IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 March 2024, accepted 17 April 2024, date of publication 23 April 2024, date of current version 1 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3392568

RESEARCH ARTICLE

Fast 3D Stylized Gaussian Portrait Generation From a Single Image With Style Aligned Sampling Loss

SHANGMING JIANG, XINYOU YU, WEIJUN GUO, AND JUNLING HUANG
Anhui Construction Engineering Group Company Ltd., Hefei, Anhui 230031, China
Corresponding author: Junling Huang (huangjl879@sina.com)

ABSTRACT Creating stylized 3D avatars and portraits from just a single image input is an emerging challenge in augmented and virtual reality. While prior work has explored 2D stylization or 3D avatar generation, achieving high-fidelity 3D stylized portraits with text control remains an open problem. In this paper, we present an efficient approach for generating high-quality 3D stylized portraits directly from a single input image. Our core representations are based on 3D Gaussian Splatting for efficient rendering, along with a surface-guided splitting and cloning strategy to reduce noise. To achieve high-fidelity stylized results, we introduce a Stylized Generation Module with a Style-Aligned Sampling Loss that injects the input image's identity information into the diffusion model while stabilizing the stylization process. Furthermore, we incorporate a multi-view diffusion model to enforce 3D consistency by generating multiple viewpoints. Extensive experimentation demonstrates that our approach outperforms existing methods in terms of stylization quality, 3D consistency, and user preference ratings. Our framework enables casual users to easily generate stylized 3D portraits through simple image or text inputs, facilitating engaging experiences in AR/VR applications.

INDEX TERMS Virtual reality, 3D generation, diffusion model, Gaussian splatting.

I. INTRODUCTION

Creating photorealistic 3D avatars is a fundamental challenge in augmented and virtual reality applications, as they enable immersive and realistic representations for remote interactions. Achieving realistic 3D portraits is particularly crucial, as they precisely capture intricate details like hair and expressions, fostering a sense of realism and vividness during virtual interactions. However, in certain scenarios, users may prefer stylized representations of themselves over photorealistic portraits. This stylization encompasses a diverse range of aesthetics, from cartoonish styles such as Disney animated films to the distinct 2D look of Japanese manga or even portraying themselves as different entities like animals. Achieving such stylization capabilities will facilitate engaging and interesting experiences within AR/VR environments. Moreover, these techniques should be accessible to casual users, enabling stylized customization through simple inputs like a single image or text prompts.

In recent years, there have been some prior efforts that have explored the topic of stylizing portraits. However, these methods are often limited to 2D stylization via Generative Adversarial Networks (GANs) [1], [2], [3] or struggle to achieve 3D stylization effects with a simple text prompt. Recent work like StyleAvatar [4] uses CLIP [5] for more flexible stylization by text input. However, they still rely on a pre-reconstruction 3D portrait rather than directly generating a 3D stylized portrait from a single image input.

With the emergence and advancement of large generative diffusion models, some methods have attempted to leverage diffusion models for 3D generation using only a single image or text input [6], [7], [8]. These approaches are based on a learnable 3D representation such as NeRF [9], where images rendered from multiple viewpoints are fed into the diffusion model to establish the SDS loss [7], thereby distilling the 3D structure to achieve 3D generation.

The associate editor coordinating the review of this manuscript and approving it for publication was Andrea Bottino.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

Among these, Zero-1-to-3 [6] learned a single-image diffusion model for generating other viewpoints, successfully achieving 3D generation from a single image. However, the current quality of 3D models generated by such methods is generally suboptimal, exhibiting color oversaturation, slow generation speed, and unstable stylization.

In this paper, we propose an efficient and high-fidelity 3D stylized generation approach capable of generating a high-quality 3D stylized portrait directly from a single input image. Our method first adopts a 3D Gaussian Splatting representation, which is an efficient 3D representation capable of real-time and high-resolution rendering. However, it is unstable during the generation process and prone to noise. To address this, we propose a surface-guided splitting and cloning strategy that distributes the generated point cloud more uniformly across the geometric surface of the 3D portrait, significantly reducing noise and improving the final generation quality. Subsequently, to achieve better stylization effects, we introduce a Stylized Generation Module with Style-Aligned Sampling Loss. The core of this module lies in its ability to inject the identity information from the given single input image into the diffusion model while also stably maintaining stylization and suppressing oversaturated style generation, ultimately enhancing the generation quality. Additionally, to ensure better 3D consistency, we incorporate a multi-view diffusion model that simultaneously generates multiple viewpoints to produce a 3D loss, improving the quality of our 3D portrait generation. In the experimental section, we conduct quantitative and qualitative experiments, demonstrating that our method outperforms previous approaches. We also conduct a user study, where users express a strong preference for the generation quality achieved by our method over other methods. In summary, our contributions are:

- Introduce Gaussian Splatting for fast 3D portrait generation and a surface-guided splitting and cloning strategy to reduce noise and improve the generation quality of 3D portraits from a single image.
- A Stylized Generation Module with Style Aligned Sampling Loss that injects identity information while stabilizing stylization and preventing oversaturation for high-quality stylized 3D portrait generation.
- Utilization of the multi-view diffusion model that generates multiple viewpoints simultaneously to produce a 3D consistency loss, enhancing the quality of the 3D portrait generation.

II. RELATED WORK

In this section, we will review methods for stylization in both 2D and 3D domains. we will also discuss Gaussian Splatting and related work for 3D generation, with a particular focus on image-to-3D generation approaches.

A. 2D AND 3D STYLIZATION

The concept of style transfer and pioneering work in this area was introduced by Gaty [10]. The goal of style transfer is to modify an input image to make it conform to a specific style. This concept has recently been extended to 3D [11], where 3D style transfer involves modifying a 3D model such that its rendered images exhibit a particular style while maintaining multi-view consistency. The style transfer concept has profoundly influenced generative adversarial network (GAN) [1]. A representative work in this field, StyleGAN [12], combines the concept of style transfer with GANs, implicitly encoding styles and enabling interpolation between different styles to generate stylized images. This idea from StyleGAN has inspired many subsequent works [2], [3], [13], [14].

Compared to 2D style transfer, 3D editing and stylization are rapidly developing [4], [15], [16], [17], [18]. Traditional 3D style transfer was generally based on traditional 3D representations like meshes and point clouds [19], [20]. In recent years, with advancements in neural 3D representations [9], [21], many methods have utilized NeRF for 3D style transfer. A representative work, SNeRF [22], achieves 3D style transfer by updating and stylizing each viewpoint image. Instruct-NeRF2NeRF [15] takes this further by combining NeRF with diffusion models [23], enabling text-guided editing and stylization of 3D scenes. In the portrait domain, AvatarCLIP [24] uses CLIP [5] to align the rendered images of 3D representations with specific styles, enabling 3D stylized portrait generation. StyleAvatar [4] also employs CLIP alignment loss to stylize dynamic head avatars. Control4D [17] goes even further by combining GANs and diffusion models. However, these methods either require pre-reconstructed 3D scenes or can only perform style transfer based on text prompts, failing to leverage the information from a given single image for 3D stylized portrait generation.

B. 3D GENERATION

In recent years, 3D generation has witnessed rapid development and progress [6], [7], [8], [25], [26], [27], [28], [29]. 3D generation can be divided into two categories: direct generation methods and lifting from 2D to 3D methods. Direct generation typically requires a large number of 3D models as a training dataset and demands enormous computational resources for training [30]. The lifting 2D to 3D approach was introduced by DreamFusion [7]. The core idea behind this approach is to utilize a neural 3D representation such as NeRF [9], render it from various viewpoints, and then align the rendered images with images generated by a large generative model such as Stable Diffusion [23]. DreamFusion introduced the SDS loss, which can backpropagate the gradients to update the neural 3D representation continuously and achieve 3D generation. While these methods are time-consuming, they ultimately produce higher-quality 3D generation results [25], [27], [31]. This line of research has also spawned numerous methods for generating 3D content from a single image. Zero-1-to-3 [6] introduced an image-to-3D diffusion model that, given a single image,

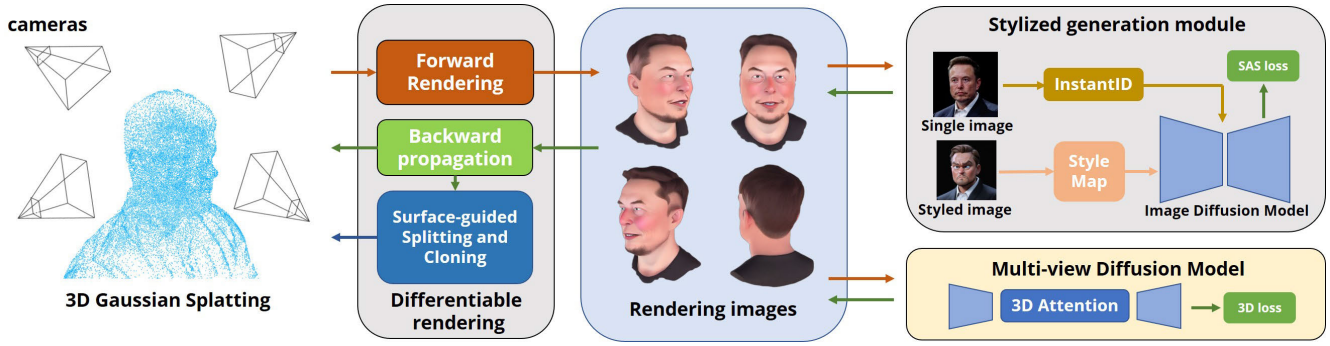

FIGURE 1. The 3D generation pipeline of our method.

generates images from other viewpoints. Subsequently, Magic123 [32] and DreamCraft3D [26] expanded in this direction, further enhancing the quality of their generated outputs. However, these methods can only generate content based on the provided single image and cannot produce 3D stylized portraits. Additionally, these approaches heavily rely on the given image being a front-facing view, and their generation results are not robust when provided with other viewpoints, such as a side view.

C. GAUSSIAN SPLATTING

Gaussian Splatting is an emerging 3D representation that leverages Gaussian point clouds for rasterized rendering [33]. Compared to neural representations like NeRF, Gaussian Splatting does not require dense ray sampling, significantly improving rendering speed and reducing the memory requirements. Gaussian Splatting has spawned many improved methods and applications [34], [35], [36], [37], [38]. In the portrait domain, Animatable Gaussian [38] introduced the combination of human 3D representations with Gaussian Splatting, enhancing the accuracy and realism of 3D avatar reconstruction. In the field of 3D generation, DreamGaussian [35] proposed the use of Gaussian Splatting and diffusion models for fast and efficient 3D generation. Although Gaussian Splatting is highly efficient, its robustness is relatively limited. In this paper, we introduce a novel splitting and cloning mechanism to improve the robustness of 3D portrait generation and reduce noise.

III. METHOD

Our method's entire pipeline is illustrated in Fig. 1, comprising three main modules: the 3D Gaussian splatting representation and rendering module, the stylized generation module with style-aligned sampling loss, and the multi-view generation module.

- First, our method leverages an efficient 3D Gaussian splatting representation to iteratively render images from various viewpoints.
- Subsequently, we feed the input single image and Gaussian Splatting rendered images into the stylized generation module. The stylized generation module can extract the identity information from the input single image and generate the style aligned sampling loss.
- We also feed the rendered images the multi-view diffusion model to generate 3D loss.
- The SAS loss and 3D loss are backpropagated through gradients to the 3D Gaussian Splatting representation, gradually achieving stylized 3D portrait generation.

A. GAUSSIAN SPLATTING REPRESENTATION

The Gaussian splatting representation we employ is a point cloud representation consisting of a series of discrete Gaussian points $p_i, i \in \{1, 2 \ldots n\}$. Each point encapsulates attributes describing properties of 3D objects or scenes. Unlike traditional point cloud representation, each point in the Gaussian splatting representation possesses additional attributes beyond position $\mathbf{x} \in R^3$, color $\mathbf{c} \in R^3$, and opacity $\alpha \in R^3$. It also includes the rotation quaternion $\mathbf{r} \in R^4$ and the scaling factor $\mathbf{s} \in R^3$. Furthermore, the core of Gaussian splatting lies in its automatic cloning and splitting of each point, gradually increasing the number of points to describe high-quality detail. In the subsequent sections, we will introduce Gaussian Splatting rendering for 3D portrait generation and propose a surface-guided splitting and cloning strategy for Gaussian Splatting, aimed at reducing noise during the 3D generation process.

1) GAUSSIAN SPLATTING RENDERING

As mentioned above, the Gaussian Splatting representation we adopt consists of an N-point Gaussian point cloud, where each point in the point cloud is characterized by five attributes: position $\mathbf{x}$, opacity α, color $\mathbf{c}$, scaling factor $\mathbf{s}$, and rotation $\mathbf{r}$. To render the Gaussian point cloud, each point p_i is first projected onto the corresponding viewpoint by project matrix P, and the covariance matrix Σ of each Gaussian point can be calculated based on its rotation $\mathbf{r}$ and scaling factor $\mathbf{s}$:

$$\Sigma = RSS^T R^T, \tag{1}$$

where R is the rotation matrix calculated by rotation quaternion r, S is the 3×3 scaling matrix where the diagonal elements are equal to the scaling factor s. Subsequently, the

rendering property $\mathcal{X}$ such as color c of each point can be transformed into Gaussian distribution:

$$G(\mathcal{X}) = \exp^{-\frac{1}{2}(\mathcal{X}-x)^T \Sigma^{-1}(\mathcal{X}-x)}. \tag{2}$$

Then we rasterize Gaussian points onto the rendering plane and obtain the color for each pixel by integrating the densities and multiplying them by their associated colors:

$$\mathbf{z}_r = \sum_{i \in N} \mathbf{c}_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j). \tag{3}$$

2) SURFACE-GUIDED CLONING AND SPLITTING

The splitting and cloning of Gaussian point clouds are fundamental operations within the original Gaussian Splatting representation. It could automatically detect points requiring splitting and cloning during the training process. This operation gradually increases the Gaussian points, which enhances the resolution of the 3D representations and improves the quality of the 3D generation.

Specifically, the original detection strategy is implemented through thresholding based on gradients backpropagation. It computes the gradients for each point in the direction of 2D screen space:

$$E = \sqrt{(\frac{\partial L}{\partial x_{2D}})^2 + (\frac{\partial L}{\partial y_{2D}})^2}, \tag{4}$$

where L is the loss function. If E exceeds the threshold β_E, which indicates significant variation in its nearby region, we require additional points through splitting or cloning operations to describe this region of the 3D model:

$$\mathbf{x}_{new} = \mathbf{x}_{old} + \mathbf{d}, \mathbf{d} \sim \mathcal{N}(0, \mathbf{s}_{old}) \tag{5}$$

However, such cloning or splitting processes can generate considerable noise. To address this, we propose a novel surface-guided cloning and splitting strategy. Since we generate 3D portrait models, the Gaussian points should be distributed on a relatively smooth surface. Thus, newly splitting or cloning points should be positioned near the surface rather than randomly offset:

$$\mathbf{x}_{new} = \mathbf{x}_{old} + (\mathbf{d} - \mathbf{d} \cdot \mathbf{n}), \mathbf{d} \sim \mathcal{N}(0, \mathbf{s}_{old}), \tag{6}$$

where $\mathbf{n}$ is the normal direction calculated by PCA decomposition of its near Gaussian points:

$$\begin{aligned} \Sigma_n &= cov(x' - x), \\ \mathbf{n} &= PCA_{min}(\Sigma_n), \end{aligned} \tag{7}$$

where x' is the near points of the center point x and their distance is less than 0.01. Through the surface-guided splitting and cloning strategies, we significantly enhance the robustness of the 3D head portrait generation process, reduce the noise of the Gaussian point cloud, and consequently improve the overall quality of the generated models.

B. STYLE ALIGNED SAMPLING LOSS

To achieve 3D stylized portrait generation, we leverage diffusion models and propose SAS (Style Aligned Sampling). Our approach is inspired by DreamFusion. Specifically, to generate stylized 3D portraits, we use text y to describe the desired style, and then iteratively train the 3D Gaussian Splatting representation to ensure its rendering results $\mathbf{z}$ aligned with the given textual style description. During this generation process, we render observations from multiple random viewpoints $\mathbf{C}$ and feed these rendered images into the diffusion model. The diffusion model then computes the distance between these images and the specified style. If we use SDS (Score Distillation Sampling), the loss function will be:

$$\nabla \mathcal{L}_{SDS}(\theta) \approx \mathbb{E}_{\mathbf{C},t,\epsilon} \left[\omega(t)(\epsilon_p(\mathbf{z}_t, t, y) - \epsilon) \frac{\partial g(\theta, \mathbf{C})}{\partial \theta} \right]. \tag{8}$$

However, this loss function suffers from two main issues. First, it cannot utilize the given input image $\mathbf{I}$, and the generated results can only be controlled by text y. Second, the SDS loss gradually oversaturates the generated style, leading to unnatural results. To address these issues, we propose SAS (Style Aligned Sampling) loss, a loss function that preserves the identity and consistency with the given input image $\mathbf{I}$ while ensuring stable and natural stylization.

1) IDENTITY-PRESERVING SAMPLING

we employ InstantID [39] to ensure identity consistency with the input image $\mathbf{I}$. InstantID maintains identity consistency of portrait by incorporating the identity information into the diffusion model in two ways. First, InstantID encodes the image and injects the encoded identity $\mathbf{I}_e$ into the cross-attention layers of the diffusion model:

$$Z_{new} = Att(Q, K, V) + \lambda \cdot Att(Q, K^{\mathbf{I}_e}, V^{\mathbf{I}_e}) \tag{9}$$

Then, InstantID extracts the landmark information $\mathbf{I}_l$ from the image and incorporates it into the diffusion model via ControlNet $\mathcal{F}_c$:

$$Y_{new} = \mathcal{F}(\mathbf{z}_t, t, y) + \mathcal{F}_c(\mathbf{I}_l). \tag{10}$$

Finally, we introduce the identity information of the input image into the diffusion model, and the loss function could be formulated as:

$$\nabla \mathcal{L}_{SDS}(\theta) \approx \mathbb{E}_{\mathbf{C},t,\epsilon} \left[\omega(t)(\epsilon_p(\mathbf{z}_t, t, y, \mathbf{I}_e, \mathbf{I}_l) - \epsilon) \frac{\partial g(\theta, \mathbf{C})}{\partial \theta} \right]. \tag{11}$$

2) STYLE-PRESERVING SAMPLING

To ensure stable stylization during training without oversaturation, we first use the diffusion model to randomly generate a set of images $\mathbf{I}_s$ that conform to the corresponding style. These images are natural and do not exhibit oversaturation effects. We then calculate the mean μ_s and

covariance matrix Σ_s of all the images in the latent space. Subsequently, when we render a new image and compute the loss by diffusion model, we first obtain the generated image:

$$\mathbf{z}' = \alpha_t \mathbf{z} + \beta_t(\epsilon_p(\mathbf{z}_t, t, y, \mathbf{I}_e, \mathbf{I}_l) - \epsilon). \tag{12}$$

Then we map it to the natural and non-oversaturated distribution using Eigenvalue Decomposition:

$$\begin{aligned} \Sigma_s = &\quad T_s V_s T_s^{-1}, \Sigma_g = T_g V_g T_g^{-1}, \\ \mathbf{z}'_m = &\quad T_s V_s^{-\frac{1}{2}} V_g^{-\frac{1}{2}} T_g^{-1}(z' - \mu_g) + \mu_g, \end{aligned} \tag{13}$$

where μ_g and Σ_g are the mean and covariance matrix of generated image $\mathbf{I}_g$. The formulation of our proposed SAS loss is:

$$\nabla \mathcal{L}_{SAS}(\theta) \approx \mathbb{E}_{\mathbf{C},t,\epsilon}\left[\omega(t)(\mathbf{z}'_m - \mathbf{z})\frac{\partial g(\theta, \mathbf{C})}{\partial \theta}\right]. \tag{14}$$

By employing SAS loss, we can maintain a stable style without introducing oversaturation artifacts.

C. MULTI-VIEW DIFFUSION MODEL

To mitigate the issue of multi-face and multi-head during the 3D portrait generation process from a single input image, we introduce the multi-view diffusion model MVDream as an additional 3D supervision. The multi-view diffusion model can simultaneously generate images from multiple viewpoints given the corresponding camera parameters $\mathbf{C}$. It establishes correlations across different views through 3D attention, enabling more consistent generation across views, thereby avoiding the multi-head or multi-face problem.

Specifically, in our 3D generation process, we incorporate the multi-view diffusion model to construct the following 3D loss:

$$\nabla \mathcal{L}_{3D}(\theta) \approx \mathbb{E}_{\mathbf{C},t,\epsilon}\left[\omega(t)(\epsilon_p(\mathbf{z}_t, t, y, \mathbf{C}) - \epsilon)\frac{\partial g(\theta, \mathbf{C})}{\partial \theta}\right]. \tag{15}$$

IV. EXPERIMENT

A. IMPLEMENTATION DETAILS

1) GAUSSIAN SPLATTING INITIALIZATION

In the initial generation process, we randomly sample $N = 10,000$ points from the head of the SMPL model to serve as the initial point cloud. Then, each Gaussian point in the point cloud is initialized with the opacity $\alpha = 0.2$, scaling factor $\mathbf{s} = (0.02, 0.02, 0.02)$, color $\mathbf{c} = (0, 0, 0)$, and rotation quaternion $\mathbf{r} = (1, 0, 0, 0)$.

2) PARAMETERS OF GAUSSIAN SPLATTING

During the 3D generation process, we adopt the following parameters for Gaussian Splatting. We implemented a learning rate of 0.05 for opacity α, 0.01 for color $\mathbf{c}$, 0.001 for rotation $\mathbf{r}$, and 0.001 for scale $\mathbf{s}$. The learning rate for the position $\mathbf{x}$ exponentially decayed from 0.001 to 0.0001 as the training progressed. In the Gaussian splitting and cloning process, we set the gradient threshold β_E to be 0.0002. For points with scaling larger than 0.01, we employ the cloning operation, while for points smaller than 0.01, we utilize the splitting operation. During the surface-guided cloning and splitting process, we use the KNN algorithm to select 6 points for estimating the surface normal and generated new points along the surface. We performed splitting and cloning every 100 iteration steps, with the entire training process consisting of 3000 iterations.

3) STYLE ALIGNED SAMPLING LOSS

In the 3D stylized generation process, we employed the Stable Diffusion 1.5 [23] model as the base diffusion model. We utilized InstantID [39] to inject the identity information of the input image $\mathbf{I}$, with a control scale of 1.0. Before 3D generation, we randomly sampled 20 images corresponding to the target style, using a classifier-free guidance scale of 7.5 and 20 inference steps. We then compute the mean and covariance of these images in the latent space for Style-preserving Sampling. During SAS sampling, the classifier-free guidance scale is set to 50.0, and the noise scale for sampling is U(0.02, 0.98) for the first 1000 iterations and U(0.02, 0.5) for the latter 2000 iterations. The weight of the SAS loss is set to 1.0 throughout the training process.

4) MULTI-VIEW DIFFUSION MODEL

We utilize MVDream [8] as the multi-view diffusion model to provide 3D supervision during the 3D generation process. At each iteration, we randomly select four viewpoints with equal intervals and render the corresponding images, which are then fed into MVDream to compute the 3D SDS loss. The classifier-free guidance scale for MVDream is set to 10.0, and the weight of the 3D loss during the entire training process is 0.05.

B. QUALITATIVE EVALUATION

We qualitatively evaluate our approach through visual effects and comparison with SOTA methods. The results of our method are showcased in Fig. 4. For each case, we first present the basic 3D cartoon stylization using the text prompt "3D cartoon style." Subsequently, we demonstrate various stylizations, such as "green orc style," "golden statue style," and "realistic style." Our approach can generate high-quality 3D portraits from a given single image. The results demonstrate a well-controlled stylization that does not appear oversaturated. Furthermore, our method uses Gaussian Splatting and exhibits fast generation speeds, completing the whole process within 10 minutes.

To further evaluate our method, we conducted comparisons with existing single-image 3D generation techniques. Since current single-image 3D generation methods do not support stylization, we first stylize the given image and then use the corresponding method for image-to-3D

FIGURE 2. Qualitative comparison case 1.

generation. The comparative results are presented in Fig. 2 and Fig. 3, where it can be observed that existing image-to-3D generation methods exhibit lower generation quality and lack stability, producing oversaturated and unnatural results.

C. QUANTITATIVE EVALUATION

We also evaluate our approach through quantitative experiments. Firstly, we render a total of 120 rendered images around the circle. Then we employ CLIP to compute the consistency between the rendered images and the given image identity, as well as the consistency with the target text style. Additionally, to assess the naturality of the stylization, we calculate the average saturation as a measure of oversaturation. The results are presented in Table 1. Our method achieves better image identity consistency, and our style consistency is higher compared to other methods. Furthermore, the saturation of our generated results is lower, indicating a more natural stylization to a certain extent.

In addition to the visual comparisons, we also report the average generation time in Table 2. It can be observed that our method exhibits fast generation speeds compared to zero123 and dreamcraft3D. Our approach requires only 5 minutes to generate high-quality results.

D. USER STUDY

To further validate the visual quality of our method, we conducted a user study. In the user study, we collected preferences from 117 participants across 50 different sets of generation results. Each participant was randomly shown 20 sets, with each set containing results from four methods: DreamGaussian, Zero123, Dreamcraft3D, and our proposed approach. The participants were asked to select the result that best preserved the input image identity, the stylization they most preferred, and the result with the highest overall quality.

FIGURE 3. Qualitative comparison case 2.

TABLE 1. Quantitative comparisons with image-to-3D methods.

Method	ID (CLIP) ↑	Style (CLIP) ↑	Saturation ↓	Generation Time
Zero-1-to-3	57.84	31.72	65.04	43min
DreamCraft3D	58.22	32.33	70.08	95min
DreamGaussian	57.16	31.65	59.35	2min
Our method (w/o ID preserving)	52.23	32.75	51.75	10min
Our method (w/o style preserving)	57.84	31.78	68.54	10min
Our method (w/o 3D loss)	55.19	32.45	51.02	10min
Our method	58.63	32.79	50.61	10min

The statistical results, as illustrated in Figure 5, demonstrate that our method achieved the best performance.

E. ABLATION STUDY

We performed an ablation study to evaluate the contributions of various components in our method, including the SAS loss and the multi-view diffusion model. The results are presented in Table 1. It can be observed that without ID-preserving sampling, our method's identity consistency significantly decreases. If Style-preserving sampling is removed, our method exhibits reduced style consistency, increased saturation, and more unnatural generation results. Furthermore, the absence of the multi-view diffusion model leads to a decline in all evaluation metrics for our method's 3D generation results.

V. LIMITATION

While our method achieves a fast stylized generation of 3D portraits from a single image, it still requires a relatively long waiting time of approximately 10 minutes, which is not ideal for practical applications and user experiences. Additionally, our current approach can only generate a static

FIGURE 4. 3D stylized portrait generation results of our method.

3D portrait and lacks the capability to control dynamic facial expressions. Furthermore, we found that incorporating the instantID during stylization can influence the diversity of stylization results.

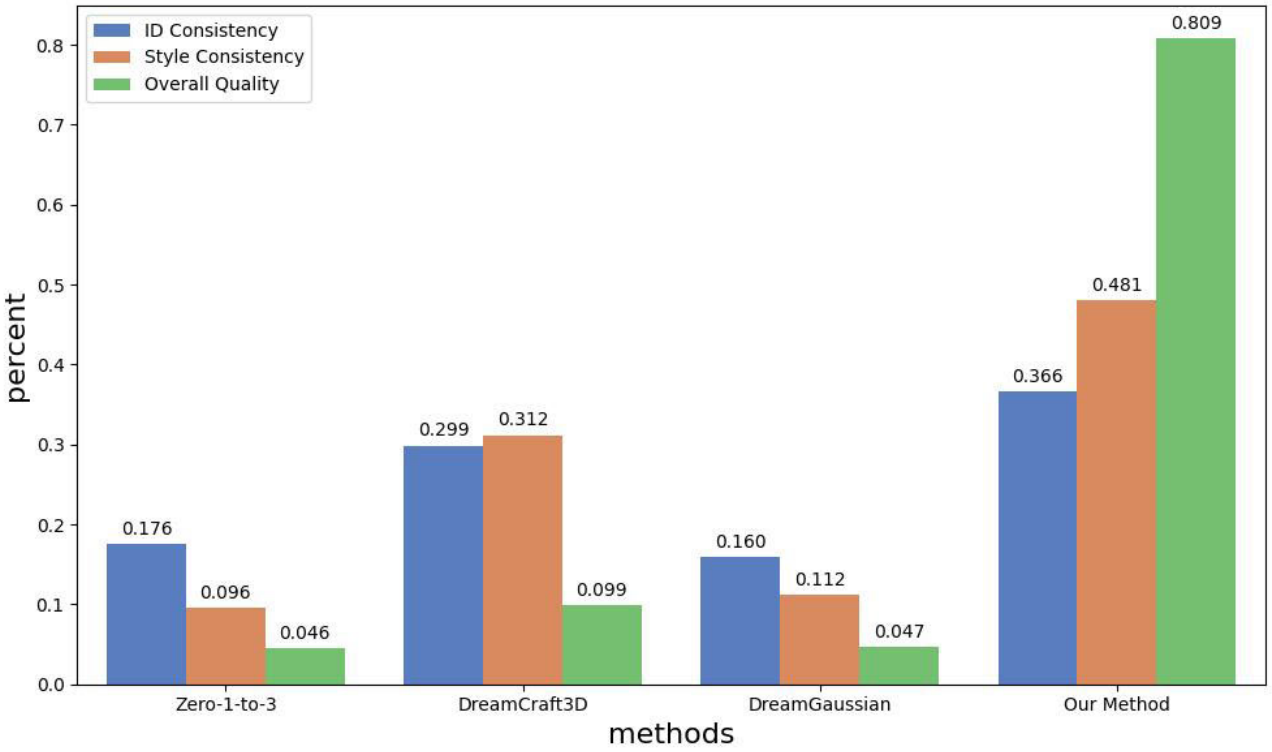

FIGURE 5. User Study of our methods and other image-to-3D methods.

VI. DISCUSSION

We presented an efficient approach for generating high-quality 3D stylized portraits from a single image input. Key strategies include: 1) A 3D Gaussian Splatting representation with a surface-guided splitting and cloning strategy to reduce noise. 2) A Stylized Generation Module injecting identity while stabilizing stylization to prevent oversaturation. 3) A multi-view diffusion model enforcing 3D consistency across viewpoints. Extensive experiments demonstrated our method's superiority over previous approaches in stylization quality, 3D consistency and user preference.

REFERENCES

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[2] R. Gal, O. Patashnik, H. Maron, G. Chechik, and D. Cohen-Or, "StyleGAN-NADA: CLIP-guided domain adaptation of image generators," 2021, *arXiv:2108.00946*.

[3] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, "StyleCLIP: Text-driven manipulation of StyleGAN imagery," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2065–2074.

[4] J. C. Pérez, T. Nguyen-Phuoc, C. Cao, A. Sanakoyeu, T. Simon, P. Arbeláez, B. Ghanem, A. Thabet, and A. Pumarola, "StyleAvatar: Stylizing animatable head avatars," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 8678–8687.

[5] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, and J. Clark, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 8748–8763.

[6] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, "Zero-1-to-3: Zero-shot one image to 3D object," 2023, *arXiv:2303.11328*.

[7] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "DreamFusion: Text-to-3D using 2D diffusion," 2022, *arXiv:2209.14988*.

[8] Y. Shi, P. Wang, J. Ye, M. Long, K. Li, and X. Yang, "MVDream: Multi-view diffusion for 3D generation," 2023, *arXiv:2308.16512*.

[9] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, no. 1, pp. 99–106, Jan. 2022.

[10] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2414–2423.

[11] Y.-H. Huang, Y. He, Y.-J. Yuan, Y.-K. Lai, and L. Gao, "StylizedNeRF: Consistent 3D scene stylization as stylized NeRF via 2D-3D mutual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18321–18331.

[12] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4396–4405.

[13] I. Skorokhodov, S. Tulyakov, and M. Elhoseiny, "StyleGAN-V: A continuous video generator with the price, image quality and perks of StyleGAN2," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3616–3626.

[14] G. Song, L. Luo, J. Liu, W.-C. Ma, C. Lai, C. Zheng, and T.-J. Cham, "AgileGAN: Stylizing portraits by inversion-consistent transfer learning," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–13, Aug. 2021.

[15] A. Haque, M. Tancik, A. A. Efros, A. Holynski, and A. Kanazawa, "Instruct-NeRF2NeRF: Editing 3D scenes with instructions," 2023, *arXiv:2303.12789*.

[16] J. Gu, L. Liu, P. Wang, and C. Theobalt, "StyleNeRF: A style-based 3D-aware generator for high-resolution image synthesis," 2021, *arXiv:2110.08985*.

[17] R. Shao, J. Sun, C. Peng, Z. Zheng, B. Zhou, H. Zhang, and Y. Liu, "Control4d: Dynamic portrait editing by learning 4D GAN from 2D diffusion-based editor," 2023, *arXiv:2305.20082*.

[18] Y. Li, Z.-H. Lin, D. Forsyth, J.-B. Huang, and S. Wang, "ClimateNeRF: Extreme weather synthesis in neural radiance field," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 3227–3238.

[19] L. Höllein, J. Johnson, and M. Nießner, "StyleMesh: Style transfer for indoor 3D scene reconstructions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6188–6198.
[20] F. Han, S. Ye, M. He, M. Chai, and J. Liao, "Exemplar-based 3D portrait stylization," *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 2, pp. 1371–1383, Feb. 2023.
[21] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34. Curran, 2021, pp. 27171–27183.
[22] T. Nguyen-Phuoc, F. Liu, and L. Xiao, "SNeRF: Stylized neural implicit representations for 3D scenes," 2022, *arXiv:2207.02363*.
[23] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10674–10685.
[24] F. Hong, M. Zhang, L. Pan, Z. Cai, L. Yang, and Z. Liu, "AvatarCLIP: Zero-shot text-driven generation and animation of 3D avatars," 2022, *arXiv:2205.08535*.
[25] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin, "Magic3D: High-resolution text-to-3D content creation," 2022, *arXiv:2211.10440*.
[26] J. Sun, B. Zhang, R. Shao, L. Wang, W. Liu, Z. Xie, and Y. Liu, "DreamCraft3D: Hierarchical 3D generation with bootstrapped diffusion prior," 2023, *arXiv:2310.16818*.
[27] R. Chen, Y. Chen, N. Jiao, and K. Jia, "Fantasia3D: Disentangling geometry and appearance for high-quality text-to-3D content creation," 2023, *arXiv:2303.13873*.
[28] X. Huang, R. Shao, Q. Zhang, H. Zhang, Y. Feng, Y. Liu, and Q. Wang, "HumanNorm: Learning normal diffusion model for high-quality and realistic 3D human generation," 2023, *arXiv:2310.01406*.
[29] Y. Cao, Y.-P. Cao, K. Han, Y. Shan, and K.-Y. K. Wong, "DreamAvatar: Text-and-shape guided 3D human avatar generation via diffusion models," 2023, *arXiv:2304.00916*.
[30] H. Jun and A. Nichol, "Shap-E: Generating conditional 3D implicit functions," 2023, *arXiv:2305.02463*.
[31] Z. Wang, C. Lu, Y. Wang, F. Bao, C. Li, H. Su, and J. Zhu, "ProlificDreamer: High-fidelity and diverse text-to-3D generation with variational score distillation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 1–36.
[32] G. Qian, J. Mai, A. Hamdi, J. Ren, A. Siarohin, B. Li, H.-Y. Lee, I. Skorokhodov, P. Wonka, S. Tulyakov, and B. Ghanem, "Magic123: One image to high-quality 3D object generation using both 2D and 3D diffusion priors," 2023, *arXiv:2306.17843*.
[33] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis, "3D Gaussian splatting for real-time radiance field rendering," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–14, Aug. 2023.
[34] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, "4D Gaussian splatting for real-time dynamic scene rendering," 2023, *arXiv:2310.08528*.
[35] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng, "DreamGaussian: Generative Gaussian splatting for efficient 3D content creation," 2023, *arXiv:2309.16653*.
[36] M. Kocabas, J.-H. Rick Chang, J. Gabriel, O. Tuzel, and A. Ranjan, "HUGS: Human Gaussian splats," 2023, *arXiv:2311.17910*.
[37] W. Zielonka, T. Bagautdinov, S. Saito, M. Zollhöfer, J. Thies, and J. Romero, "Drivable 3D Gaussian avatars," 2023, *arXiv:2311.08581*.
[38] Z. Li, Z. Zheng, L. Wang, and Y. Liu, "Animatable gaussians: Learning pose-dependent Gaussian maps for high-fidelity human avatar modeling," 2023, *arXiv:2311.16096*.
[39] Q. Wang, X. Bai, H. Wang, Z. Qin, A. Chen, H. Li, X. Tang, and Y. Hu, "InstantID: Zero-shot identity-preserving generation in seconds," 2024, *arXiv:2401.07519*.

SHANGMING JIANG received the B.S. degree in electrical engineering from Wuhan University of Technology, Wuhan, China. He was an Electrical Engineer with Anhui Construction Engineering Group Company Ltd., Hefei, China, where he has been a Senior Professional Engineer with the Mechanical and Electrical Intelligence Branch, Architectural Design and Research Institute. His research interests include electrical systems and intelligent building design.

XINYOU YU received the B.S. degree in electrical engineering from Anhui University of Technology, Ma'anshan, China. He was an Electrical Engineer with Anhui Construction Engineering Group Company Ltd., Lu'an, China, where he has been a Senior Engineer with Lu'an Branch Company. His research interests include electrical power systems and building electrification.

WEIJUN GUO received the B.S. degree in automation from Fuzhou University, Fuzhou, China. He has been a Senior Engineer with the Mechanical and Electrical Intelligence Branch, Architectural Design and Research Institute, Anhui Construction Engineering Group Company Ltd., Hefei, China, where he is currently an Automation Engineer. His research interests include building automation systems and intelligent controls.

JUNLING HUANG received the B.S. degree in automation from Jiamusi University, Jiamusi, China. He has been a Senior Engineer with the Mechanical and Electrical Intelligence Branch, Architectural Design and Research Institute, Anhui Construction Engineering Group Company Ltd., Hefei, China, where he is currently an Automation Engineer. His research interests include intelligent building systems and electrical automation.