X-Mesh: Towards Fast and Accurate Text-driven 3D Stylization via Dynamic Textual Guidance

Yiwei Ma[1‡] **Xiaoqing Zhang**[1‡] **Xiaoshuai Sun**[1 *] **Jiayi Ji**[1]
Haowei Wang[1] **Guannan Jiang**[2] **Weilin Zhuang**[2] **Rongrong Ji**[1]

[1]Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University, 361005, P.R. China.
[2]Contemporary Amperex Technology Co., Limited (CATL), Fujian, China

Abstract

Text-driven 3D stylization is a complex and crucial task in the fields of computer vision (CV) and computer graphics (CG), aimed at transforming a bare mesh to fit a target text. Prior methods adopt text-independent multilayer perceptrons (MLPs) to predict the attributes of the target mesh with the supervision of CLIP loss. However, such text-independent architecture lacks textual guidance during predicting attributes, thus leading to unsatisfactory stylization *and* slow convergence. *To address these limitations, we present X-Mesh, an innovative text-driven 3D stylization framework that incorporates a novel Text-guided Dynamic Attention Module (TDAM). The TDAM dynamically integrates the guidance of the target text by utilizing text-relevant spatial and channel-wise attentions during vertex feature extraction, resulting in more accurate attribute prediction and faster convergence speed. Furthermore, existing works lack standard benchmarks and automated metrics for evaluation, often relying on subjective and non-reproducible user studies to assess the quality of stylized 3D assets. To overcome this limitation, we introduce a new standard text-mesh benchmark, namely MIT-30, and two automated metrics, which will enable future research to achieve fair and objective comparisons. Our extensive qualitative and quantitative experiments demonstrate that X-Mesh outperforms previous state-of-the-art methods. Our codes and results are available at our project webpage:* https://xmu-xiaoma666.github.io/Projects/X-Mesh/

1. Introduction

In recent years, 3D asset creation through stylization, *i.e.,* transforming bare meshes to match text prompts [39, 6, 68], images [66, 80], and 3D shapes [77], has received significant attention in the fields of computer vision and graphics [14, 15, 21]. The resulting stylized 3D assets are applied to a range of practical applications, such as gaming, virtual reality, and film. Among the stylization techniques available, text-driven 3D stylization is particularly user-friendly, as text prompts are more readily available than images or 3D shapes. However, creating stylized 3D assets through text input presents a significant challenge due to the significant gap between visual and linguistic information.

The emergence of Contrastive Language-Image Pre-training (CLIP) [47] has made it possible to achieve text-driven 3D stylization. Recently, Text2Mesh [39] and TANGO [6] have made significant contributions in this field by predicting the attributes of each vertex on the mesh with the supervision of CLIP loss. Specifically, Text2Mesh pre-

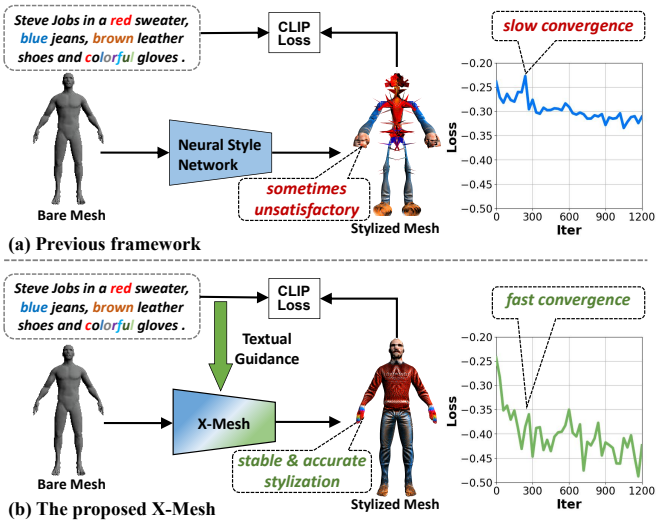


Figure 1. (a) A typical text-driven 3D stylization framework. (b) Our proposed X-Mesh framework. X-Mesh achieves better stylization and faster convergence.

*Corresponding author; ‡Equal contributions.

dicts the color and displacement of each mesh vertex to generate a stylized mesh that aligns with the target text prompt. Similarly, TANGO employs neural networks to forecast diffuse, roughness, specular, and normal maps to create photorealistic 3D meshes following a comparable approach.

Despite achieving impressive results, existing text-driven 3D stylization methods have limitations that hinder their effectiveness and efficiency. One major drawback is their failure to fully consider the semantics of the input text during the prediction of mesh vertex attributes. Current methods only rely on CLIP loss to align the rendered images from the stylized mesh with the text prompt, without any additional textual semantic guidance during predicting vertex attributes. Such approaches lead to several issues, including *unsatisfactory stylization* and *slow convergence*. For instance, as shown in Fig. 1(a), conventional neural style networks do not utilize textual guidance during attribute prediction. As a result, the predicted vertex attributes may not align with the semantic context of the target text prompt, leading to an inconsistent stylized mesh. Moreover, the lack of additional text guidance makes it difficult to rapidly converge to an acceptable result. Typically, previous methods require over 500 iterations (equivalent to over 8 minutes of training) to attain stable stylized outcomes, which is impractical for users.

To address the issues of inconsistency and slow convergence in conventional neural style networks, we propose *X-Mesh*, a framework that leverages textual semantic guidance to predict vertex attributes. As shown in Fig. 1(b), X-Mesh produces high-quality stylized results that are consistent with the input text. Besides, with textual guidance during vertex attribute prediction, X-Mesh usually achieves stable results in just 200 iterations (approximately 3 minutes of training). Our approach relies on a novel *Text-guided Dynamic Attention Module (TDAM)* for text-aware attribute prediction. Fig. 2(b) illustrates how spatial and channel-wise attentions are employed in TDAM to extract text-relevant vertex features. Notably, the parameters of the attention modules are dynamically generated by textual features, which makes the vertex features prompt-aware.

Additionally, the quality evaluation of the stylized results from existing text-driven 3D stylization methods [6, 39] poses a significant challenge. This challenge is mainly reflected in two aspects. Firstly, the lack of a standard benchmark for the text-driven 3D stylization problem presents a challenge in evaluating the effectiveness of existing methods. Without fixed text prompts and meshes, the results obtained from previous methods are incomparable. This in turn hinders progress and the development of more effective solutions. Secondly, the current evaluation of stylized 3D assets relies heavily on user studies, which is a time-consuming and expensive process. Furthermore, this evaluation method is also subject to individual interpretation, which further hinders the reproducibility of results.

To address the aforementioned challenges, we propose a standardized text-mesh benchmark and two automatic evaluation metrics for the fair, objective, and reproducible comparison of text-driven 3D stylization methods. The proposed benchmark, called *Mesh wIth Text (MIT-30)*, contains 30 categories of bare meshes, each of which is annotated with 5 different text prompts for diverse stylization. The proposed two evaluation metrics aims to overcome the limitations of subjective and non-reproducible user studies used in prior work. Specifically, we render 24 images of the stylized 3D mesh from fixed elevation and azimuth angles, and propose two metrics, *Multi-view Expert Score (MES)* and *Iteration for Target Score (ITS)*, to evaluate the stylization quality and convergence speed.

This paper presents two main contributions:

- We propose X-Mesh that incorporates a novel text-guided dynamic attention module (TDAM) to improve the accuracy and convergence speed of 3D stylization.
- We construct a standard benchmark and propose two automatic evaluation metrics, which facilitate objective and reproducible assessments of text-driven 3D stylization techniques, and may aid in advancing this field of research.

2. Related Work

2.1. Text-to-Image Manipulation/Generation

Several previous works have attempted to combine GAN and CLIP to achieve text-to-image generation [4, 52, 75]. Specifically, StyleGAN [26, 27, 25, 59] focuses on the latent space to enable better control over generated images. Building on StyleGAN, StyleCLIP [44] leverages the guidance of CLIP to realize text-to-image generation. DAE-GAN [54] uses a dynamic perception module to comprehensively perceive text information as a development architecture of GAN. Stack-GAN [78, 79] divides the task into two stages, generating basic color and shape constraints of the objects described in the text and then adding more details to produce high-quality images with high resolution. VQGAN [11] improves the performance of visual generation on multiple tasks. MirrorGAN [46] combines the global-to-local attention mechanism with a text-to-image-to-text framework to preserve semantics effectively.

Meanwhile, diffusion models have made significant contributions to image generation. DALL-E [50] and CogView [8, 9, 17] are based on transformer and parallel auto-regressive architectures. GLIDE [12] leverages classifier-free guidance for image generation and restoration after fine-tuning. DALL-E2 [49] generates original and realistic images given a text prompt by encoding image features according to the text features of CLIP and then decod-

ing them via a diffusion model. EDiff-I [2] trains a text-to-image diffusion model for different synthesis stages to achieve high visual quality. Imagen [57] benefits from the semantic encoding ability of the large pre-trained language model T5 [48] and the diffusion model in generating high-fidelity images.

2.2. Text-to-3D Manipulation/Generation

The field of text-to-3D generation has seen significant advancements with the development of text-to-image techniques. Among these techniques, some NeRF-based methods have shown promise, especially when used in combination with CLIP. Some notable examples of such methods include CLIP-NeRF [67], PureCLIPNeRF [28], and DreamFields [22]. Additionally, recent studies have explored the fusion of CLIP with other algorithms, such as ISS [31] with SVR [42], CLIP-Forge [58] using a normalizing flow network [10], and AvatarCLIP [16] leveraging SMLP [34]. Furthermore, the diffusion model [55] has recently demonstrated impressive results in text-to-image generation, leading to its integration into the text-to-3D generation process. Examples of studies that have incorporated the diffusion model into their generation process include DreamFusion [45], Magic3D [30], and Dream3D [73].

Besides, mesh-based stylization is also widely researched due to its wide applicability. Traditionally, the stylization of bare meshes in computer graphics requires professional knowledge. However, recent studies [62, 13] have made strides in the automation of stylizing 3D representations using text prompts. For instance, CLIP-Mesh [40] uses CLIP and loop subdivision [33] to achieve 3D asset generation. While TANGO [6] incorporates reflection knowledge, it is limited in shape manipulation. Text2Mesh [39], on the other hand, predicts both color and displacement of each vertex to achieve stronger stylization. This paper proposes a text-guided dynamic attention module in the vertex attribute prediction phase. This module not only leads to a better stylization effect but also achieves a fast convergence speed.

2.3. Attention Mechanism

Attention mechanism is a widely-used technique in deep learning that has been applied to a variety of tasks, including computer vision [20, 69, 18, 19], natural language processing [35, 61, 65], and multimodal fields [36, 74, 37, 38, 76, 24]. The concept of attention was first introduced in the context of neural machine translation by Bahdanau *et al.* [1], who proposed a model that learns to align the source and target sentences by focusing on different parts of the source sentence at each decoding step. Since then, various attention mechanisms have been proposed to improve the performance of different models. For example, Hu *et al.* [20] proposed channel attention to enhance the image recognition ability of the model. Woo *et al.* [71] leveraged both channel attention and spatial attention to focus on important areas and channels. Ye *et al.* [76] introduced dynamic attention for visual grounding, where different visual features are generated for different referring expressions. Self-attention [65], which is an effective global attention mechanism first proposed for NLP tasks, has been widely used to improve the performance of different models. Wang *et al.* [70] introduced a non-local attention mechanism for video understanding tasks. Liu *et al.* [32] improved self-attention by introducing shifted windows, which enhances the local perception ability of the model. In this paper, we propose a text-guided dynamic attention mechanism for text-driven 3D stylization, which enables the spatial (vertex) and channel information of the input mesh to be dynamically focused based on the target text prompt.

3. Approach

In this section, we first explain the overall architecture of X-Mesh in Sec. 3.1. Then, we provide the details of the proposed Text-guided Dynamic Attention Module in Sec. 3.2.

3.1. Architecture

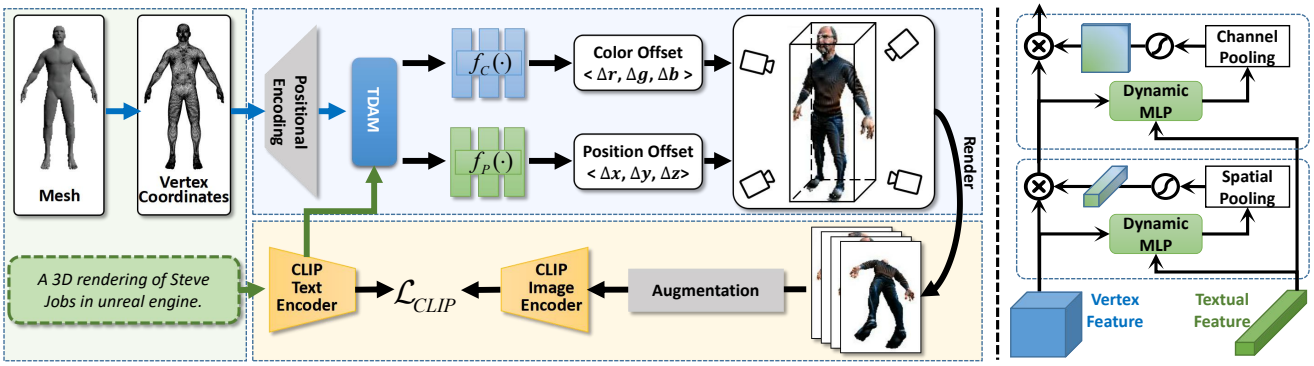
An illustration of the proposed X-Mesh is shown in Fig. 2(a). The goal of X-Mesh is to modify an input mesh to match a given text prompt by predicting its appearance and geometry. Specifically, an input mesh $\mathcal{M}$ is defined as a set of vertices $\mathcal{V} \in \mathbb{R}^{n\times 3}$ and faces $\mathcal{F} \in \{1, \ldots, n\}^{m\times 3}$, which are kept constant during training. Here, n and m denote the number of vertices and faces, respectively. Given an input mesh and a target text prompt, X-Mesh predicts the appearance attribute (*i.e.,* the color offset $\Delta C_p \in \mathbb{R}^3$) and the geometry attribute (*i.e.,* the position offset $\Delta P_p \in \mathbb{R}^3$) of each vertex $p \in \mathcal{V}$, and finally generates a stylized mesh $\mathcal{M}^{\mathcal{S}}$ that conforms to the target text.

We start by initializing the color of each vertex to $(0.5, 0.5, 0.5)$ and normalizing the vertex coordinates to fit within a unit cube. To synthesize high-frequency details, we apply positional encoding using Fourier feature mappings to each vertex. Specifically, given a vertex $p \in \mathcal{V}$ of the mesh, we compute the positional encoding $PE_{(p)}$ as follows:

$$PE_{(p)} = [\cos(2\pi \mathbf{B}p), \sin(2\pi \mathbf{B}p)]^{\mathrm{T}}, \tag{1}$$

where $\mathbf{B} \in \mathbb{R}^{C\times 3}$ is a random Gaussian matrix, and each value in this matrix is randomly sampled from a normal distribution with mean 0 and variance σ^2.

Then, the proposed TDAM takes in the vertex positional encoding feature $PE_{(p)}$, which is dynamically processed under the guidance of the target text prompt. The resulting feature is further passed through two MLP branches, the Color MLP $f_C(\cdot)$ and the Position MLP $f_P(\cdot)$, which generate the color offset ΔC_p and the position offset ΔP_p,



(a) The proposed X-Mesh **(b) Text-Guided Dynamic Attention**

Figure 2. (a) Illustration of the proposed X-Mesh model, which modifies the appearance and geometry of the input mesh according to the text prompt. (b) An overview of TDAM, which aims to process vertex features under the guidance of target text.

respectively. Following [39], the position offset ΔP_p is constrained to a small value, specifically $|\Delta P_p|_2 \leq 0.1$, to prevent excessive deformation. The new color and position attributes of each point are defined as $C'_p = C_p + \Delta C_p$ and $P'_p = P_p + \Delta P_p$, respectively. Here, $C_p \in \mathbb{R}^3$ and $P_p \in \mathbb{R}^3$ represent the RGB color and coordinates of p on the original input mesh, respectively. To enhance geometry, a gray stylized mesh $\mathcal{M}^{\mathcal{S}}_{gray}$ is used, which has the same geometry as $\mathcal{M}^{S}$ but the color of all vertices are set to gray.

We employ an interpolation-based differentiable renderer [5] for $\mathcal{M}^{\mathcal{S}}$ and $\mathcal{M}^{\mathcal{S}}_{gray}$ from n_θ different views. For each view θ, we could obtain two rendered images, *i.e.*, I^{color}_θ for $\mathcal{M}^{\mathcal{S}}$ and I^{gray}_θ for $\mathcal{M}^{\mathcal{S}}_{gray}$. We then apply 2D augmentation $\psi(\cdot)$ to each rendered image, and extract their features using the CLIP visual encoder $E_v(\cdot)$ [47]. We obtain the final feature representation by averaging the features across all views, which can be formulated as follows:

$$\phi_{color} = \frac{1}{n_\theta} \sum_{\theta} E_v(\psi(I^{color}_\theta)), \tag{2}$$

$$\phi_{gray} = \frac{1}{n_\theta} \sum_{\theta} E_v(\psi(I^{gray}_\theta)), \tag{3}$$

To align the rendered images and the target text in CLIP space, we adopt CLIP textual encoder $E_t(\cdot)$ to embed the text prompt. The framework is trained using the CLIP loss, and the training objective can be formulated as:

$$\mathcal{L} = -\text{sim}\left(\phi_{color}, E_t(\mathcal{T})\right) - \text{sim}\left(\phi_{gray}, E_t(\mathcal{T})\right), \tag{4}$$

where $\mathcal{T}$ represents the target text prompt, and $\text{sim}(a, b)$ denotes the cosine similarity between a and b.

3.2. Text-guided Dynamic Attention Module

Previous works on text-driven 3D stylization have been limited by their inability to fully exploit the target text to guide the prediction of vertex attributes, resulting in suboptimal stylization results. To address this limitation, we propose a novel Text-guided Dynamic Attention Module (TDAM) that leverages the target text to guide the attribute prediction process. An overview of our approach is shown in Fig. 2(b), which illustrates how TDAM calculates text-related vertex attention at both channel and spatial levels. Our proposed TDAM is based on a dynamic linear layer, whose parameters are generated dynamically based on the target textual features. We first explain how the dynamic linear layer is implemented and then describe how we design TDAM based on this layer to compute text-aware dynamic channel and spatial attention maps.

Dynamic Linear Layer. Existing text-driven 3D stylization methods use static MLPs to predict the attributes of each vertex on the mesh. However, since the parameters of these MLPs are randomly generated, the target text cannot provide additional guidance during attribute prediction. To address this limitation, we propose a dynamic linear layer, whose parameters are generated based on the target textual feature $\mathbf{F}_t \in \mathbb{R}^{D_t}$. The dynamic linear layer is defined as follows:

$$\mathbf{x}_{out} = \mathbf{x}_{in}\mathbf{W}_t + \mathbf{b}_t, \tag{5}$$

where $\mathbf{x}_{in} \in \mathbb{R}^{D_{in}}$ and $\mathbf{x}_{out} \in \mathbb{R}^{D_{out}}$ represent the input and output vectors of the dynamic linear layer, respectively. The trainable parameters of the dynamic linear layer are denoted as $\mathbf{M}_d \in \mathbb{R}^{(D_{in}+1)\times D_{out}} = \{\mathbf{W}_t \in \mathbb{R}^{D_{in}\times D_{out}}, \mathbf{b}_t \in \mathbb{R}^{D_{out}}\}$, which are generated based on the target textual feature $\mathbf{F}_t$.

A straightforward method to generate dynamic parameters is to use a plain linear layer, defined as follows:

$$\mathbf{M}_d = \mathbf{F}_t\mathbf{W}_m + \mathbf{b}_m, \tag{6}$$

where $\mathbf{W}_m \in \mathbb{R}^{D_t\times(D_{in}+1)*D_{out}}$ and $\mathbf{b}_m \in \mathbb{R}^{(D_{in}+1)*D_{out}}$. However, this method requires a large number of trainable parameters, specifically

$(D_t + 1) * (D_{in} + 1) * D_{out}$, which can result in an unaffordable training cost and overfitting.

Thus, we use matrix decomposition to reduce the number of trainable parameters. Specifically, We decompose $\mathbf{M}_d \in \mathbb{R}^{(D_{in}+1)\times D_{out}}$ into $\mathbf{U} \in \mathbb{R}^{(D_{in}+1)\times K}$ and $\mathbf{V} \in \mathbb{R}^{K\times D_{out}}$, where K is a hyper-parameter that determines the compression ratio. It can be formulated as follows:

$$\mathbf{M}_d = \mathbf{U}\mathbf{V}, \tag{7}$$

where $\mathbf{U}$ is a parameter matrix dynamically generated from $\mathbf{F}_t$ and $\mathbf{V}$ is a static trainable matrix. The formulation of $\mathbf{U}$ is presented as follows:

$$\mathbf{U} = \Phi(\mathbf{F}_t\mathbf{W}_l + \mathbf{b}_l), \tag{8}$$

where $\mathbf{W}_l \in \mathbb{R}^{D_t\times(D_{in}+1)*K}$ and $\mathbf{b}_l \in \mathbb{R}^{(D_{in}+1)*K}$. $\Phi(\cdot)$ is a reshape function that transfers the input from $\mathbb{R}^{(D_{in}+1)*K}$ to $\mathbb{R}^{(D_{in}+1)\times K}$.

Through the matrix decomposition technique, the number of trainable parameters is reduced from $(D_t + 1) \times (D_{in} + 1) * D_{out}$ to $(D_t + 1) \times (D_{in} + 1) * K + K \times D_{out}$, which saves on additional training cost and avoids the risk of over-fitting.

Dynamic Channel and Spatial Attention. As explained earlier, our goal is to obtain vertex features that are sensitive to the target text. To achieve this, we propose a Text-guided Dynamic Attention Module (TDAM) that builds upon the dynamic linear layer and comprises two types of attention mechanisms, *i.e.,* channel attention and spatial attention.

The key element of TDAM is the dynamic MLP, which comprises two dynamic linear layers separated by a ReLU activation function. Inspired by squeeze-and-excitation networks [20], the input and output dimensions of the dynamic MLP are identical, while the hidden dimension is reduced by a factor r.

In TDAM, the objective of channel attention is to activate the channels of the vertex feature that are related to the target text. Specifically, given the vertex feature $\mathbf{F}_v \in \mathbb{R}^{N_v\times D_v}$, where N_v is the number of vertices of the input mesh and D_v is the channel dimension of the input mesh, we first pass it through a dynamic MLP and then aggregate spatial dimensions through average pooling. To obtain the channel-wise attention map, we normalize the values to a range of 0 to 1 using the Sigmoid activation function as follows:

$$\mathbf{A}_{ca} = \sigma\left(\frac{1}{N_v}\sum_{i=1}^{N_v}\eta_1(\mathbf{F}_v)[i,:]\right), \tag{9}$$

where $\mathbf{A}_{ca} \in \mathbb{R}^{1\times D_v}$ denotes the channel-wise attention map, $\sigma(\cdot)$ represents the Sigmoid function, and $\eta_1(\cdot)$ refers to the dynamic MLP. To obtain the channel-activated vertex feature $\mathbf{F}'_v \in \mathbb{R}^{N_v\times D_v}$, we take the element-wise product of $\mathbf{F}_v$ and $\mathbf{A}_{ca}$ as follows:

$$\mathbf{F}'_v = \mathbf{F}_v \otimes \mathbf{A}_{ca}, \tag{10}$$

where $\otimes$ is the element-wise product.

The goal of spatial attention in TDAM is to activate the vertices that are related to the target text. First, we feed the channel-activated vertex feature $\mathbf{F}'$ into another dynamic MLP and aggregate the channel dimensions using the average function. The output is then normalized using the Sigmoid activation function as follows:

$$\mathbf{A}_{sa} = \sigma\left(\frac{1}{D_v}\sum_{j=1}^{D_v}\eta_2(\mathbf{F}'_v)[:,j]\right), \tag{11}$$

where $\mathbf{A}_{sa} \in \mathbb{R}^{N_v\times 1}$, and $\eta_2(\cdot)$ is a dynamic MLP with non-shared parameters with $\eta_1(\cdot)$. Finally, to obtain the spatially-activated vertex feature $\mathbf{F}''_v$, we perform element-wise product between $\mathbf{F}'_v$ and $\mathbf{A}_{sa}$:

$$\mathbf{F}''_v = \mathbf{F}'_v \otimes \mathbf{A}_{sa}. \tag{12}$$

4. Benchmarks and Metrics

Benchmark. In this paper, we construct a text-mesh benchmark to standardize the evaluation process of text-driven 3D stylization. The proposed MIT-30 benchmark includes 30 categories of bare meshes, collected from various public 3D datasets such as COSEG [63], Thingi10K [81], Shapenet [3], Turbo Squid [64], and ModelNet [72]. To ensure a diverse range of stylization, each mesh is annotated with five different text prompts. We found that the prompt template of *'A 3D rendering of $\cdots$ in unreal engine.'* is a good default, so all meshes are annotated with this prompt template if not specified.

Metrics. Some previous works [6, 39] have used user studies to evaluate the perceived quality of stylized 3D assets, which is often subjective and non-reproducible. Other works [23, 29] have employed the metric [43] for text-to-image generation to assess the quality of 3D assets. However, this metric does not account for the continuity of 3D assets, as it only measures the similarity between a single-angle rendered image of the 3D asset and the target text. Given that text-driven 3D stylization aims to produce a 3D asset that conforms to the target text, evaluating rendered images from multiple angles is necessary.

To enable objective and reproducible comparisons, we propose two automatic metrics that are based on multi-angle rendered images of 3D assets. These metrics will replace manual evaluation in user studies, allowing for a reliable evaluation of text-driven 3D stylization methods.

Given a stylized 3D asset, we begin by rendering 24 images $\mathbf{I} = \{I_i\}_{i=1}^{24}$ from 24 fixed views, taking into account both azimuth angle θ_{azi} and elevation angle θ_{ele}. For each

Figure 3. Text-driven 3D stylization results. X-Mesh provides high-quality stylization results for a collection of prompts and meshes.

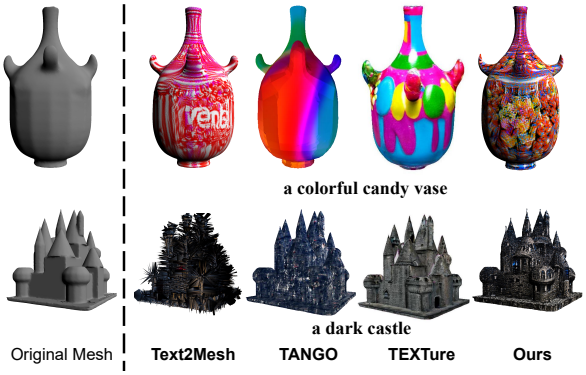


Figure 4. Text-driven 3D stylization results of Text2Mesh [39], TANGO [6], TEXTure [53], and X-Mesh (Ours) given the same mesh and prompt. X-Mesh provides high-quality and realistic stylization results.

3D asset, we establish a standard view where $\theta_{\text{azi}} = 0°$ and $\theta_{\text{ele}} = 0°$. Using this standard view as a basis, we leverage 8 azimuth angles (0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°) and 3 elevation angles (-30°, 0°, 30°) to render 24 rendered images. To address the subjective and non-reproducible nature of user studies, we use an automatic expert model [7] [1] trained on LAION-400M [60] for evaluation. Based on these 24 rendered images and the expert model, we propose two automatic evaluation metrics. Specifically, MES is used to evaluate the extent to which the stylized 3D asset conforms to the target text, and ITS is used to evaluate the convergence rate of the model.

For MES, we first embed the 24 rendered images and the corresponding text prompt into a shared space using the visual and textual encoders of the expert model. Then, we calculate the cosine similarity scores between the rendered images and the corresponding text, and obtain MES by averaging them. The formulation of MES is as follows:

$$\text{MES}(\mathcal{M}^{\mathcal{S}}, \mathcal{T}) = \frac{1}{24} \sum_{i=1}^{24} \text{sim}\Big(E'_v(I_i), E'_t(\mathcal{T})\Big), \qquad (13)$$

where $\mathcal{M}^{\mathcal{S}}$ and $\mathcal{T}$ is the stylized 3D mesh and the corresponding text prompt, respectively. $E'_v(\cdot)$ and $E'_t(\cdot)$ refer to the visual encoder and textual encoder of the expert model.

[1] https://github.com/mlfoundations/open_clip

ITS represents the minimum number of iterations needed to achieve the target MES. For instance, $\text{ITS}_{0.3}(\mathcal{M}^{\mathcal{S}}, \mathcal{T})$ indicates the minimum number of iterations required when $\text{MES}(\mathcal{M}^{\mathcal{S}}, \mathcal{T}) = 0.3$. In our experiment, we set the maximum number of training iterations for each mesh to 1200. If a mesh fails to reach the target MES within 1200 iterations, we set ITS of this sample to 2000. The final MES and ITS are obtained by averaging them across all samples in the benchmark.

5. Experiments

We conducted all experiments using the public PyTorch library on a single RTX 3090 24GB GPU. We trained our proposed X-Mesh using the Adam optimizer with a learning rate of $5e$-4. We set C, n_θ, r, σ, and K to 256, 5, 8, 12, and 30, respectively. $\psi(\cdot)$ includes RandomPerspective and RandomResizedCrop. Our method typically achieves high-quality stylized results in just 3 minutes due to its fast convergence rate. In comparison, previous methods [6, 39] typically take more than 8 minutes to produce stable results on the same GPU.

In Sec. 5.1, we qualitatively compare X-Mesh with state-of-the-art text-driven 3D stylization approaches on MIT-30. In Sec. 5.2, we conduct the ablation study to explore the effectiveness of the proposed module. Finally, We evaluate our method and previous SOTA methods with quantitative metrics in Sec. 5.3.

5.1. Text-driven Stylization

Qualitative Results. Fig. 3 showcases some stylized results generated by X-Mesh for various meshes and driving prompts. The results demonstrate that the stylized meshes are not only faithful to the target text, but also visually plausible. For instance, when given the prompt *"a colourful lamp"*, X-Mesh produces a lamp with vibrant colors that match the prompt while preserving the lamp's shape and structure. Moreover, the generated outputs exhibit a high degree of consistency across different viewpoints. For instance, when given the prompt *"a wooden phoenix"*, the rendered images from different angles exhibit consistent stylization.

Qualitative Comparisons. In this comparison study presented in Fig. 4, we provide evidence of the superiority of our proposed method, X-Mesh, over several existing state-

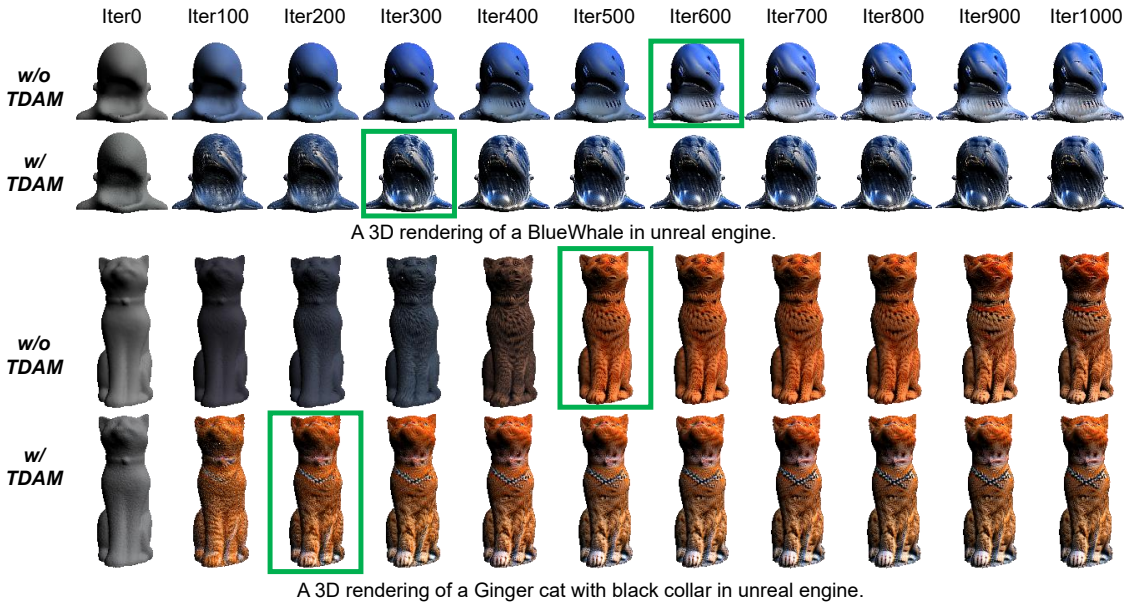


Figure 5. Visualization of text-driven 3D stylization process with and without the proposed TDAM under different iterations. The green box indicates the first iteration to obtain a stable stylization result.

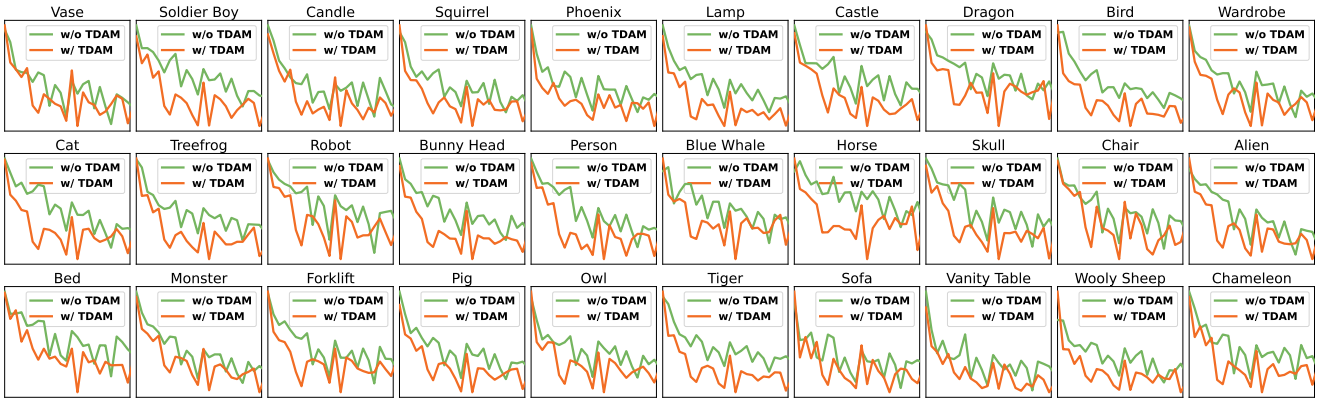


Figure 6. The loss change of each mesh category during training for models with and without TDAM, where the loss values of 5 prompts for each mesh are averaged. The x-axis represents the training iteration, and the y-axis is the loss value. Due to the limitation of page space, we have omitted the contents of the x-axis and y-axis. See *supplementary materials* for a detailed version.

of-the-art approaches for text-driven 3D stylization. We observe that Text2Mesh [39] frequently produces unreasonable deformation, which can be attributed to excessive displacement of vertices. For instance, we provide the example of the *"a dark castle"* in the bottom part of Fig. 4, where Text2Mesh generates several spikes that do not conform to the original structure of the castle.

On the other hand, TANGO [6] and TEXTure [53], which do not displace the vertices of the original mesh, do not suffer from the deformation problem observed in Text2Mesh. However, They still has several shortcomings in terms of stylization quality and text understanding. We demonstrate this by showing the example of the *"a colorful candy vase"* in the top part of Fig. 4, where TANGO and TEXTure simply apply several colors to the vase without taking into account its underlying structure.

In contrast, our proposed method, X-Mesh, overcomes both issues and generates textures that conform to the target text through proper displacement and color prediction for each vertex. We attribute this advantage to the introduction of dynamic guidance of text during vertex attribute prediction. By incorporating dynamic textual guidance, our method is able to generate more accurate results that are in line with the target text.

5.2. Ablation Study

Convergence Speed. Convergence speed is a crucial factor to consider when assessing the effectiveness of text-driven

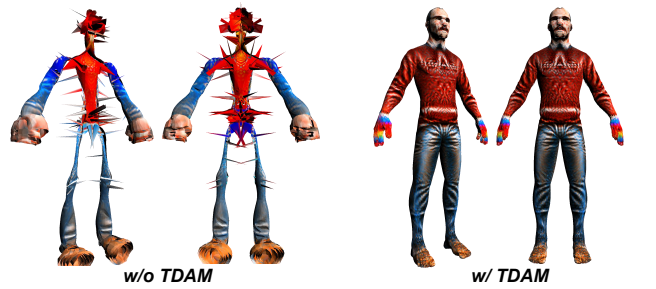


Figure 7. Qualitative comparison of 3D assets generated based on complex prompts without TDAM and with TDAM.

3D stylization. A fast-converging model enables users to obtain the desired 3D asset for a given prompt quickly, while a slow-converging model can be frustrating for users to wait for. The results presented in Fig. 5 demonstrate that our proposed TDAM significantly improves convergence speed, allowing the model to reach an acceptable result in under 100 iterations. In contrast, the model without TDAM requires more than 300 iterations to achieve similar results. Additionally, the TDAM-equipped model reaches stable results in fewer than 300 iterations, while the model without TDAM requires more than 500 iterations. This remarkable improvement in convergence speed can be attributed to the TDAM module, which introduces textual guidance in the attribute prediction process. As a result, the proposed model achieves faster convergence speeds, making it an efficient and effective solution for text-driven 3D stylization.

Moreover, in Fig. 6, we provide the loss curves for 30 categories of meshes in MIT-30. These curves illustrate that the loss value of the model with TDAM decreases faster than that of the model without TDAM during training. The superior performance of the proposed model suggests that TDAM significantly improves the efficiency and effectiveness of text-driven 3D stylization, making it a highly promising tool for 3D content creation. Overall, these findings underscore the importance of TDAM for text-driven 3D stylization, which can significantly improve the convergence speed and reduce the training time for users.

Robustness to Complex Prompts. In this section, we aim to investigate the ability of the proposed X-Mesh to handle complex text prompts with the aid of the TDAM module. To achieve this goal, we conduct several experiments with complex prompts and report our observations as follows:

Firstly, we observe that the model without the TDAM module is highly susceptible to collapse when presented with complex prompts. In particular, as shown in the first line of Fig. 7, the final stylized mesh exhibits numerous spikes and loses its normal geometry when the model lacks the TDAM module. In contrast, the TDAM-equipped model can accurately predict the appropriate color and geometric attributes that match the target text.

Furthermore, we observe that the model without TDAM may fail to capture some critical details in complex prompts. For example, the model without TDAM ignores *"black collar"* in the third line of Fig. 5 and *"colorful gloves"* in the first line of Fig. 7. By contrast, our method can make accurate predictions through comprehensive text understanding.

Overall, our experimental results demonstrate that the TDAM-enhanced model can effectively handle complex text prompts and produce high-quality stylized 3D meshes.

Table 1. Qualitative comparison of state-of-the-art methods for text-driven 3D stylization. Note that a higher MES and a lower $ITS_{0.22}$ is preferable in this table.

	MES ↑	$ITS_{0.22}$ ↓
TANGO [6]	23.21	795.47
Text2Mesh [39]	28.85	173.27
X-Mesh	**29.26**	**88.53**

5.3. Quantitative Comparison

In previous works, user studies are used to evaluate stylization results. However, this evaluation approach has limitations, as it is subjective and non-reproducible. To overcome these limitations, we propose two automatic evaluation metrics, MES and ITS, which respectively measure the quality of the stylized assets and the convergence speed of stylization models. As presented in Tab. 1, our proposed X-Mesh outperforms previous methods. Specifically, X-Mesh achieves a 0.41 absolute improvement in MES on MIT-30, indicating that our method produces better stylization quality than previous works. Moreover, X-Mesh obtains the lowest $ITS_{0.22}$, highlighting that our method converges faster than previous methods. The superior performance of our proposed method demonstrated in both MES and ITS, metrics further validates the effectiveness and superiority of X-Mesh over previous methods, and supports its potential for practical applications.

5.4. Limitations and Future Work

While the proposed X-Mesh has shown promise in generating high-quality 3D assets from barely meshes, it is important to acknowledge its limitations. One such limitation is the possibility of the stylized 3D assets having text or text-like symbols on their surface in some cases, as demonstrated in Fig. 8. This is an issue that has also been observed in previous methods [6, 39] that utilize CLIP similarity as a guiding metric for optimization.

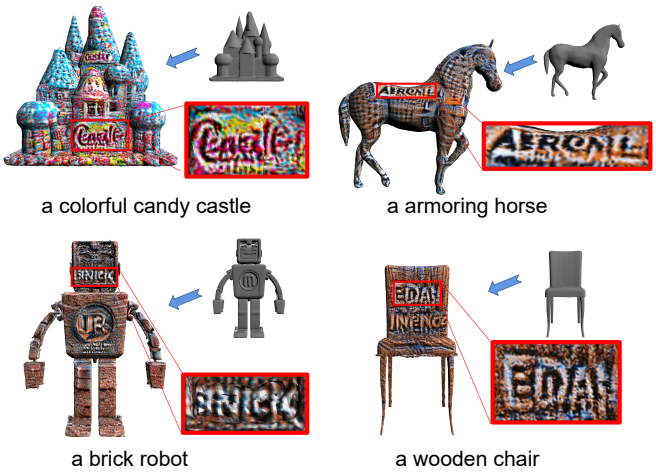


Figure 8. Limitations of the proposed X-Mesh.

To address this limitation, alternative approaches that avoid the use of CLIP similarity could be explored in future research. One such approach that shows promise is the use of diffusion models [51, 41, 56]. These models use iterative denoising to synthesize high-quality images, and have the potential to generate 3D assets from text prompts without relying on CLIP similarity as a guiding metric. This could be a valuable avenue for future research to further enhance the performance and overcome the limitations of current text-driven 3D stylization methods. By exploring alternative methods and techniques, future research could further improve the performance and address existing limitations to advance the state-of-the-art in this area.

6. Conclusion

In this paper, we propose X-Mesh, a novel text-driven 3D stylization framework that leverages a text-guided dynamic attention module to predict vertex attributes, resulting in accurate stylization and fast convergence. Furthermore, we construct a text-mesh benchmark and introduce two automatic metrics to facilitate an objective and reproducible evaluation of this field. Extensive experiments demonstrate that X-Mesh outperforms existing state-of-the-art methods both qualitatively and quantitatively.

Acknowledgement

This work was supported by National Key R&D Program of China (No.2022ZD0118201), the National Science Fund for Distinguished Young Scholars (No.62025603), the National Natural Science Foundation of China (No. U21B2037, No. U22B2051, No. 62176222, No. 62176223, No. 62176226, No. 62072386, No. 62072387, No. 62072389, No. 62002305 and No. 62272401), China Postdoctoral Science Foundation (No.2023M732948), and the Natural Science Foundation of Fujian Province of China (No.2021J01002, No.2022J06001).

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 3

[2] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. 2022. 3

[3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 5

[4] Hila Chefer, Sagie Benaim, Roni Paiss, and Lior Wolf. Image-based clip-guided essence transfer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIII*, pages 695–711. Springer, 2022. 2

[5] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. *Advances in neural information processing systems*, 32, 2019. 4

[6] Yongwei Chen, Rui Chen, Jiabao Lei, Yabin Zhang, and Kui Jia. Tango: Text-driven photorealistic and robust 3d stylization via lighting decomposition. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2022. 1, 2, 3, 5, 6, 7, 8

[7] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. *arXiv preprint arXiv:2212.07143*, 2022. 6

[8] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021. 2

[9] Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. Cogview2: Faster and better text-to-image generation via hierarchical transformers. *arXiv preprint arXiv:2204.14217*, 2022. 2

[10] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016. 3

[11] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. 2020. 2

[12] Richard A Friesner, Jay L Banks, Robert B Murphy, Thomas A Halgren, Jasna J Klicic, Daniel T Mainz, Matthew P Repasky, Eric H Knoll, Mee Shelley, Jason K Perry, et al. Glide: a new approach for rapid, accurate docking and scoring. 1. method and assessment of docking ac-

curacy. *Journal of medicinal chemistry*, 47(7):1739–1749, 2004. 2

[13] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances In Neural Information Processing Systems*, 35:31841–31854, 2022. 3

[14] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 1

[15] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David Salesin. Image analogies. *international conference on computer graphics and interactive techniques*, 2001. 1

[16] Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *arXiv preprint arXiv:2205.08535*, 2022. 3

[17] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022. 2

[18] Jie Hu, Liujuan Cao, Yao Lu, ShengChuan Zhang, Yan Wang, Ke Li, Feiyue Huang, Ling Shao, and Rongrong Ji. Istr: End-to-end instance segmentation with transformers. *arXiv preprint arXiv:2105.00637*, 2021. 3

[19] Jie Hu, Linyan Huang, Tianhe Ren, Shengchuan Zhang, Rongrong Ji, and Liujuan Cao. You only segment once: Towards real-time panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17819–17829, 2023. 3

[20] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 3, 5

[21] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017. 1

[22] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 867–876, 2022. 3

[23] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 867–876, 2022. 5

[24] Jiayi Ji, Yiwei Ma, Xiaoshuai Sun, Yiyi Zhou, Yongjian Wu, and Rongrong Ji. Knowing what to learn: a metric-oriented focal mechanism for image captioning. *IEEE Transactions on Image Processing*, 31:4321–4335, 2022. 3

[25] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34:852–863, 2021. 2

[26] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2

[27] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 2

[28] Han-Hung Lee and Angel X Chang. Understanding pure clip guidance for voxel grid nerf models. *arXiv preprint arXiv:2209.15172*, 2022. 3

[29] Han-Hung Lee and Angel X Chang. Understanding pure clip guidance for voxel grid nerf models. *arXiv preprint arXiv:2209.15172*, 2022. 5

[30] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440*, 2022. 3

[31] Zhengzhe Liu, Peng Dai, Ruihui Li, Xiaojuan Qi, and Chi-Wing Fu. Iss: Image as stetting stone for text-guided 3d shape generation. *arXiv preprint arXiv:2209.04145*, 2022. 3

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 3

[33] Charles Loop. Smooth subdivision surfaces based on triangles. 1987. 3

[34] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 3

[35] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015. 3

[36] Yiwei Ma, Jiayi Ji, Xiaoshuai Sun, Yiyi Zhou, and Rongrong Ji. Towards local visual modeling for image captioning. *Pattern Recognition*, 138:109420, 2023. 3

[37] Yiwei Ma, Jiayi Ji, Xiaoshuai Sun, Yiyi Zhou, Yongjian Wu, Feiyue Huang, and Rongrong Ji. Knowing what it is: semantic-enhanced dual attention transformer. *IEEE Transactions on Multimedia*, 2022. 3

[38] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 638–647, 2022. 3

[39] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13492–13502, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[40] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. Clip-mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–8, 2022. 3

[41] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 9

[42] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020. 3

[43] Dong Huk Park, Samaneh Azadi, Xihui Liu, Trevor Darrell, and Anna Rohrbach. Benchmark for compositional text-to-image synthesis. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. 5

[44] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021. 2

[45] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 3

[46] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. Mirrorgan: Learning text-to-image generation by redescription. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1505–1514, 2019. 2

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *international conference on machine learning*, 2021. 1, 4

[48] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 3

[49] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. 2023. 2

[50] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 2

[51] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 9

[52] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR, 2016. 2

[53] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. *arXiv preprint arXiv:2302.01721*, 2023. 6, 7

[54] Shulan Ruan, Yong Zhang, Kun Zhang, Yanbo Fan, Fan Tang, Qi Liu, and Enhong Chen. Dae-gan: Dynamic aspect-aware gan for text-to-image synthesis. *arXiv: Computer Vision and Pattern Recognition*, 2021. 2

[55] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 3

[56] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 9

[57] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar, Seyed Ghasemipour, Burcu Karagol, S Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. 2023. 3

[58] Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshan. Clip-forge: Towards zero-shot text-to-shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18603–18613, 2022. 3

[59] Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. *arXiv preprint arXiv:2301.09515*, 2023. 2

[60] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 6

[61] Lifeng Shang, Zhengdong Lu, and Hang Li. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*, 2015. 3

[62] Yawar Siddiqui, Justus Thies, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Texturify: Generating textures on 3d shape surfaces. In *European Conference on Computer Vision*, pages 72–88. Springer, 2022. 3

[63] Oana Sidi, Oliver van Kaick, Yanir Kleiman, Hao Zhang, and Daniel Cohen-Or. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–10, 2011. 5

[64] TurboSquid. Turbosquid 3d model repository, 2021. https://www.turbosquid.com/. 5

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[66] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3835–3844, 2022. 1

[67] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3835–3844, 2022. 3

[68] Can Wang, Ruixiang Jiang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Nerf-art: Text-driven neural radiance fields stylization. *arXiv preprint arXiv:2212.08070*, 2022. 1

[69] Haowei Wang, Jiayi Ji, Yiyi Zhou, Yongjian Wu, and Xiaoshuai Sun. Towards real-time panoptic narrative grounding by an end-to-end grounding network. *arXiv preprint arXiv:2301.03160*, 2023. 3

[70] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 3

[71] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 3

[72] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 5

[73] Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. *arXiv preprint arXiv:2212.14704*, 2022. 3

[74] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015. 3

[75] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018. 2

[76] Jiabo Ye, Junfeng Tian, Ming Yan, Xiaoshan Yang, Xuwu Wang, Ji Zhang, Liang He, and Xin Lin. Shifting more attention to visual backbone: Query-modulated refinement networks for end-to-end visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15502–15512, 2022. 3

[77] Kangxue Yin, Jun Gao, Maria Shugrina, Sameh Khamis, and Sanja Fidler. 3dstylenet: Creating 3d shapes with geometric and texture style variations. *international conference on computer vision*, 2023. 1

[78] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017. 2

[79] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1947–1962, 2018. 2

[80] Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields. In *European Conference on Computer Vision*, pages 717–733. Springer, 2022. 1

[81] Qingnan Zhou and Alec Jacobson. Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797*, 2016. 5