StdGEN: Semantic-Decomposed 3D Character Generation from Single Images

Supplementary Material

A. Ablation Study

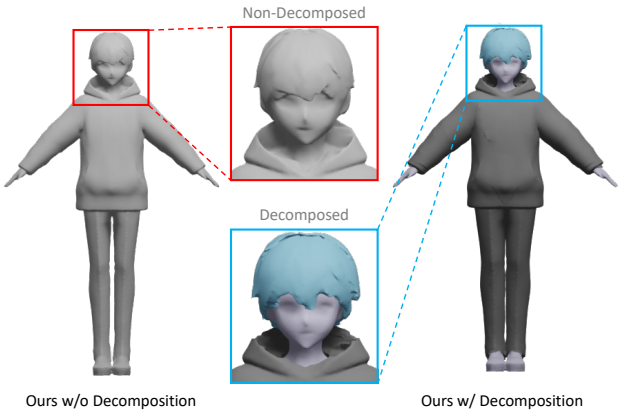

Figure 1. Ablation study on character decomposition.

Character Decomposition. To demonstrate the decomposition capabilities of our core S-LRM and its impact on the results, we compared our method with a direct refinement approach that does not employ semantic decomposition. The visual comparison in Fig. 1 reveals that without decomposition, the results exhibit a fusion of hair, clothing, and the base human model, significantly limiting their potential for downstream applications. In contrast, our method successfully separates these components while maintaining high mesh precision, showcasing the effectiveness of our semantic decomposing approach.

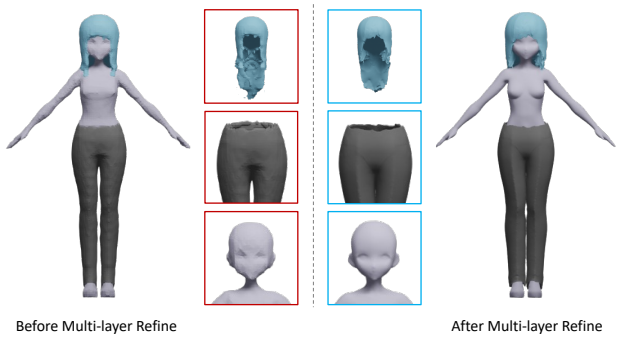

Figure 2. Ablation study on multi-layer refinement. Zoom in for better details.

Multi-layer Refinement. We further illustrate the distinction between the direct output of our S-LRM and the results after multi-layer refinement in Fig. 2. The pre-optimization results demonstrate that our S-LRM successfully decomposes various mesh components with correct geometry and shape, validating the capabilities of our S-LRM. However, the precision is limited due to the inherent characteristics of FlexiCubes [12] and memory constraints. Post-refinement, we observe a substantial enhancement in precision while maintaining the overall structure. This improvement underscores the effectiveness of our multi-layer refinement process in preserving the structure of the decomposed components while significantly elevating the geometric accuracy and overall quality of the reconstructed character.

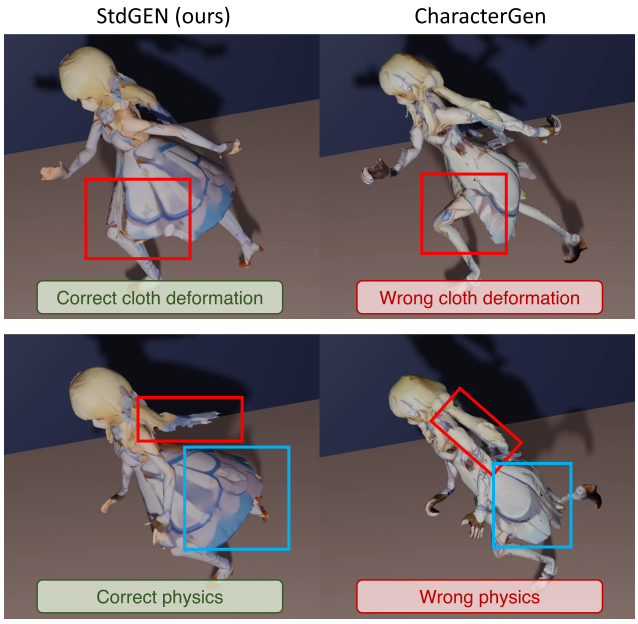

Figure 3. Rigging and animation comparisons on 3D character generation. Our method demonstrates superior performance in human perception and physical characteristics.

B. More Applications

Compared to other 3D character generation methods, our decomposed generation in A-pose is more suitable for downstream animation and 3D applications. In Fig. 3, we rig the 3D character generated by our approach and by CharacterGen [9] for comparison. Without decomposition, the hair and clothing stick together and are attached to the base human model. In contrast, our approach maintains separated parts, aligning more closely with natural perception. Additionally, the non-decomposed nature leads to inaccurate deformations and physical characteristics during movement, which our method effectively avoids.

C. Time Breakdown Analysis

Process	Time (s)
Canonicalization Diffusion	7
Multi-view Diffusion	29
S-LRM	12
Refinement	
- Single-layer Settings	18
- Multi-layer Settings	117

Table 1. Time breakdown for each processing step.

In Tab. 1, we present the time breakdown of different components in our method. Creating a single-layered 3D character takes only about 1 minute while generating a decomposed, multi-layered 3D character requires less than 3 minutes. Our S-LRM is relatively efficient, with minimal additional time and memory overhead compared to InstantMesh's LRM.

D. More Quantitative Results

3D Semantic Metrics. To demonstrate our 3D semantic decomposition capability, we extracted separate meshes for three semantic categories (hair, cloth, and base human model) and rendered masks from eight different views for comparison with ground truth. In the arbitrary pose setting, we achieved IoU scores of 0.73 for hair, 0.86 for cloth, and 0.88 for the base human model. These results demonstrate effective semantic decomposition, particularly considering the significant challenges in single-image-based 3D reconstruction, such as spatial ambiguity and occlusion.

3D Geometric Metrics. Additional comparisons in the arbitrary pose setting are presented in Tab. 2, where our method demonstrates superior performance on both the Chamfer distance and the F-score, showing a more precise prediction of 3D geometry.

Metric	Unique3D	CharacterGen	Ours
Chamfer Distance ↓	0.109	0.035	**0.023**
F-Score ↑	0.137	0.465	**0.664**

Table 2. 3D metric comparison (0.01 for f-score threshold).

Ablation Study on Refinement Stage. We conduct experiments without refinement in Tab. 3 under arbitrary pose setting, where we only apply color back-projection to the mesh generated by S-LRM. The results indicate that our method outperforms CharacterGen and Unique3D even without refinement, demonstrating that our S-LRM is effective even without the refinement step.

Method	SSIM ↑	LPIPS ↓	FID ↓	CLIP Similarity ↑
Unique3D	0.856	0.190	0.042	0.903
CharacterGen	0.869	0.134	0.119	0.901
Ours (w/o refine)	0.912	0.094	0.026	0.933
Ours	**0.916**	**0.084**	**0.011**	**0.936**

Table 3. Ablation study on refinement stage.

E. Dataset Construction

Semantic Definition. Unlike general 3D models, 3D character modeling typically involves multiple components rather than a single entity. This segmentation is essential for downstream applications such as rigging and physical simulation, which often require the manipulation of distinct parts. Conventional reconstruction models only capture the character's surface appearance, lacking internal information and the ability to decompose the model, which limitation severely restricts subsequent applications. After considering both practical applications and data composition, we have categorized character composition into three semantic categories: base minimal-clothed human model, clothing, and hair (specifically, shoes and underwear are classified as part of the base human model, considering downstream applications and data characteristics). By incorporating these semantic categories into our reconstruction process, we aim to produce 3D character models that are not only visually accurate but also functionally versatile for various uses in 3D game and animation pipelines. Note that our method supports the learning and extracting of an arbitrary number of semantic categories.

Data Cleaning. We begin by filtering out data that cannot be layered according to semantic structure. Using multiple prompts combined with ImageReward [15], we remove low-quality or malicious data with low scores. Additionally, we identify instances where semantic information predictions are inaccurate, then manually review and remove data that appears semantically incorrect to human perception. Since the base human model in the original dataset can occasionally contain defects, we apply a connectivity check on the front rendering of every base human model. Examples lacking connectivity are only used to supervise either the complete model or the base human model with clothing, but not the base human model alone.

Rendering Settings. Our rendering process goes beyond standard image generation, incorporating depth, normal, and semantic maps. 3D Character models are adjusted to an A-pose configuration, with arms rotated 45 degrees downward from the horizontal position. To facilitate multi-layered reconstruction supervision, we rendered the complete model and two additional configurations: the base minimal-clothed human model alone, and the base model with clothing. The rendering includes eight views at 45-degree azimuth intervals with zero elevation, supplemented

by top-down and bottom-up views. We enriched the dataset with five close-up facial views and 20 random viewpoints. To enhance the training of our diffusion model, we implemented data augmentation on varying arm angles.

We use the orthographic camera for all renderings. For non-close-up views of the character, after normalizing the character to fit within a unit cube, we set the ortho scale to 1.2. For close-up views of the face, we locate the 3D center position and bounding box of the facial semantics, aligning the camera's center projection with the 3D center of the face and setting the ortho scale to 1.2 times the bounding box size. We render five facial close-up views at elevation = 0° and azimuth angles of $\{-90°, -45°, 0°, 45°, 90°\}$. For inputs to the canonicalization diffusion model, we add outline and shading with a 50% probability. Rim lighting, shading, and outlines (except on the face) are consistently removed to supervise diffusion and S-LRM outputs. Semantic maps are rendered by modifying non-transparent regions of the texture map assigned to specific semantic parts.

Multi-layer Settings. We provide three different rendering levels: the complete model, the base human model only, and the base human model with clothes, each generated by selectively removing specific semantic elements. For supervising S-LRM, these correspond to (1) no semantic masking, (2) masking of hair and clothing, and (3) masking of hair only. By mixing these levels of 2D supervision, we can train S-LRM to reconstruct multi-layered density, color, and semantic information automatically.

F. Details of Loss Functions

In this section, we provide a detailed description of each loss component in our framework. $\mathcal{L}_{\text{mse}}$ is the commonly used mean squared error loss defined as:

$$\mathcal{L}_{\text{mse}} = \sum_k \left\| \hat{I}_k - I_k^{gt} \right\|_2^2 \tag{1}$$

where $\hat{I}_k$, I_k^{gt} denotes the k-th view of rendered images and ground-truth images, respectively.

$\mathcal{L}_{\text{lpips}}$ is the perceptual loss defined as

$$\mathcal{L}_{\text{lpips}} = \sum_k \tau \left(\phi(\hat{I}_k), \phi(I_k^{gt}) \right) \tag{2}$$

where ϕ is the VGG feature extractor, τ transforms deep embedding to a scalar LPIPS score.

$\mathcal{L}_{\text{mask}}$ is the mask loss defined as

$$\mathcal{L}_{\text{mask}} = \sum_k \left\| \hat{M}_k - M_k^{gt} \right\|_2^2 \tag{3}$$

where $\hat{M}_k$ and M_k^{gt} denote the rendered non-transparent mask, and ground-truth masks, respectively.

The deviation loss $\mathcal{L}_{\text{dev}}$ penalizes the Euclidean distances between each dual vertex v and the edge crossings $u_e \in \mathcal{N}_v$ that bound its primal face, encourages vertices to center within their cells and allowing flexibility for connectivity adaptation:

$$\mathcal{L}_{\text{dev}} = \sum_{v \in V} \text{MAD}[\{|v - u_e|_2 \ : \ u_e \in \mathcal{N}_v\}] \tag{4}$$

where $|\cdot|_2$ is the Euclidean distance, $\text{MAD}(Y) = \frac{1}{|Y|} \sum_{y \in Y} |y - \text{mean}(Y)|$ denotes the mean absolute deviation. We use the same approach as the implementation of InstantMesh [14], applying L2 regularization with a weight of 0.1 to the FlexiCubes weights.

During S-LRM training, we specifically incorporated facial semantics as an additional component to enhance the training process and facilitate potential applications. In subsequent stages, facial semantics were treated as an integral part of the base human model. For the semantic cross-entropy loss $\mathcal{L}_{\text{sem}}$, we empirically assigned weights to four semantic categories - hair (1.255), face (1.758), base human model (0.913), and cloth (0.650) - based on their respective rendering proportions in the dataset to optimize semantic learning.

G. Implementation Details

We divide our Anime3D++ dataset into a training and testing set in a 99:1 ratio. We first train the canonicalization diffusion model at a 512 resolution with a learning rate of 5e-5, then reduce it to 1e-5 as we progressively increase the resolution to 768 and 1024. Similarly, the multi-view diffusion model is trained at a constant learning rate of 5e-5 while scaling from 512 to 768 and finally to 1024 resolution. We use LoRA with a 128-rank for S-LRM, a learning rate of 4e-5, and three supervision stages with rendering resolutions of 192, 144, and 512. The loss function parameters are set as $\lambda_{\text{lpips}}, \lambda_{\text{mask}}, \lambda_{\text{sem}}, \lambda_{\text{depth}}, \lambda_{\text{normal}}, \lambda_{\text{dev}} = 2.0, 1.0, 1.0, 0.5, 0.2, 0.5$. For multi-layer refinement, we set $\lambda'_{\text{mask}}, \lambda'_{\text{normal}}, \lambda_{\text{col}} = 1.0, 1.0, 100.0$, and we further extract the coarse hair mask, applying additional normal and mask loss for hair refinement with a weight of 1 and 10.

Detailed Structure of S-LRM. Following InstantMesh [14], our S-LRM adopts 6 RGB images generated by multi-view diffusion in a resolution of 320×320 as model input. In the training stage 3 (training on meshes with multi-layer semantics), we set the sampling grid size for FlexiCubes extraction to $100 \times 100 \times 150$, with dimensions scaled to $0.7 \times 0.7 \times 1.05$ of the triplane size, and the centers of both are aligned. We integrate the LoRA [4] structure into the S-LRM transformer, modifying both the self-attention and cross-attention modules. For self-attention, where q, k, and v values are produced by shared linear layers, we substitute all input and output

linear layers with LoRA structures. In cross-attention, where q, k, and v are produced through separate linear layers, we replace the linear layers for q, k, v, and the outputs with LoRA structures. The specifics are as follows:

$$h^i = W_0^i + \Delta W_{tp}^i x = W_0^i x + B_{tp}^i A_{tp}^i x \tag{5}$$

Here, i denotes the i-th transformer layer. In self-attention, tp represents the linear projection for inputs and outputs, while in cross-attention, tp denotes the linear projections for q, k, v, and outputs. During the training process, the DINO [1] encoder is kept frozen while the feature decoder (including color/density decoder and semantic decoder) remains trainable. In the triplane transformer, the positional embeddings, de-convolution layers, and all LoRA layers are set as learnable parameters, while all other layers are frozen.

Detailed Settings of Canonicalization Diffusion. Our canonicalization diffusion model comprises a U-Net and a ReferenceNet with an identical architecture, both networks are initialized with the weights from Stable Diffusion 2.1. The U-Net takes the CLIP-encoded features of the reference image as input for the encoder-hidden states. ReferenceNet, on the other hand, receives the image latents of the reference image (encoded by a VAE without added noise) as input, along with the features of a fixed text prompt, "high quality, best quality," encoded by CLIP [11], which are fed into the encoder-hidden states. A cross-attention operation is applied for each corresponding layer pair in the U-Net and ReferenceNet, using the current U-Net layer as the query and the corresponding ReferenceNet layer as the key and value. This cross-attention mechanism transfers the detailed information of the reference image into the U-Net.

Detailed Settings of Multi-View Diffusion. Building upon Era3D's [8] multi-view model, we start training from its inherited weights. We concatenate the noisy VAE latent and reference image VAE latent as input to the U-Net. For each view's color and normal output, we specify fixed prompts ("a rendering image of 3D models, {view} view, {color/normal} map"), which are encoded through CLIP and fed into the U-Net's encoder hidden states. For U-Net's class labels, we reserve the first 1024 dimensions for the CLIP embedding of the reference image, and replace the noise level embedding in the latter 1024 dimensions with a level switcher. This level switcher uses different one-hot vectors for three distinct rendering levels to support the specific semantic combinations in the diffusion output, serving as supervision signals for multi-layer refinement. Since the previous canonicalization diffusion step already ensures that the output A-pose character reference image has elevation=0 and is orthographic, we do not employ the regression loss from Era3D. Additionally, we fix the noise level of the image VAE latent to 0 to achieve optimal fidelity.

Details of Color Back-projection. We employ a multi-view projection method similar to Unique3D [13] to back-project the texture onto the 3D character model. For each vertex v that is visible in at least one view, we calculate its final color $C(v)$ in the 3D mesh using the following formula:

$$C(v) = \sum_{i \in I} \frac{w_i (\mathbf{n}_v \cdot \mathbf{d}_i)^2 c_{v,i}}{w_i (\mathbf{n}_v \cdot \mathbf{d}_i)^2} \tag{6}$$

where $c_{v,i}$ represents the color corresponding to v in the i-th view texture; $\mathbf{n}_v$ and $\mathbf{n}_i$ is the vertex normal of v and the view direction of the i-th view respectively; w_i is the projection weight of the i-th view, and I is the set of views where v is visible. In practice, we assign w_i values of $\{2.0, 0.5, 0.0, 1.0, 0.0, 0.5\}$ for views at azimuths of $\{0°, 45°, 90°, 180°, 270°, 315°\}$ respectively. For vertices that are not visible in any view, we treat the 3D mesh as a graph composed of vertices and edges, and iteratively perform convolution and mean pooling to transfer colors from vertices with determined colors to those without, until all vertices obtain their colors.

Pre- and Post-dilation of Mesh. To better solve the problematic intersections among meshes in the multi-layer refinement stage, we introduce a "dilation" process applied both before and after optimization. This process constructs an approximate "flow field" based on the original positions of the inner and outer layer meshes.

For each vertex on the outside mesh, we utilize a kd-tree to query its nearest vertex neighbors of the fixed inside mesh. The movement range is then smoothly weighted based on the exponential inverse distance from these neighbors, with distant points remaining stationary. This approach creates a "dilation" effect, ensuring that when inner layers are moved outward to resolve intersections, the outer layers follow suit in a natural, gradual manner.

H. Discussions

Comparison with other decomposition methods. We note that some works have also applied the concept of decomposition, while they differ from our method in problem definition and scope. GALA [5] and TELA [3] use real-world 3D mesh scans and text as input respectively, employing Score Distillation Sampling (SDS) [10] with class-specific text prompts and pose control for layered 3D avatar generation. Frankenstein [16] takes a textureless, 2D semantic layout as input and generates semantic-decomposed 3D meshes (also textureless) through triplane diffusion. In contrast, our method accepts RGB reference images of arbitrary characters and generates 3D character meshes that faithfully preserve the reference texture while enabling semantic decomposition in a feed-forward manner.

Regarding specific decomposition techniques, GALA and TELA use SDS and different prompts to optimize both individual parts and the whole iteratively, typically requiring hours or more for a single case; Frankenstein outputs

separate SDFs for each semantic class, training and inferring on datasets with specific semantics, while our method treats geometry and semantics information independently, enabling greater compatibility and scalability. Our method can extract equivalent surfaces by specifying any combination of semantics, while maintaining compatibility with general datasets like Objaverse [2] and preserving LRM's general performance. It also has the potential to achieve semantic decomposition for multiple data types through a single LRM paired with multiple semantic decoders. Moreover, when adding new semantic classes, our approach can inherit geometric priors, making fine-tuning more efficient.

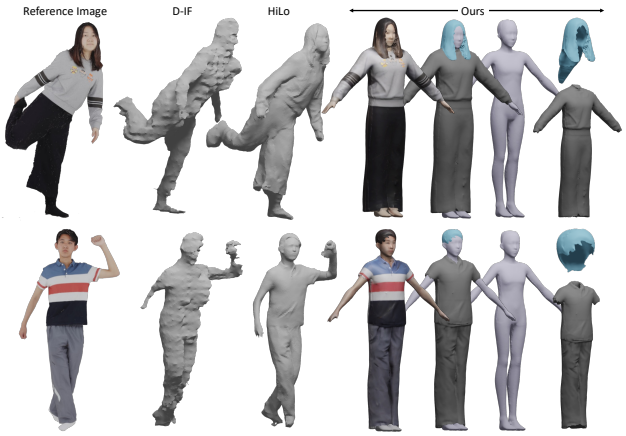

Figure 4. Comparison on THuman 2.0 dataset.

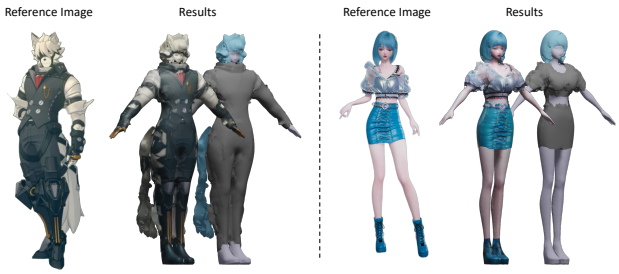

Figure 5. Our result on furry and 2.5D style images.

Non-anime style results. Our method demonstrates generalization across diverse character types, as shown in Figs. 4 and 5. For realistic style, we compare with D-IF [17] and HiLo [18] on THuman 2.0 dataset [19]. While the results show slight stylistic bias inherently from the Anime3D++ training data (e.g., slim faces), our method is general with robustness, canonicalization, and decomposition capabilities even without real-human training data. To further show our method's 3D editing and decomposing ability, we also directly compare the editing case in AvatarPopUp [7] (Fig. 6). Our approach shows similar effectiveness on real-human examples as AvatarPopUp and offers semantic decomposition and style flexibility capabilities.

Figure 6. 3D editing comparison with AvatarPopUp.

Semantic definition and possible improvements. Our Anime3D++ dataset adopts the VRoid-Hub data standard, which is designed to align with VR/game requirements, particularly for animation and collision detection. Following this standard, close-fitting garments are classified as part of base human model, while outerwear (e.g., pants, skirts, hoodies with long sleeves) is categorized as clothing. Although our results currently support relatively few semantic categories due to the limitations in datasets, our method is general and the results demonstrate the feasibility of semantic awareness generation. In future work, our framework can be easily extended to support fine-grained semantic decomposition by incorporating Segment Anything Model (SAM) [6] to generate detailed semantic labels for S-LRM training.

Design choice of Semantic-Equivalent SDF. Here we discuss why we don't assign a dedicated decoder for each semantic class to predict their SDFs separately. First, this approach would not effectively utilize the prior knowledge from the NeRF training stage. The SDF information predicted by these decoders would differ significantly from what was learned in the previous stage, with each decoder only retaining "its own" portion of the whole original SDF. The semantic information learned during the last stage would also become unusable. In contrast, our method almost completely inherits the prior knowledge from the NeRF training stage and smoothly transitions to the SDF stage without any modifications to the network architecture.

Additionally, this alternative approach would suffer from poor scalability - adding a new semantic class would require adding and training a new triplane feature decoder nearly from scratch and modifying the network structure, while our method does not require relearning geometric information when adding or removing semantics. It would also increase computational and memory costs during both training and inference. Furthermore, without cross-semantic constraints, the surfaces extracted from SDFs of different semantic classes might intersect, which contradicts our requirements. Another issue is that such a representation would not be unified - we could only extract a surface for each individual semantic class, but not for the entire char-

Figure 7. More visualizations on semantic-decomposed 3D generations.

acter or multiple selected semantic classes simultaneously. In contrast, our solution can extract equivalent surfaces for any combination of selected semantic classes with only one decoder employed.

Limitations. We note several limitations that leave room for future work: (1) Following InstantMesh [14], our S-LRM produces triplanes with the resolution of 64×64. After switching to SDF training, the FlexiCubes sampling uses a grid size of only 150 height and 100 width. This resolution may constrain and limit further improvement in the results. (2) While our pipeline enables high-quality generation by the high-resolution diffusion output up to a resolution of 1024, this also slows the overall generation speed, presenting a trade-off. Our S-LRM requires only about 10 seconds for one inference, with the majority of time consumed in the diffusion and refinement, suggesting room for further optimization. (3) The restricted style and category diversity in the training data affects its ability to handle inputs that deviate significantly from the human-centric categories (e.g., animals or general 3D objects). Despite such challenges, our framework is extensible, allowing further improvements through tools like SAM for semantic labeling or SMPL label transfer for human datasets.

I. More Visualizations

We demonstrate more visualizations of semantic-decomposed 3D results (Fig. 7), outputs of canonicalization (Fig. 9) and multi-view (Fig. 8) diffusion model, comparisons with other methods (Fig. 10, 11) and multi-view renderings (Fig. 12).

Figure 8. Visualizations of the 2D multi-view diffusion model results.

Figure 9. Visualizations of the 2D canonicalization diffusion model results.

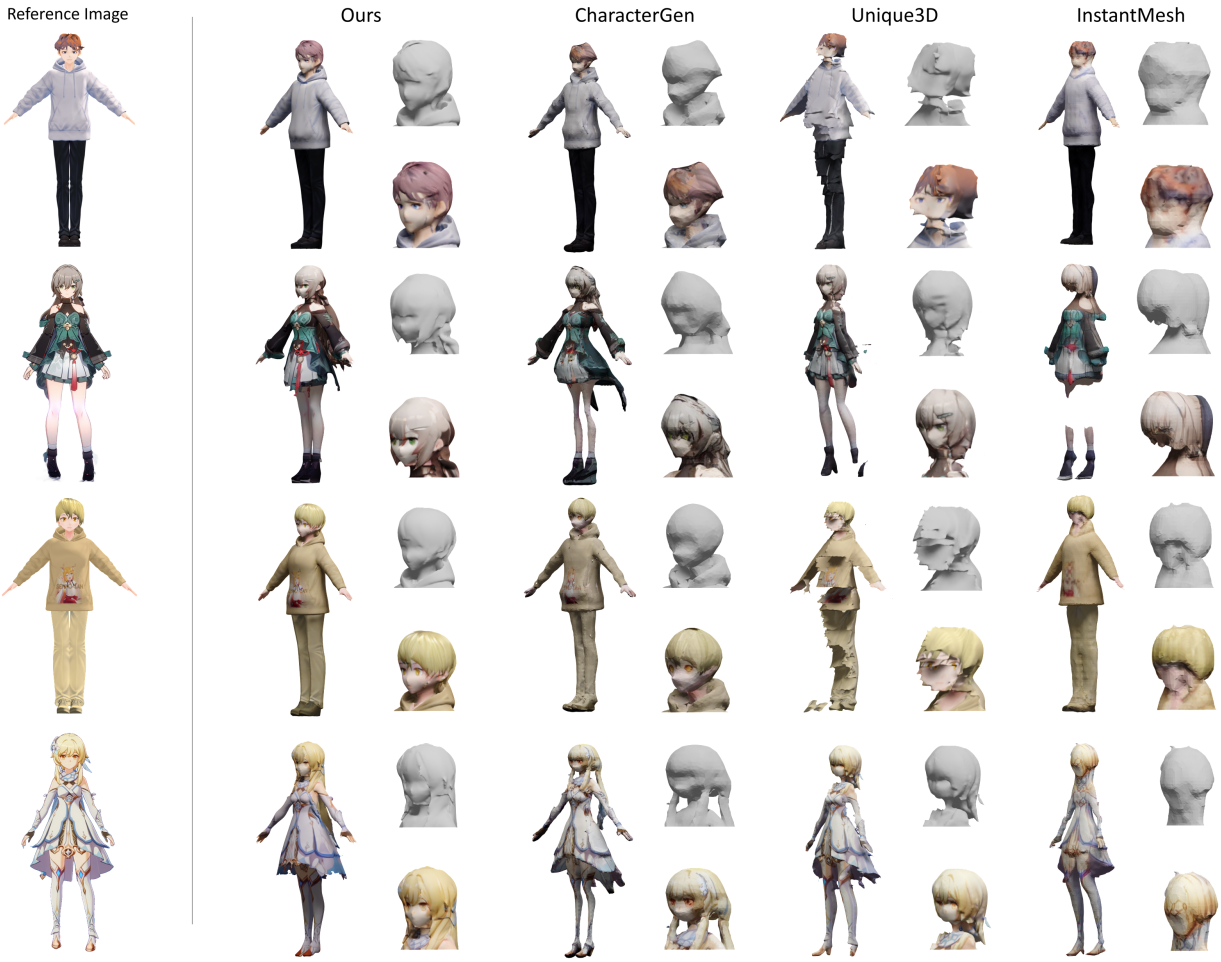

Figure 10. More qualitative comparisons of 3D character generations (#1).

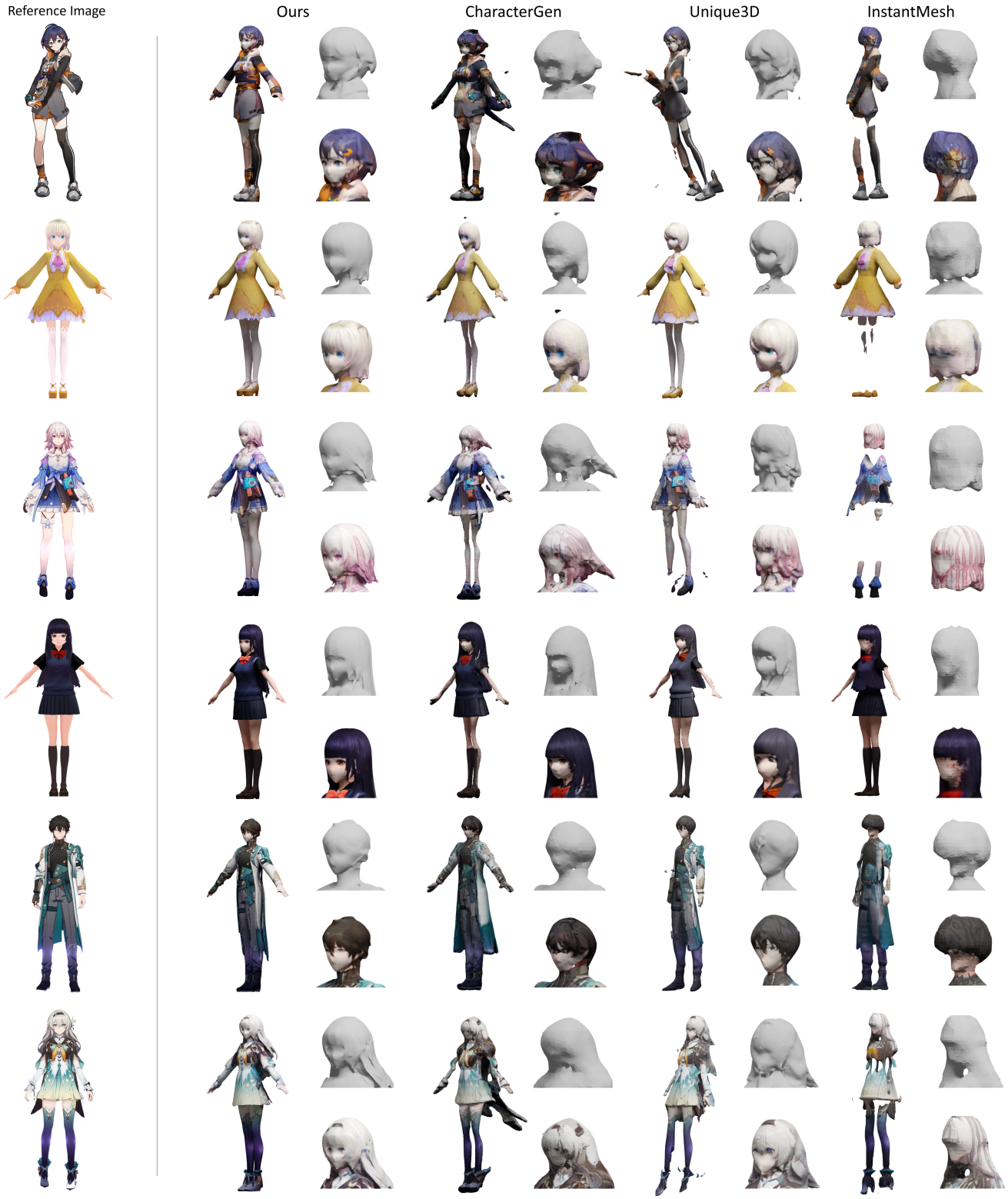

Figure 11. More qualitative comparisons of 3D character generations (#2).

Figure 12. More qualitative comparisons of 3D character generations (multi-view renderings).

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 4

[2] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects, 2023. 5

[3] Junting Dong, Qi Fang, Zehuan Huang, Xudong Xu, Jingbo Wang, Sida Peng, and Bo Dai. Tela: Text to layer-wise 3d clothed human generation. *arXiv preprint arXiv:2404.16748*, 2024. 4

[4] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 3

[5] Taeksoo Kim, Byungjun Kim, Shunsuke Saito, and Hanbyul Joo. Gala: Generating animatable layered assets from a single scan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1535–1545, 2024. 4

[6] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 5

[7] Nikos Kolotouros, Thiemo Alldieck, Enric Corona, Eduard Gabriel Bazavan, and Cristian Sminchisescu. Instant 3d human avatar generation using image diffusion models. In *European Conference on Computer Vision*, pages 177–195. Springer, 2024. 5

[8] Peng Li, Yuan Liu, Xiaoxiao Long, Feihu Zhang, Cheng Lin, Mengfei Li, Xingqun Qi, Shanghang Zhang, Wenhan Luo, Ping Tan, et al. Era3d: High-resolution multiview diffusion using efficient row-wise attention. *arXiv preprint arXiv:2405.11616*, 2024. 4

[9] Hao-Yang Peng, Jia-Peng Zhang, Meng-Hao Guo, Yan-Pei Cao, and Shi-Min Hu. Charactergen: Efficient 3d character generation from single images with multi-view pose canonicalization. *ACM Transactions on Graphics (TOG)*, 43(4): 1–13, 2024. 1

[10] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion, 2022. 4

[11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 4

[12] Tianchang Shen, Jacob Munkberg, Jon Hasselgren, Kangxue Yin, Zian Wang, Wenzheng Chen, Zan Gojcic, Sanja Fidler, Nicholas Sharp, and Jun Gao. Flexible isosurface extraction for gradient-based mesh optimization. *ACM Trans. Graph.*, 42(4):37–1, 2023. 1

[13] Kailu Wu, Fangfu Liu, Zhihan Cai, Runjie Yan, Hanyang Wang, Yating Hu, Yueqi Duan, and Kaisheng Ma. Unique3d: High-quality and efficient 3d mesh generation from a single image. *arXiv preprint arXiv:2405.20343*, 2024. 4

[14] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. 3, 6

[15] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[16] Han Yan, Yang Li, Zhennan Wu, Shenzhou Chen, Weixuan Sun, Taizhang Shang, Weizhe Liu, Tian Chen, Xiaqiang Dai, Chao Ma, et al. Frankenstein: Generating semantic-compositional 3d scenes in one tri-plane. *arXiv preprint arXiv:2403.16210*, 2024. 4

[17] Xueting Yang, Yihao Luo, Yuliang Xiu, Wei Wang, Hao Xu, and Zhaoxin Fan. D-if: Uncertainty-aware human digitization via implicit distribution field. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9122–9132, 2023. 5

[18] Yifan Yang, Dong Liu, Shuhai Zhang, Zeshuai Deng, Zixiong Huang, and Mingkui Tan. Hilo: Detailed and robust 3d clothed human reconstruction with high-and low-frequency information of parametric models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10671–10681, 2024. 5

[19] Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. Function4d: Real-time human volumetric capture from very sparse consumer rgbd sensors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5746–5756, 2021. 5