SeparateGen: Semantic Component-based 3D Character Generation from Single Images

Dong-Yang Li, Yi-Long Liu, Zi-Xian Liu, Yan-Pei Cao, Meng-Hao Guo, and Shi-Min Hu *Fellow, IEEE*

***Abstract*—Creating detailed 3D characters from a single image remains challenging due to the difficulty in separating semantic components during generation. Existing methods often produce entangled meshes with poor topology, hindering downstream applications like rigging and animation. We introduce SeparateGen, a novel framework that generates high-quality 3D characters by explicitly reconstructing them as distinct semantic components (e.g., body, clothing, hair, shoes) from a single, arbitrary-pose image. SeparateGen first leverages a multi-view diffusion model to generate consistent multi-view images in a canonical A-pose. Then, a novel component-aware reconstruction model, SC-LRM, conditioned on these multi-view images, adaptively decomposes and reconstructs each component with high fidelity. To train and evaluate SeparateGen, we contribute SC-Anime, the first large-scale dataset of 7,580 anime-style 3D characters with detailed component-level annotations. Extensive experiments demonstrate that SeparateGen significantly outperforms state-of-the-art methods in both reconstruction quality and multi-view consistency. Furthermore, our component-based approach effectively resolves mesh entanglement issues, enabling seamless rigging and asset reuse. SeparateGen thus represents a step towards generating high-quality, application-ready 3D characters from a single image. The SC-Anime dataset and our code will be publicly released.**

***Index Terms*—3D character generation, single image generation, semantic components, multi-view images, large reconstruction model.**

I. Introduction

GENERATING high-quality, animatable 3D characters from a single image is a highly sought-after goal with significant implications for film, gaming, and virtual reality. While manual 3D modeling remains the gold standard, it is a time-consuming and skill-intensive process. Recent advances in image-conditioned 3D generative models offer a promising alternative, but generating detailed, *reusable* 3D characters from a single image remains a significant challenge [1]–[4]. Existing methods often struggle with complex character structures, diverse poses, and the inherent ambiguity of reconstructing a 3D object from a single 2D projection. These challenges frequently result in models with topological errors, poor geometry, and entangled components, severely limiting their use in downstream applications like animation and asset reuse.

Earlier approaches for image-driven character generation often relied on parametric human models like SMPL [5], using image generation models to guide the optimization process [6]–[10]. However, these methods are inherently limited by the expressiveness of the underlying parametric model and often struggle to accurately capture diverse body shapes, clothing, and hairstyles. Recent methods like CharacterGen [11] have shown impressive results by directly generating 3D characters using multi-view diffusion models, avoiding the limitations of parametric models. Nevertheless, CharacterGen, like other holistic approaches, often produces topologically incorrect meshes with fused components. This "mesh entanglement" arises from the difficulty in separating semantic parts during the generation process, hindering downstream tasks that require well-defined, independently manipulable components.

In stark contrast to these holistic approaches, professional 3D artists typically model characters as a collection of semantically distinct parts (e.g., body, clothing, hair, accessories). This component-based approach allows for greater control, facilitates detail modeling, and simplifies tasks like rigging, animation, and asset reuse. Inspired by this practice, we argue that explicitly decomposing the character into its semantic components during generation is a crucial step towards creating high-quality, application-ready 3D characters.

In this paper, we propose SeparateGen, a novel method for generating 3D characters composed of semantic components from a single image with arbitrary poses. As illustrated in Fig. 1, our method differs from traditional holistic 3D model generation approaches by explicitly decomposing the character into four distinct components: body, clothing, hair, and shoes. The key concept behind SeparateGen is to explicitly separate the entire character model into components while using the same set of input images to condition the generation process. Each component maintains a consistent reconstruction resolution with the whole character and preserves the relative spatial relationships between components and the overall character. This effectively addresses issues related to model errors and the loss of reconstruction details due to the lack of semantic information.

SeparateGen contains two mainly stages: canonical pose multi-view images generation and semantic component-based 3D character reconstruction. The first step converts input images in arbitrary poses into a canonical pose and generates

Dong-Yang Li, Meng-Hao Guo, Shi-Min Hu are with BNRist, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: ldy23@mails.tsinghua.edu.cn; shimin@tsinghua.edu.cn, Corresponding author).

Yi-Long Liu is with Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China.

Zi-Xian Liu is with Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing 100084, China.

Yan-Pei Cao is with VAST, Beijing 100084, China.

Manuscript received March 14, 2025;

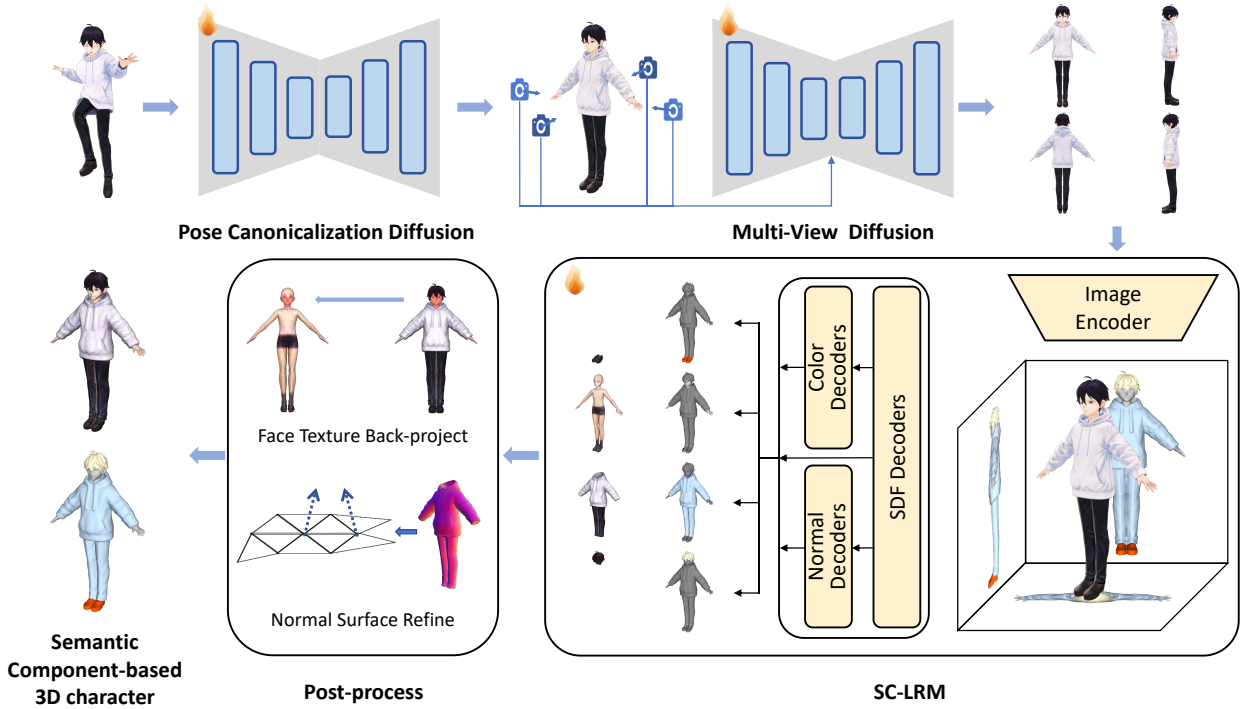

Fig. 1. Overview of SeparateGen. Given an arbitrary-pose image, we first apply Pose Canonicalization Diffusion to obtain an A-pose, followed by a multi-view diffusion model to generate multi-view RGB images (see Sec. III-A). Then, SC-LRM decomposes the model into four semantic components and extracts color, normal, and mesh for each (see Sec. III-B). Finally, Face Texture Back-project and Normal Surface Refinement enhance facial details and mesh quality (see Sec. III-B5), yielding a semantic component-based 3D character.

consistent multi-view A-pose images. This step guarantees multi-view consistency and addresses the challenges posed by self-occlusion and the diversity of human poses in the subsequent process.

In the second step, we use SC-LRM, designed based on LRM [4], to generate high-quality semantic components using the multi-view images produced in the first stage. SC-LRM overcomes the challenge of sparse-view 3D character reconstruction for semantic components. It incorporates multiple semantic decoders, which decompose the character into distinct components according to semantic information, effectively resolving the mesh errors and component fusion problems caused by the lack of semantic clarity in holistic model generation. Due to the inherent resolution limitations of LRM, SC-LRM addresses the issue of insufficient precision by utilizing feature plane super-resolution. This ensures that each semantic component corresponds to high-resolution feature planes, thus improving detail accuracy and quality. Additionally, we unify the entire image as the conditioning input, enabling automatic decomposition and ensuring that the generated components preserve their relative spatial relationships. This approach effectively addresses the challenges posed by inter-component occlusions and the loss of spatial relationships after component generation, eliminating the need for subsequent spatial position optimization. Furthermore, inspired by [12], SeparateGen incorporates a surface refinement process after extracting the initial coarse meshes, enhancing details through the normal field guidance and inverse texture mapping.

To train our SeparateGen, we build SC-Anime, a dataset focused on anime characters with semantic parts. We decompose each character into four semantic parts. The resulting dataset includes 7,580 semantically annotated characters, with both the full characters and individual components rendered from multiple viewpoints.

Quantitative and qualitative comparisons with the state-of-the-art methods demonstrate that our approach outperforms previous works in terms of both generation quality and consistency. The qualitative comparison highlights how component decomposition resolves mesh connection errors, proving its superior performance in supporting downstream tasks such as skeletal rigging and asset reuse. Our SeparateGen enables the decomposition of online character images, providing change to build a vast collection of 3D character components for freely constructing character models.

In summary, our work makes the following contributions:

- a reconstruction model, SC-LRM, utilizes multiple decoders to adaptively separate the geometry, color, and normal fields of semantic character components to generate characters with high-quality geometry and detailed textures.
- a dataset, SC-Anime, contains 7,580 semantically decomposed characters for training and testing our method.
- a convincing result, shows that our method outperforms previous works in both quantitative and qualitative experiments, indicating the usability of generated models in downstream character applications.

II. Related Works

A. Decomposed 3D Representation and Generation

With remarkable advancements in 3D generation [13], [14], significant progresses have made in this domain, ranging from 2D diffusion guidance-based generation [2], [3], [15]–[17] using Score Distillation Sampling (SDS) [1], to multiview-based generation [18]–[20], to video-based generation for multi-view consistency [21]–[23], and to LRM-based generation [4], [24]–[27]. Additionally, the emergence of native 3D generation

techniques [28]–[31] has further enhanced the quality of 3D outputs. However, the primary focus of these advancements remains on the generation of single objects.

Compared to single-object generation, multi-object 3D model generation remains underdeveloped, as it requires both high-quality object reconstruction and consideration of the spatial relationships of individual objects within the generated 3D model, such as positioning and occlusion. ObjectSDF [32] and ObjectSDF++ [33] propose a neural radiance field-based object decoupling technique, which uses multiple decoders to transform a neural field into the corresponding Signed Distance Function(SDF) [34] values of individual objects.

DELTA [35] utilizes a hybrid combination of explicit and implicit 3D representations, enabling the joint reconstruction of compositional avatars. LayGa [36] leverages multi-layered Gaussian representations based on segmented semantic information, enabling the creation of layered animatable avatars for clothing transfer with realistic details. NCHO [37] learns to decompose the human body and objects by generating multiple triplanes and then recombining them into a synthesized human model in an unsupervised manner. To disentangle each component, TELA [38] introduces a multi-layer clothed human model parameterized by multiple Neural Radiance Fields (NeRF) [39]. Part123 [40] generates images through multi-view diffusion, and achieves 3D reconstruction with corresponding vertex classification information by utilizing 2D segmentation and contrastive learning. REPARO [41] employs a two-step approach: extracting and reconstructing individual objects as 3D meshes with pre-trained image models, followed by optimizing them for coherent scene composition using differentiable rendering. Frankenstein [42] proposes a diffusion-based framework capable of generating semantically composed 3D scenes in a single process. However, its application is limited to geometry generation and lacks the ability to enforce conditional constraints.

B. 3D Avatar Generation

Early character representation methods often relied on parametric body models such as SMPL [5] and SMPLX [43], which were combined with neural networks for prediction tasks. For instance, BCNet [44] constructs a clothing model based on SMPL and estimates clothing skinning weights using neural networks. Similarly, LGN [45] extends SMPL by incorporating signed distance fields (SDF) to represent different layers of clothing, offering a more detailed geometric representation. ICON [46] and ECON [47] further integrate SMPL-based models to optimize dressed human reconstructions by predicting body normal maps. Meanwhile, implicit function-based approaches such as PIFu [48], PIFuHD [49], and Geo-PIFu [50] leverage pixel-aligned implicit functions to represent the human body, enabling more efficient and high-resolution reconstructions. Some recent works [51], [52] focus on dynamic garment modeling by generating clothed human animations and predicting fine-grained cloth deformations.

More recently, diffusion-based methods for 3D human generation [6]–[10] have emerged, primarily relying on 2D diffusion models and optimizing the generation process using SDS loss. Some approaches, such as DreamAvatar [6] and TADA [9], employ SMPL [5] as a geometric prior to enhance structural consistency. Others, such as AvatarVerse [10], integrate ControlNet [53] to provide additional SDS guidance, improving controllability during generation. However, these methods often suffer from the ”Janus problem,” where multi-view avatar reconstructions exhibit inconsistencies between the front and back views, posing challenges for achieving globally coherent 3D representations.

Recently, SO-SMPL [54] introduced a representation that models the human body and clothing as two separate meshes, linked through offsets to ensure physical alignment between the body and clothing. This method also uses SDS for distillation to enhance generation quality. Additionally, Frankenstein [42] demonstrated certain capabilities in part-based modeling. However, these approaches remain unable to leverage image prompts for text-driven 3D character generation, which is essential for controllable character creation. In parallel, MikuDance [55] showcased the potential of arbitrarily driving 2D characters for diverse animations, demonstrating impressive effectiveness in 2D character animation and visualization.

CharacterGen [11] combines a multi-view diffusion model with a large reconstruction model (LRM) [4] to enable image-conditioned 3D character generation. It significantly improves view consistency and condition control, but still faces limitations in part-based modeling and the quality of geometric details. Unlike the original LRM adopted in CharacterGen, our proposed SC-LRM introduces a disentangled representation by separating geometric and image feature planes, thereby decoupling geometry from texture appearance. Beyond the rendering supervision used in LRM, SC-LRM incorporates additional normal supervision and SDF-based geometric supervision, and further leverages super-resolution techniques on the feature planes to enhance fine details. Moreover, instead of generating an entire character mesh in prior works, our framework explicitly generates semantic components, which are subsequently composed into a complete character model.

III. Methods

A. Pose Canonicalization and Multi-view Generation

Given a reference character image in an arbitrary pose, our goal is to generate multi-view images under a canonicalized pose while maintaining 3D consistency. Due to the complex joint structures of 3D character models, rendered 2D images often exhibit intricate occlusion relationships, making subsequent multi-view generation and 3D reconstruction more challenging. To address this, we convert the input image into a canonical A-pose, which is widely used in 3D character modeling.Inspired by previous works [11], [56], we employ a diffusion model that integrates features from the conditional image during generation. For A-pose generation, we retrain the model by injecting the arbitrary-pose image into the network using the duplication of spatial self-attention proposed in [57], combined with cross-image attention. Only the attention layers are trained, with the A-pose image serving as the supervision signal. This design enables effective transformation of a 2D reference image with an arbitrary pose into its corresponding

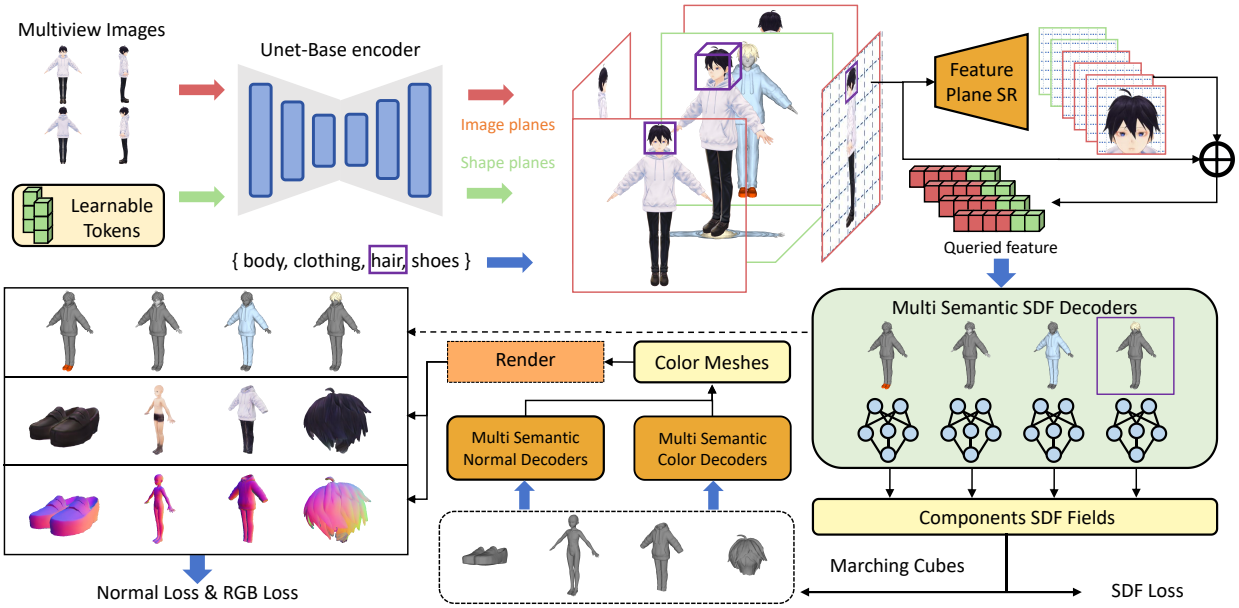

Fig. 2. SC-LRM Framework. SC-LRM combines input multi-view images and learnable tokens, passing them through a UNet-based encoder to generate six feature planes. Taking the hair component as an example, valid regions are cropped and enhanced through feature plane super-resolution. Then, the queried features are decoded into SDF, normal, and color fields using multi semantic decoders. The SDF fields are used to extract the mesh using the Marching Cubes and supervised by the SDF loss. The colored meshes are rendered using "Render", a differentiable renderer employed for both RGB and normal supervision.

A-pose image. Once the A-pose image is obtained, we build upon MV-Adapter [57] as our baseline. To accommodate the fixed four-view setting, we constrain the camera conditions to specific perspectives and freeze the cond-encoder, while keeping other components unchanged. We then train only the multi-view and cross-image attention modules within the self-attention layers. We train the model using the A-pose image rendered at a $0°$ elevation and $0°$ azimuth as input, with supervision provided by four perspective views at a $10°$ elevation with azimuths of $0°$, $90°$, $180°$, and $270°$. Additionally, during training, we randomly resize the resolution of the reference images to improve the robustness of the multi-view generation process. As a result, the model can generate high-resolution multi-view images even when provided with low-resolution input images.

B. Semantic Decomposed 3D Character Generation

1) SC-LRM: Our SC-LRM aims to generate high-quality, semantically decomposed 3D character that can adaptively reconstruct components based on the input four-view images. The decomposition is designed to enable each component to be generated at higher resolution with independent details, while separating different component features to strengthen each decoder's capacity for modeling specific features.

An overview of our SC-LRM is shown in Fig. 2. In SC-LRM, the input four-view images, together with a set of learnable tokens with shape $2 \times d_{\text{img\_emb}} \times H \times W$, where $d_{\text{img\_emb}} = 32$, are processed by a UNet-based encoder. The input images are first projected to the same embedding dimension as the learnable tokens, and then concatenated along the channel dimension to form the overall UNet input. These learnable tokens act as additional parameters that introduce geometric priors beyond what can be directly extracted from images, enabling the network to capture complex geometric information. The UNet output consists of six implicit feature planes: four correspond to the input four-view images, denoted asP_{img1}, P_{img2}, P_{img3}, and P_{img4}while the remaining two are geometry-specific feature planes derived from the learnable tokens, corresponding to the XY and YZ planes, denoted as P_{geo1} and P_{geo2}. This design allows for better extraction of image information while explicitly encoding geometric features. The following equations illustrate how these six planes are queried to compute SDF values and surface values:

$$\text{sdf} = \textbf{Query}(P_{\text{geo1}}, P_{\text{geo2}}, P_{\text{img1}}, P_{\text{img2}}), \tag{1}$$

$$\text{color} = \textbf{Query}(P_{\text{img1}}, P_{\text{img2}}, P_{\text{img3}}, P_{\text{img4}}). \tag{2}$$

For semantic components, we first obtain coarse components by directly processing the six feature planes through the decoders to determine their effective regions on the feature planes. Then, we crop their effective regions from the six planes and apply super-resolution to enhance their details. Based on the super-resolved feature planes, multiple decoders generate the SDF fields corresponding to each semantic component from the shared set of six planes. The Marching Cubes (MC) [58] algorithm is then used to extract the meshes of the components from their SDF fields. During training, the vertex positions of the extracted meshes are used to compute their corresponding color and normal information. The meshes are then differentiably rendered into 2D images using Nvdiffrast [59], which are supervised against the ground-truth 2D images. Additionally, the predicted SDF fields are supervised using a loss function based on the ground-truth SDF values to ensure the accuracy of geometric reconstruction.

2) Decompsed Generations: Inspired by prior works such as [32], [33], [42], it has been demonstrated that decoding semantic information as an additional output often leads to rigid segmentation, resulting in incomplete or fragmented components. So we extend the single feature-plane decoder in SC-LRM into a decoder group, where the geometric decoder and surface decoder are expanded into multiple decoders, each controlled by a semantic switch. This design facilitates the direct generation of multiple semantic components from the same implicit feature planes.

Given the limited availability of 3D data, particularly the scarcity of datasets with semantic component annotations, it is challenging for models to directly learn to reconstruct semantically decomposed 3D characters from scratch. To address this limitation, we pretrain our model on the Objaverse [60] dataset under supervision for reconstructing complete objects, enabling it to acquire general 3D reconstruction capabilities. The feature plane decoder trained during this pretraining phase serves as the initialization for decoding multiple semantic components. Each decoder with sufficient 3D reconstruction capabilities is further optimized using the component data from our dataset. This optimization allows each decoder to specialize in preserving the features related to its corresponding semantic component, while discarding features unrelated. After training, each decoder is fine-tuned to decode only the information related to the specific semantic component, ultimately reconstructing the corresponding components.

With this approach, SC-LRM generates geometric SDF fields and normal fields for four semantic components, including body, clothing, hair, and shoes, by using the input four-view images. Additionally, as our dataset preserves the relative spatial relationships between individual components and the complete model, the components reconstructed by SC-LRM maintain spatial consistency with the input images, ensuring precise alignment in 3D space.

3) Region-Based Plane Super-Resolution: Due to the low resolution of the feature planes in LRM, as specified in the original work with 96×96 resolution, our SC-LRM adopts a higher initial resolution of 448×448. However, for semantic components, the effective regions on feature planes remain relatively small. Directly increasing the resolution of feature planes would significantly increase computational overhead. In theory, the features of any point in space can be linearly interpolated from features of surrounding grid points, and an MLP allows the model to output shapes with continuous representations at arbitrary resolutions. However, grid interpolation often struggles to capture fine-grained details within individual grid cells, thereby constraining the actual output resolution of the reconstructed models to the resolution of grid points.

Inspired by [61], we hypothesize that the feature planes of objects of the same class share inherent similarities, which can potentially be enhanced using super-resolution. Based on this insight, we first sample across the entire space to localize the effective regions of each component, then project their spatial positions from the bounding boxes onto the feature plane and crop the corresponding areas. These regions are then upsampled to a higher resolution and optimized using a zero-initialized convolutional network that refines the upsampled feature planes while maintaining consistent resolution between input and output. Residual connections were employed to integrate the output with the upsampled regions, producing a refined, higher-resolution feature plane.

4) Training Loss: The original LRM [4] minimized image reconstruction objectives, including MSE Loss and LPIPS Loss [62], between the rendered views and the ground-truth views. Additionally, we introduced a binary-cross-entropy loss between the mask rendered from the model and the mask derived from the ground-truth alpha channel. This additional loss helps the model better identify effective regions:

$$L_{\text{render}} = \lambda_{\text{mse}} L_{\text{MSE}} + \lambda_{\text{lpips}} L_{\text{LPIPS}} + \lambda_{\text{mask}} L_{\text{mask}}. \quad (3)$$

To accelerate geometric reconstruction, we employed explicit 3D supervision in addition to the 2D rendering-based supervision, avoiding the two-stage training process of NeRF-SDF and achieving faster and more stable convergence for geometry. Specifically, we used the ground-truth SDF as supervision. In practice, we randomly sampled 100,000 points in the normalized space and another 100,000 points near the mesh surface. The SDF values of these points were computed as ground truth, and the SDF values predicted by SC-LRM for the same points were minimized using MSE Loss:

$$L_{\text{SDF}} = \lambda_{\text{sdf\_random}} L_{\text{SDF\_random}} + \lambda_{\text{sdf\_near}} L_{\text{SDF\_near}}. \quad (4)$$

Furthermore, instead of computing normals from the reconstructed mesh geometry, we directly predicted the normal field and queried normals at the input points. Similar to texture predictions, the normal field was optimized using the similar loss:

$$L_{\text{normal}} = \lambda_{\text{mse\_n}} L_{\text{MSE\_n}} + \lambda_{\text{lpips\_n}} L_{\text{LPIPS\_n}}. \quad (5)$$

The loss function for a single component and the total loss function for the model are defined as:

$$L_{\text{single}} = L_{\text{render}} + L_{\text{SDF}} + L_{\text{normal}}, \quad (6)$$

$$L = \sum L_i, \quad i \in \text{body, shoes, hair, clothing}. \quad (7)$$

5) Post-processing Refinement: Our proposed SC-LRM efficiently reconstructs 3D character models composed of semantic components. However, the final surface extracted from the SDF field using the Marching Cubes (MC) [63] algorithm often exhibits surface irregularities, making further mesh optimization necessary. As an essential property of 3D geometry, normal information reflects the surface details of 3D models. Inspired by MeshFormer [12], we avoid deriving a normal map from the reconstructed geometry and using normal loss as a supervisory constraint for geometric features. Instead, we treat normals as a texture and apply the same processing pipeline as surface color, decoupling the final normal generation from the geometric computation process. The learned normal texture can then be exported through the mesh. Additionally, we adopt a post-processing algorithm [64], where the vertex positions of the generated mesh are optimized using the corresponding normal information as a reference, further enhancing geometric quality. Despite the super-resolution applied to the body features, the effective surface area corresponding to facial details remains relatively small. To address this, we utilize the SAM model [65] to segment the facial region from the frontal view of the character in the conditional image. The segmented facial details are back-projected onto the optimized mesh, significantly enhancing the quality of facial textures.

C. SC-Anime Dataset

1) Data Process: Existing large-scale 3D datasets, such as Objaverse [60], OmniObject3D [66], and the latest 3D character dataset Anime3D [11], do not contain component-level semantic information. Thus we collected a variety of

Fig. 3. An example character from our SC-Anime, showcasing image in arbitrary pose and its A-pose image, with examples of decomposed components.

anime character models from VRoid-Hub [67]. Through rigorous manual screening and semantic annotation, we excluded models unsuitable for decomposition, such as those with inconsistent geometry or rendering artifacts caused by transparent materials. The final dataset contains 7,580 high-quality character models annotated with semantic component information. An example from the dataset is shown in Fig. 3.

2) Semantic Component Annotation and Rendering: We first separated character components according to the part naming conventions defined in the VRM format [68], which is the standard data format used in VRoidHub. The components were categorized into four semantic groups: body, clothing, hair, and shoes. In addition, some misannotated parts were manually corrected and reassigned to their appropriate semantic categories. Using Blender [69], we rendered both the complete character models and the semantically separated components. We rendered both the complete model and the individual components in A-pose and arbitrary poses. For A-pose characters, we added specific adjustments. The left and right arms were rotated $45°$ about the Z-axis, and to enhance the hand details and prevent finger overlaps, we sequentially rotated the fingers from the thumb to the pinky finger by $\{0°, 22.5°, 0°, -20°, -45°\}$ to avoid any sticking. Other joint parameters remained unchanged. For characters in arbitrary poses, we also used 10 human skeleton animations from Mixamo [70]. The camera was configured with a $40°$ field of view (FoV) and positioned 1.5 units away from the origin of the scene. During rendering, the camera's position was randomly sampled around the model, always directed towards its center to ensure proper focus. For component rendering, no additional normalization was applied in order to preserve the spatial relationships of each component with respect to the full character model. This ensures that the components retain their original spatial information.

3) Geometric Representation: Since our 3D reconstruction method relies on SDF representations [34], it is essential to convert the original non-watertight meshes into watertight ones. While prior works such as PatchGrid [71] and 3PSDF [72] have shown promising results in computing SDFs directly on non-watertight meshes for single-mesh reconstruction tasks, 3D generation requires a more generic and robust SDF formulation. Drawing inspiration from the preprocessing strategy of [29] [73], we first extract the unsigned distance field (UDF) [74] from the input mesh. We then apply the Marching Cubes (MC) algorithm [58] to generate a thin watertight surface near the zero-level set by extracting the isosurface where the UDF values equal a small threshold ϵ. This watertight surface is subsequently used to compute the SDF field for training.

IV. Experiments

A. Implementation Details

We split our SC-Anime dataset into a training set and a testing set with a 99:1 ratio. SDXL [75] is used for pose canonicalization and the multi-view generation. Both processes are trained at an image resolution of 768×768. We use AdamW as the optimizer with a learning rate of 5×10^{-5}, momentum parameters $\beta = (0.9, 0.999)$, and a weight decay of 0.01. The training is conducted for 10 epochs with batch size of 1. For training SC-LRM, we initialize the decoder group using a single decoder pre-trained on the Objaverse dataset [60]. During each training step, we randomly select one semantic component, render it from arbitrary viewpoints, and select four images for 2D supervision. Additionally, for geometric supervision of the SDF representation, we randomly sample points from the ground-truth SDF field. In the semantic component training phase, Region-Based Plane Super-Resolution operations on the feature planes are not applied during the first 20,000 steps to ensure that the coarse reconstruction results, generated from the component decoders across the entire feature space, reach a near-stable state. After 20,000 steps, the region-based operations are activated to refine the features further. The loss function parameters are set as $\lambda_{\text{sdf\_random}}, \lambda_{\text{sdf\_near}}, \lambda_{\text{mse}}, \lambda_{\text{mask}}, \lambda_{\text{lpips}}, \lambda_{\text{mse\_n}}, \lambda_{\text{lpips\_n}} = 1.0, 1.0, 1.0, 0.1, 1.0, 1.0, 1.0$. All models were trained on 8 × A100 GPUs. The A-pose and multi-view generation models required only a few hours of training, while SC-LRM was first trained as a general-purpose 3D generator and then fine-tuned for multi-component generation over about two days.

B. Results and Comparison

We conducted experiments on 2D character multi-view image generation and 3D character mesh generation to evaluate the effectiveness of our character generation method. The evaluation primarily utilized SSIM [76], LPIPS [62], and FID, as well as using CLIP [77] to compute the cosine similarity between the condition images and the generated images as quantitative metrics.

1) Generation from Canonical Pose Inputs: Noting that many existing methods do not support direct generation from arbitrary pose inputs, we first conducted experiments using canonical A-pose images as input. We evaluated our 2D multi-view image generation model on our SC-Anime dataset and compared the results with Zero123 [18], SyncDreamer [78], CharacterGen [11], SV3D [21], Hi3D [23], V3D [22] and MV-Adapter [57]. For methods whose original training datasets did not include character data but provided training scripts,

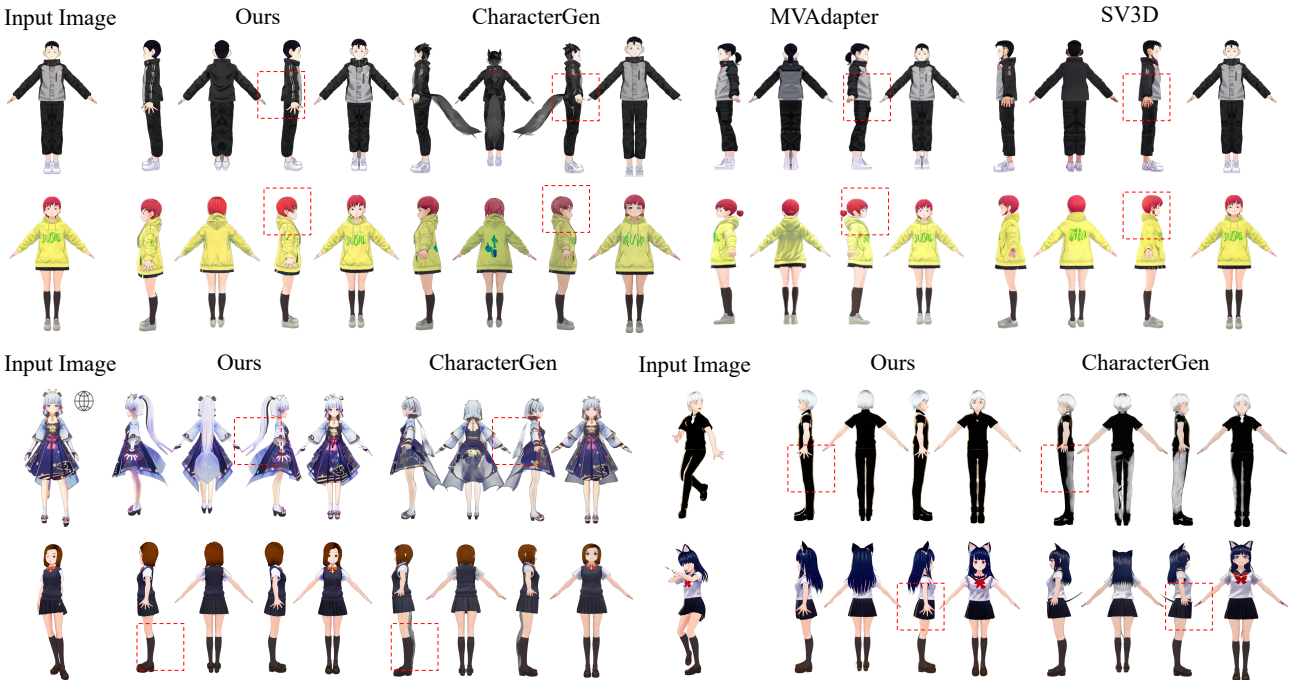

Fig. 4. Comparison of generated multi-view A-pose character images produced by our method and others. The top two rows correspond to results obtained from A-pose inputs, while the bottom two rows correspond to results obtained from arbitrary-pose inputs. denotes images collected from the Internet.

TABLE I
QUANTITATIVE COMPARISON OF 2D MULTI-VIEW GENERATION METHODS ON THE TEST SPLIT OF SC-ANIME

Methods	SSIM↑	LPIPS↓	FID↓	CLIP Score↑
Zero123	0.8070	0.1871	0.3970	0.8990
Zero123(Finetuned)	0.8133	0.1806	0.3260	0.8984
SyncDreamer	0.8463	0.1686	0.0577	0.8687
SyncDreamer(Finetuned)	0.8606	0.1363	0.0514	0.8936
SV3D	0.8947	0.0963	0.0730	0.9281
V3D	0.8787	0.1324	0.2110	0.9069
Hi3D	0.8968	0.1121	0.0838	0.8986
MV-Aapater	0.8788	0.1180	0.0659	0.9507
CharacterGen(2D)	0.8870	0.1009	0.0618	0.9226
SeparateGen(2D)	**0.9287**	**0.0551**	**0.0415**	**0.9695**

TABLE II
QUANTITATIVE COMPARISON OF 3D CHARACTER GENERATION METHODS ON THE TEST SPLIT OF SC-ANIME

Methods	SSIM↑	LPIPS↓	FID↓	CLIP Score↑	CD↓
Era3D	0.8906	0.1037	0.0953	0.9033	0.0078
LGM	0.8969	0.1009	0.1874	0.8736	0.0088
InstantMesh	0.9019	0.0946	0.1494	0.8789	0.0081
Unique3D	0.9022	0.0922	0.1078	0.9037	0.0088
SV3D	0.8775	0.1370	0.1723	0.8009	0.0091
V3D	0.8866	0.1105	0.1175	0.8069	0.0081
CharacterGen(3D)	0.8937	0.1006	0.0800	0.8939	0.0078
SeparateGen(3D)	**0.9091**	**0.0743**	**0.0769**	**0.9147**	**0.0009**

we additionally performed fine-tuning. The quantitative results are shown in Table I, with the generated results displayed in the upper part of Fig. 4. As shown, our method demonstrates superior performance compared to existing methods, which struggle to maintain sufficient geometric and appearance consistency in generated images.

Since existing methods are unable to generate semantically decomposed 3D character models from a single image with arbitrary poses, we compared our semantically decomposed character models with monolithic models generated by other methods on the SC-Anime test split, including Era3D [31], LGM [24], InstantMesh [26], Unique3D [79], SV3D [21], V3D [22] and CharacterGen [11]. The texture quality of the models was evaluated by comparing rendered images from four camera viewpoints with the ground truth images. Additionally, Chamfer Distance (CD) was used as a metric to assess the geometric quality of the generated meshes. The quantitative results are shown in Table II. The generated character models from multiple methods are displayed in Fig.5. The comparative results reveal significant limitations in existing methods, including low geometric quality and reconstruction accuracy. In contrast, our component-base generation not only improves fidelity but also better aligns with the original mesh design compared to monolithic mesh generation.

Furthermore, to quantitatively assess the consistency of the 2D and 3D generated results with the input images, we used CLIP to compute the cosine similarity of image features, which we refer to as the CLIP score. The detailed results are presented in Table I and Table II. Our method performs well in both the 2D and 3D generation phases, demonstrating that our generated results maintain strong consistency in appearance features with the input images.

2) Generation from Arbitrary Pose Inputs: Since CharacterGen supports generating 2D multi-view images and 3D character models from images with arbitrary poses converted into canonical poses, we also compared our method using arbitrary pose inputs on our proposed dataset's test set with CharacterGen. The quantitative results for 2D multi-view generation and 3D character reconstruction are shown in Table III, with visual results in the lower part of Fig.4 for 2D and Fig.6 for 3D. CharacterGen exhibits issues such as color shifts, texture detail loss, and limited geometric accuracy when converting arbitrary poses to A-pose. In contrast, our method demonstrates superior texture quality and enhanced geometric detail, while maintaining 3D character decomposability. CharacterGen exhibits issues such as geometric distortion, loss of details, and limited accuracy when generating 3D characters, while other image-conditioned generation methods tend to produce fragmented geometric models, making them unsuitable

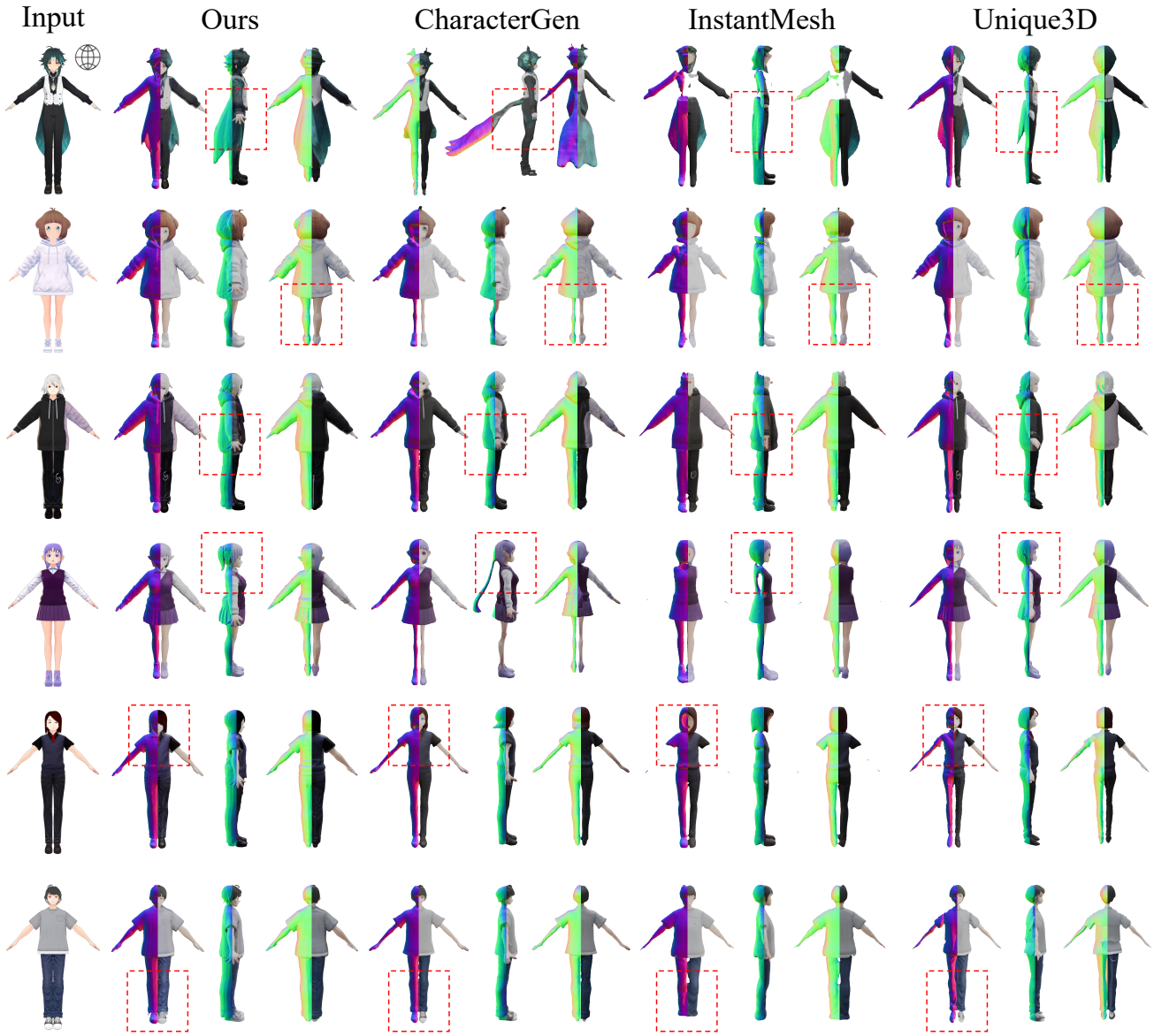

Fig. 5. Comparison of the appearance and geometry of 3D characters generated by our method and other methods using A-pose inputs. 🌐 denotes images collected from the Internet.

TABLE III
COMPARISON WITH CHARACTERGEN USING IMAGES WITH ARBITRARY POSES AS INPUT

Methods	SSIM↑	LPIPS↓	FID↓	CLIP Score↑	CD↓
CharacterGen(2D)	0.8586	0.1404	0.0762	0.8781	-
SeparateGen(2D)	**0.8643**	**0.1334**	**0.0570**	**0.9043**	-
CharacterGen(3D)	0.8713	0.1218	0.1009	0.8834	0.0090
SeparateGen(3D)	**0.8933**	**0.0887**	**0.0840**	**0.9068**	**0.0014**

for downstream applications. These results indicate that our method consistently maintains high consistency in multi-view generation and high-quality 3D character generation across various conditions.

3) Semantic Components Generation: SeparateGen can generate semantically decomposed 3D character models from input images with arbitrary poses. Fig. 7 shows the corresponding 3D components of shoes, hair, body, and clothing generated from the input image. Despite challenges such as self-occlusion caused by the complexity of the input pose and inter-occlusion between the components into corresponding components while preserving the original spatial relationships of the input image. This allows for better adaptability to downstream tasks.

C. User Study

To evaluate the robustness of semantic component generation by SeparateGen, we conducted a user study using 10 generated 3D characters and 24 components from different character models. The study was carried out through an online questionnaire with 29 volunteers. For each question, volunteers were presented with the input image and the 2D renderings of the 3D meshes generated by different methods, where the display order was randomized to avoid bias. Participants were asked to assess the results based on geometry quality, texture quality, and consistency with the input images, where "consistency" was explicitly explained as the similarity of color and geometry to the input, i.e., whether the user perceives them as belonging to the same character. They were then asked to select the best result for each example. As shown in Table IV, SeparateGen outperformed other methods in character generation tasks, receiving significantly higher ratings.

Fig. 6. Comparison of 3D characters generated by our method and other methods using arbitrary-pose inputs. 🌐 denotes images collected from the Internet.

TABLE IV
USER STUDY VOTING RESULTS

Methods	SeparateGen	CharacterGen	Unique3d	InstantMesh
Geometry	**72.75%**	11.37%	13.10%	2.75%
Texture	**61.72%**	20.68%	14.82 %	2.75%
Consistency	**64.48%**	15.86%	15.17%	4.48%

Additionally, volunteers rated the semantic validity and overall quality of the generated components on a scale from 1 to 5, with higher scores indicating better performance. The components achieved scores of 4.53 and 4.32 for semantic validity and quality, respectively, demonstrating strong alignment with human expectations.

D. Ablation Study

1) Component Decomposition: To demonstrate the effectiveness of our component decomposition, we compared the characters composed of our semantic components with the directly generated full characters, as shown in Fig. 8(a). As illustrated, the monolithic character model suffers from structural issues during generation due to the lack of explicit semantic information. For example, the clothing and hair of the character are fused together, which can lead to errors in subsequent downstream tasks. In contrast, the character model composed of semantic components avoids such issues due to the clear semantic separation of each component, resulting in a more accurate model structure that is more suitable for downstream tasks.

2) Region-Based Plane Super-Resolution: Region-Based Plane Super-Resolution plays a crucial role in enhancing model details in our SC-LRM. In Fig. 8(b), we additionally show the results where plane SR was not used during the generation. The generated low-resolution components clearly exhibit defects in terms of detail.

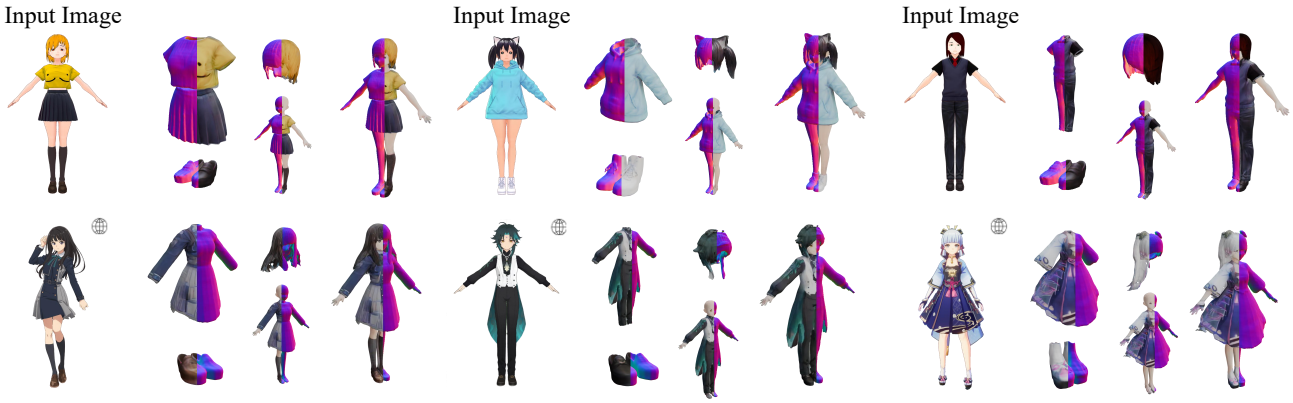

Fig. 7. Visualization of the components generated from the input image using our method, including clothing, shoes, hair, and body. In this visualization, the body is presented jointly with clothing rather than in isolation. 🌐 denotes images collected from the Internet.

TABLE V
ABLATION STUDIES ON THE DESIGNED MODULES

Methods	SSIM↑	LPIPS↓	FID↓	CLIP Score↑	CD↓
w/o Decompsed	0.8888	0.1004	0.0816	0.8933	0.0061
w/o PSR	0.8891	0.1039	0.0929	0.8821	0.0010
w/o Refinement	0.8918	0.0934	0.0804	0.9104	0.0010
2-Parts	0.9010	0.0880	0.0784	0.9019	0.0052
3-Parts	0.9057	0.0864	0.0791	0.9104	0.0037
Full	**0.9091**	**0.0743**	**0.0769**	**0.9147**	**0.0009**

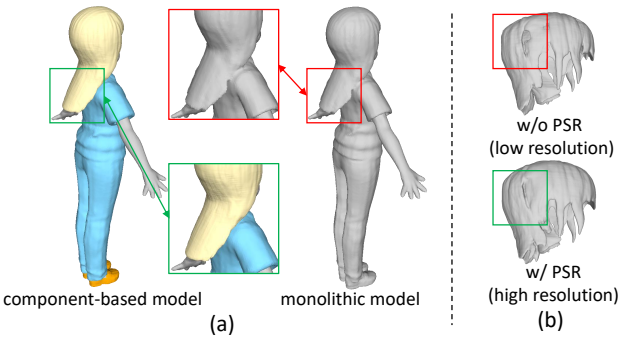

Fig. 8. (a) Direct generation of monolithic character models without semantic decomposition often results in incorrect mesh structures. (b) Component representation at a low feature-plane resolution (w/o PSR) may cause local defects, whereas applying Plane Super-Resolution (PSR) yields high-resolution components with improved geometric quality.

3) Post-Process Refinement: We present a comparison between the direct outputs of SC-LRM and the results after geometric refinement in Fig. 9, showcasing the results of each component before and after refinement. The results indicate that the direct outputs of SC-LRM exhibit grid-like artifacts. By leveraging the normal maps predicted by SC-LRM, the geometry can be further optimized to achieve higher precision and improved geometric quality.

4) Quantitative Analysis: We conducted a quantitative evaluation of the proposed design, with the results presented in Table V. The experimental findings indicate that each module contributes positively to the overall performance. More importantly, the study on semantic decomposition shows that increasing the number of components significantly improves

TABLE VI
COMPARISON OF 3D METHODS USING GROUND TRUTH MULTIVIEW IMAGES AS INPUT

Methods	SSIM↑	LPIPS↓	FID↓	CLIP Score↑	CD↓
Unique3D	0.9028	0.0874	0.0792	0.9030	0.0078
CharacterGen	0.8980	0.0890	0.0774	0.9046	0.0074
SeparateGen	**0.9199**	**0.0634**	**0.0669**	**0.9242**	**0.0009**

Fig. 9. Post-processing refinement can effectively address the grid-like artifacts in the coarse results of the model. Left: results before refinement; Right: results after refinement.

reconstruction quality. Compared with the monolithic setting (w/o Decomposition), 2-Parts and 3-Parts designs already reduce errors, and the full 4-Parts design achieves the best results, with a notable decrease in Chamfer Distance and consistent improvements across other metrics. This validates our design choice of decomposing the model into four semantic components. In addition, we evaluated Unique3D, CharacterGen, and the 3D stage of our proposed method using the same ground-truth multi-view images as input. The quantitative results in Table VI demonstrate the effectiveness of the proposed SC-LRM improvements. While ground-truth multi-view inputs improve the performance of all baseline methods, our approach consistently achieves the best results across the main metrics, further confirming its advantage in 3D generation tasks.

E. Applications

In Fig. 10, we present the results of assigning skeletons and performing rigging on the generated characters, which are then driven to perform diverse motions. This demonstrates the effectiveness of our component-based generation in supporting downstream tasks such as rigging and animation. Additionally, in Fig. 11, we showcase two 3D character models reusing components, demonstrating the potential of our method in 3D asset reuse and its ability to provide vast assets. We also constructed a scene using the character composed of generated components, as shown in Fig.12.

Fig. 10. Rigging and Animation Results of Generated Characters.

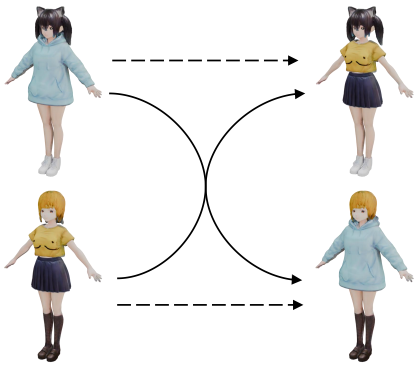

Fig. 11. 3D component asset exchange between two characters.

Fig. 12. A scene constructed with generated 3D Characters.

V. Conclusion

This paper proposes SeparateGen, a novel method for generating 3D characters composed of semantic components from a single image with arbitrary poses. The method first uses a multi-view generation model to convert an input image with an arbitrary pose into multi-view images with a standardized A-pose. Subsequently, we design SC-LRM, which reconstructs 3D character models composed of four semantic component. The method further enhances geometric details through post-processing refinement. Experimental results demonstrate that SeparateGen can generate high-quality 3D character models with semantic decomposition, making it well-suited for various downstream tasks.

Despite these promising results, our method still has several limitations. Specifically, it struggles when the character has overly complex structures or components beyond the four predefined semantic categories. In particular, characters with cloaks or similar garments are often inaccurately reconstructed, as their geometric features tend to overlap with hair. In addition, since our dataset mainly focuses on human-like anime characters, the component-based generation stage cannot directly handle non-human characters, although the non-component version of our model can still generate them.

For future work, we aim to extend the component-based design to a broader range of categories and improve the adaptability to non-human and structurally complex characters. Incorporating more advanced animation-driven generation techniques is also a promising direction to explore.

Acknowledgments

We sincerely appreciate all participants in the user study and thank Zi-Xin Zou and Jia-Hao Chen for helpful discussions.

References

[1] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *arXiv preprint arXiv:2209.14988*, 2022.

[2] 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," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[3] 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," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 300–309.

[4] Y. Hong, K. Zhang, J. Gu, S. Bi, Y. Zhou, D. Liu, F. Liu, K. Sunkavalli, T. Bui, and H. Tan, "Lrm: Large reconstruction model for single image to 3d," *arXiv preprint arXiv:2311.04400*, 2023.

[5] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," in *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 2023, pp. 851–866.

[6] 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," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 958–968.

[7] Y. Huang, H. Yi, Y. Xiu, T. Liao, J. Tang, D. Cai, and J. Thies, "Tech: Text-guided reconstruction of lifelike clothed humans," in *2024 International Conference on 3D Vision (3DV)*. IEEE, 2024, pp. 1531–1542.

[8] N. Kolotouros, T. Alldieck, A. Zanfir, E. Bazavan, M. Fieraru, and C. Sminchisescu, "Dreamhuman: Animatable 3d avatars from text," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[9] T. Liao, H. Yi, Y. Xiu, J. Tang, Y. Huang, J. Thies, and M. J. Black, "Tada! text to animatable digital avatars," in *2024 International Conference on 3D Vision (3DV)*. IEEE, 2024, pp. 1508–1519.

[10] H. Zhang, B. Chen, H. Yang, L. Qu, X. Wang, L. Chen, C. Long, F. Zhu, D. Du, and M. Zheng, "Avatarverse: High-quality & stable 3d avatar creation from text and pose," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 7, 2024, pp. 7124–7132.

[11] H.-Y. Peng, J.-P. Zhang, M.-H. Guo, Y.-P. Cao, and S.-M. Hu, "Charactergen: Efficient 3d character generation from single images with multi-view pose canonicalization," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 4, pp. 1–13, 2024.

[12] M. Liu, C. Zeng, X. Wei, R. Shi, L. Chen, C. Xu, M. Zhang, Z. Wang, X. Zhang, I. Liu *et al.*, "Meshformer: High-quality mesh generation with 3d-guided reconstruction model," *arXiv preprint arXiv:2408.10198*, 2024.

[13] Q.-C. Xu, T.-J. Mu, and Y.-L. Yang, "A survey of deep learning-based 3d shape generation," *Computational Visual Media*, vol. 9, no. 3, pp. 407–442, 2023.

[14] C. Wang, H.-Y. Peng, Y.-T. Liu, J. Gu, and S.-M. Hu, "Diffusion models for 3d generation: A survey," *Computational Visual Media*, vol. 11, no. 1, pp. 1–28, 2025.

[15] J. Tang, T. Wang, B. Zhang, T. Zhang, R. Yi, L. Ma, and D. Chen, "Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 22 819–22 829.

[16] A. Raj, S. Kaza, B. Poole, M. Niemeyer, N. Ruiz, B. Mildenhall, S. Zada, K. Aberman, M. Rubinstein, J. Barron *et al.*, "Dreambooth3d: Subject-driven text-to-3d generation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 2349–2359.

[17] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng, "Dreamgaussian: Generative gaussian splatting for efficient 3d content creation," *arXiv preprint arXiv:2309.16653*, 2023.

[18] 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," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 9298–9309.

[19] G. Qian, J. Mai, A. Hamdi, J. Ren, A. Siarohin, B. Li, H.-Y. Lee, I. Skorokhodov, P. Wonka, S. Tulyakov *et al.*, "Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors," *arXiv preprint arXiv:2306.17843*, 2023.

[20] P. Wang and Y. Shi, "Imagedream: Image-prompt multi-view diffusion for 3d generation," *arXiv preprint arXiv:2312.02201*, 2023.

[21] V. Voleti, C.-H. Yao, M. Boss, A. Letts, D. Pankratz, D. Tochilkin, C. Laforte, R. Rombach, and V. Jampani, "Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion," in *European Conference on Computer Vision*. Springer, 2024, pp. 439–457.

[22] Z. Chen, Y. Wang, F. Wang, Z. Wang, and H. Liu, "V3d: Video diffusion models are effective 3d generators," *arXiv preprint arXiv:2403.06738*, 2024.

[23] H. Yang, Y. Chen, Y. Pan, T. Yao, Z. Chen, C.-W. Ngo, and T. Mei, "Hi3d: Pursuing high-resolution image-to-3d generation with video diffusion models," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 6870–6879.

[24] J. Tang, Z. Chen, X. Chen, T. Wang, G. Zeng, and Z. Liu, "Lgm: Large multi-view gaussian model for high-resolution 3d content creation," in *European Conference on Computer Vision*. Springer, 2025, pp. 1–18.

[25] Z. Wang, Y. Wang, Y. Chen, C. Xiang, S. Chen, D. Yu, C. Li, H. Su, and J. Zhu, "Crm: Single image to 3d textured mesh with convolutional reconstruction model," in *European Conference on Computer Vision*. Springer, 2025, pp. 57–74.

[26] J. Xu, W. Cheng, Y. Gao, X. Wang, S. Gao, and Y. Shan, "Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models," *arXiv preprint arXiv:2404.07191*, 2024.

[27] S. Li, C. Li, W. Zhu, B. Yu, Y. Zhao, C. Wan, H. You, H. Shi, and Y. Lin, "Instant-3d: Instant neural radiance field training towards on-device ar/vr 3d reconstruction," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, 2023, pp. 1–13.

[28] S. Wu, Y. Lin, F. Zhang, Y. Zeng, J. Xu, P. Torr, X. Cao, and Y. Yao, "Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer," *arXiv preprint arXiv:2405.14832*, 2024.

[29] L. Zhang, Z. Wang, Q. Zhang, Q. Qiu, A. Pang, H. Jiang, W. Yang, L. Xu, and J. Yu, "Clay: A controllable large-scale generative model for creating high-quality 3d assets," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 4, pp. 1–20, 2024.

[30] W. Li, J. Liu, R. Chen, Y. Liang, X. Chen, P. Tan, and X. Long, "Craftsman: High-fidelity mesh generation with 3d native generation and interactive geometry refiner," *arXiv preprint arXiv:2405.14979*, 2024.

[31] P. Li, Y. Liu, X. Long, F. Zhang, C. Lin, M. Li, X. Qi, S. Zhang, W. Xue, W. Luo *et al.*, "Era3d: high-resolution multiview diffusion using efficient row-wise attention," *Advances in Neural Information Processing Systems*, vol. 37, pp. 55 975–56 000, 2024.

[32] Q. Wu, X. Liu, Y. Chen, K. Li, C. Zheng, J. Cai, and J. Zheng, "Object-compositional neural implicit surfaces," in *European Conference on Computer Vision*. Springer, 2022, pp. 197–213.

[33] Q. Wu, K. Wang, K. Li, J. Zheng, and J. Cai, "Objectsdf++: Improved object-compositional neural implicit surfaces," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 21 764–21 774.

[34] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 165–174.

[35] Y. Feng, W. Liu, T. Bolkart, J. Yang, M. Pollefeys, and M. J. Black, "Learning disentangled avatars with hybrid 3d representations," *arXiv preprint arXiv:2309.06441*, 2023.

[36] S. Lin, Z. Li, Z. Su, Z. Zheng, H. Zhang, and Y. Liu, "Layga: Layered gaussian avatars for animatable clothing transfer," in *ACM SIGGRAPH 2024 Conference Papers*, 2024, pp. 1–11.

[37] T. Kim, S. Saito, and H. Joo, "Ncho: Unsupervised learning for neural 3d composition of humans and objects," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 14 817–14 828.

[38] J. Dong, Q. Fang, Z. Huang, X. Xu, J. Wang, S. Peng, and B. Dai, "Tela: Text to layer-wise 3d clothed human generation," in *European Conference on Computer Vision*. Springer, 2025, pp. 19–36.

[39] 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," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[40] A. Liu, C. Lin, Y. Liu, X. Long, Z. Dou, H.-X. Guo, P. Luo, and W. Wang, "Part123: part-aware 3d reconstruction from a single-view image," in *ACM SIGGRAPH 2024 Conference Papers*, 2024, pp. 1–12.

[41] H. Han, R. Yang, H. Liao, J. Xing, Z. Xu, X. Yu, J. Zha, X. Li, and W. Li, "Reparo: Compositional 3d assets generation with differentiable 3d layout alignment," *arXiv preprint arXiv:2405.18525*, 2024.

[42] H. Yan, Y. Li, Z. Wu, S. Chen, W. Sun, T. Shang, W. Liu, T. Chen, X. Dai, C. Ma *et al.*, "Frankenstein: Generating semantic-compositional 3d scenes in one tri-plane," in *SIGGRAPH Asia 2024 Conference Papers*, 2024, pp. 1–11.

[43] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3d hands, face, and body from a single image," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 10 975–10 985.

[44] B. Jiang, J. Zhang, Y. Hong, J. Luo, L. Liu, and H. Bao, "Bcnet: Learning body and cloth shape from a single image," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*. Springer, 2020, pp. 18–35.

[45] A. Aggarwal, J. Wang, S. Hogue, S. Ni, M. Budagavi, and X. Guo, "Layered-garment net: Generating multiple implicit garment layers from a single image," in *Proceedings of the Asian Conference on Computer Vision*, 2022, pp. 3000–3017.

[46] Y. Xiu, J. Yang, D. Tzionas, and M. J. Black, "Icon: Implicit clothed humans obtained from normals. in 2022 ieee," in *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 13 286–13 296.

[47] Y. Xiu, J. Yang, X. Cao, D. Tzionas, and M. J. Black, "Econ: Explicit clothed humans optimized via normal integration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 512–523.

[48] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2304–2314.

[49] S. Saito, T. Simon, J. Saragih, and H. Joo, "Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 84–93.

[50] T. He, J. Collomosse, H. Jin, and S. Soatto, "Geo-pifu: Geometry and pixel aligned implicit functions for single-view human reconstruction," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9276–9287, 2020.

[51] Y. Li, M. Tang, Y. Yang, R. Tong, S. Yang, Y. Li, B. An, and Q. Kou, "Ctsn: Predicting cloth deformation for skeleton-based characters with a two-stream skinning network," *Computational Visual Media*, vol. 10, no. 3, pp. 471–485, 2024.

[52] M. Shi, W. Feng, L. Gao, and D. Zhu, "Generating diverse clothed 3d human animations via a generative model," *Computational Visual Media*, vol. 10, no. 2, pp. 261–277, 2024.

[53] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3836–3847.

[54] J. Wang, Y. Liu, Z. Dou, Z. Yu, Y. Liang, C. Lin, R. Xie, L. Song, X. Li, and W. Wang, "Disentangled clothed avatar generation from text descriptions," in *European Conference on Computer Vision*. Springer, 2025, pp. 381–401.

[55] J. Zhang, X. Zeng, X. Chen, W. Zuo, G. Yu, and Z. Tu, "Mikudance: Animating character art with mixed motion dynamics," *arXiv preprint arXiv:2411.08656*, 2024.

[56] L. Hu, "Animate anyone: Consistent and controllable image-to-video synthesis for character animation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 8153–8163.

[57] Z. Huang, Y.-C. Guo, H. Wang, R. Yi, L. Ma, Y.-P. Cao, and L. Sheng, "Mv-adapter: Multi-view consistent image generation made easy," *arXiv preprint arXiv:2412.03632*, 2024.

[58] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *Seminal graphics: pioneering efforts that shaped the field*, 1998, pp. 347–353.

[59] S. Laine, J. Hellsten, T. Karras, Y. Seol, J. Lehtinen, and T. Aila, "Modular primitives for high-performance differentiable rendering," *ACM Transactions on Graphics (ToG)*, vol. 39, no. 6, pp. 1–14, 2020.

[60] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, "Objaverse: A universe of annotated 3d objects," in *Proceedings of the IEEE/CVF*

conference on computer vision and pattern recognition, 2023, pp. 13 142–13 153.
[61] R. Wu and C. Zheng, "Learning to generate 3d shapes from a single example," *arXiv preprint arXiv:2208.02946*, 2022.
[62] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.
[63] G. M. Nielson, "Dual marching cubes," in *IEEE visualization 2004*. IEEE, 2004, pp. 489–496.
[64] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi, "Efficiently combining positions and normals for precise 3d geometry," *ACM transactions on graphics (TOG)*, vol. 24, no. 3, pp. 536–543, 2005.
[65] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.
[66] T. Wu, J. Zhang, X. Fu, Y. Wang, J. Ren, L. Pan, W. Wu, L. Yang, J. Wang, C. Qian *et al.*, "Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 803–814.
[67] VRoid, "Vroid hub," 2022. [Online]. Available: https://vroid.com/
[68] VRM, "Vrm specification," 2022. [Online]. Available: https://vrm.dev/
[69] Blender Online Community, "Blender - a 3d modelling and rendering package," 2024, version 4.1, Available at: https://www.blender.org/.
[70] MixamoInc., "Mixamo's online services," 2009. [Online]. Available: https://www.mixamo.com/
[71] G. Lin, L. Yang, C. Zhang, H. Pan, Y. Ping, G. Wei, T. Komura, J. Keyser, and W. Wang, "Patch-grid: an efficient and feature-preserving neural implicit surface representation," *ACM Transactions on Graphics*, vol. 44, no. 2, pp. 1–21, 2025.
[72] W. Chen, C. Lin, W. Li, and B. Yang, "3psdf: Three-pole signed distance function for learning surfaces with arbitrary topologies," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 18 522–18 531.
[73] P.-S. Wang, Y. Liu, and X. Tong, "Dual octree graph networks for learning adaptive volumetric shape representations," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.
[74] J. Chibane, G. Pons-Moll *et al.*, "Neural unsigned distance fields for implicit function learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 638–21 652, 2020.
[75] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach, "Sdxl: Improving latent diffusion models for high-resolution image synthesis," *arXiv preprint arXiv:2307.01952*, 2023.
[76] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
[77] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.
[78] Y. Liu, C. Lin, Z. Zeng, X. Long, L. Liu, T. Komura, and W. Wang, "Syncdreamer: Generating multiview-consistent images from a single-view image," *arXiv preprint arXiv:2309.03453*, 2023.
[79] K. Wu, F. Liu, Z. Cai, R. Yan, H. Wang, Y. Hu, Y. Duan, and K. Ma, "Unique3d: High-quality and efficient 3d mesh generation from a single image," *arXiv preprint arXiv:2405.20343*, 2024.

Dong-Yang Li received the B.S. degree in the Department of Computer Science and Technology from Tsinghua University, Beijing, China, in 2023. He is currently working toward the Ph.D. degree in the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include 3D generation, 3D reconstruction.

Yi-Long Liu is an undergraduate student at the Department of Computer Science and Technology, Tsinghua University. His research interests include computer vision, computer graphics and deep learning.

Zi-Xian Liu is an undergraduate student at the Institute for Interdisciplinary Information Sciences, Tsinghua University. His research interests include computer vision and computer graphics.

Yan-Pei Cao received his bachler and Ph.D. degrees in Computer Science from Tsinghua University in 2013 and 2018, respectively. He is currently the Co-founder and Chief Scientist at VAST. His research interests include Computer Graphics, Generative AI, and 3D Computer Vision.

Meng-Hao Guo is a fifth-year Ph.D candidate under the supervision of Prof. Shi-Min Hu in the Department of Computer Science and Technology, Tsinghua University. His research interests include computer vision, deep learning and computer graphics. He has published papers in some journals and conferences, including IEEE TPAMI, ACM TOG, NeurIPS, CVPR, ICLR *etc*. He also serves as reviewer for some journals and conferences such as IEEE TPAMI, IJCV, IEEE TIP, CVPR, ICCV, NeurIPS, ICLR, ICML, *etc*.

Shi-Min Hu (Fellow, IEEE) received his Ph.D. degree from Zhejiang University in 1996. He is currently a Professor in the Department of Computer Science and Technology at Tsinghua University, Beijing, China. His research interests span digital geometry processing, video processing, rendering, computer animation, and computer-aided geometric design. He has authored over 100 papers published in leading journals and refereed conferences. Dr. Hu serves as the Editor-in-Chief of Computational Visual Media and is on the editorial boards of several prestigious journals, including Computer Aided Design and Fundamental Research. He is a Senior Member of the Association for Computing Machinery (ACM) and a Fellow of both the China Computer Federation (CCF) and the Solid Modeling Association (SMA).