan, Zhuoxun Chen, Fang-Lue Zhang, Zhenliang He, Shiguang Shan, and Lin Gao. Stylizedgs: Controllable stylization for 3d gaussian splatting. *TPAMI*, 2025. 2

[57] Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields. In *ECCV*, 2022. 2

[58] Yuechen Zhang, Zexin He, Jinbo Xing, Xufeng Yao, and Jiaya Jia. Ref-npr: Reference-based non-photorealistic radiance fields for controllable scene stylization. In *CVPR*, 2023. 2