HumanNorm: Learning Normal Diffusion Model for High-quality and Realistic 3D Human Generation (Supplemental Materials)

Xin Huang[1†,*], Ruizhi Shao[2*], Qi Zhang[1], Hongwen Zhang[2], Ying Feng[1],
Yebin Liu[2], Qing Wang[1]
[1]Northwestern Polytechnical University, [2]Tsinghua University

1. 3D representations

SDF representation. Signed Distance Fields (SDF) is a 3D representation used to describe the geometry surface of an object. It is expressed implicitly through neural networks like MLP. For a sampling point x, everything satisfying $f(x) = 0$ is considered to be part of the object's surface, while the region where $f(x) < 0$ represents the object's interior, and $f(x) > 0$ indicates the object's exterior. SDF can be employed in the synthesis of images from arbitrary viewpoints through methods such as differentiable volume rendering or differentiable marching cubes for geometry extraction and re-rendering.

DMTET representation. DMTET [12] is a hybrid 3D representation that combines explicit and implicit forms. It divides 3D space into dense tetrahedra, which is an explicit partition. Simultaneously, the vertices of these tetrahedra record properties of the 3D object, including SDF, deformation, color, etc. These properties are expressed through the implicit functions of neural networks. By combining explicit and implicit representations, DMTET can be optimized more efficiently and easily transformed into explicit structures like mesh representations. During the generation process, DMTET can be converted into a mesh in a differentiable manner, enabling rapid high-resolution multi-view rendering. We utilize DMTET as the 3D representation in both the geometry generation and texture generation phases.

2. Implementation Details

Dataset. Our dataset comprises 2952 3D human body models. These include 526 models from the THuman2.0 dataset [14], 1779 models from the Twindom dataset [13], and 647 models from the CustomHumans dataset [3]. We use these models to generate depth maps, normal maps, and color maps. To augment the dataset, we divide the human body into four distinct sections: the head, the upper body, the lower body, and the full body. For each model, we render a set of 120 images, each set comprising depth maps, normal maps, and color maps. The normal maps are transformed into camera coordinates by the rotation of the camera parameter. We utilize CLIP [9] to generate prompts for the images, supplementing them with additional text to label various data types such as "depth map" and "normal map". We also include view-dependent descriptors for the view direction, such as "front view", "back view", "left side view", and "right side view", as well as body-aware text for specific regions of the human body, including "head only", "upper body", "lower body", and "full body".

Training of normal-adapted and depth-adapted diffusion models. The base stable diffusion model used in our method is Stable Diffusion V1.5 [11]. We fine-tune the stable diffusion model using our depth pairs and normal pairs for 15K iterations. The learning rate is set to 1×10^{-5} and the batch size is set to 4. Exponential Moving Average (EMA) is used during the training. After fine-tuning, we obtain a normal-adapted diffusion model and a depth-adapted diffusion model. The fine-tuning code is from Diffusers (https://huggingface.co/docs/diffusers/index), a library for state-of-the-art pretrained diffusion models for generating images, audio, and even 3D structures of molecules

Training of normal-aligned diffusion model. To guide the generation of stable diffusion using a normal map, we follow the fine-tuning strategy of ControlNet [16]. We fine-tune Stable Diffusion V1.5 for 30K iterations using normal-image pairs. The normal maps are used as extra conditions. The learning rate is set to 1×10^{-5} and the batch size is set to 4. The fine-tuning code is also from Diffusers.

Details of progressive positional encoding. In progressive geometry generation, we employ progressive positional encoding. Specifically, the position encoding for SDF features in DMTET has a total of 32 dimensions, where the lower dimensions represent lower-frequency features and higher dimensions represent higher-frequency features. Initially,

[†] Work done during an internship at Tsinghua University.

[*] Equal contribution.

we utilize a 32-dimensional mask with the first 16 dimensions set to 1 and the latter 16 dimensions set to 0. We multiply this mask with the SDF's position encoding to remove the high-frequency components. During training, every 500 iterations, we convert 2 of the 0 positions in the mask to 1, gradually enabling the network to learn high-frequency components. After 4,000 iterations, all positions in the mask become 1, resulting in the position encoding encompassing both low-frequency and high-frequency components.

Details of progressive SDF loss. During the training process, at the 3,000 iterations, we extract the current geometry to form a coarse mesh. This coarse mesh exhibits a reasonable shape and features a relatively smooth surface. We utilize it to compute the SDF loss for subsequent stages. Specifically, within the bounding box of the 3D generation, we randomly sample 100,000 points at each iteration. Then we calculate the SDF loss by comparing the SDF values of these points in the coarse mesh with the SDF values predicted by the network. The weight of the SDF loss among all the losses is set to 1500 and is only computed after the 3,000 iterations.

Details of geometric resolution. We similarly adopt an approach to gradually increase the geometric resolution. Initially, the resolution of the DMTET in 3D space is set to 128^3. As training proceeds, we incrementally double this resolution every 3,000 iterations. So at 3,000 iterations, the resolution is set to 256^3, and it will eventually reach 512^3 at 6,000 iterations. In the early training stages, this results in fewer generated geometry facets, with each facet occupying more pixels in the rendered images. Consequently, the gradients produced by the loss are more evenly distributed across the points of each facet, leading to more stable geometry generation. As the geometric resolution increases, the number of geometry facets also increases, allowing for the representation of more intricate details, including features like hair and clothing folds.

Details of texture generation. In texture generation, the initial 2,000 iterations are utilized as coarse-level optimization and employ SDS loss, while the subsequent 8,000 iterations serve as fine-level optimization, using the multi-step SDS loss and perceptual loss. For the multi-step SDS loss, the diffusion model performs varying numbers of iterations based on the timestep t with added noise. Specifically, The total timestep of our diffusion model is 1000, when the timestep is t, the diffusion model iterates $(t/25+1)$ times. We employ the DPM++ solver [6] as our diffusion scheduler. To enhance training stability, we also incorporate a DU (Dataset Update) strategy similar to what was proposed in Instruct-NeRF2NeRF [2]. During computation for the multi-step loss at each iteration, we save the image results of multi-step diffusion denoising in a cached dataset, which are reused in subsequent training processes. Every 10 iterations, we will use multi-step diffusion denoising to update the images in the cached dataset.

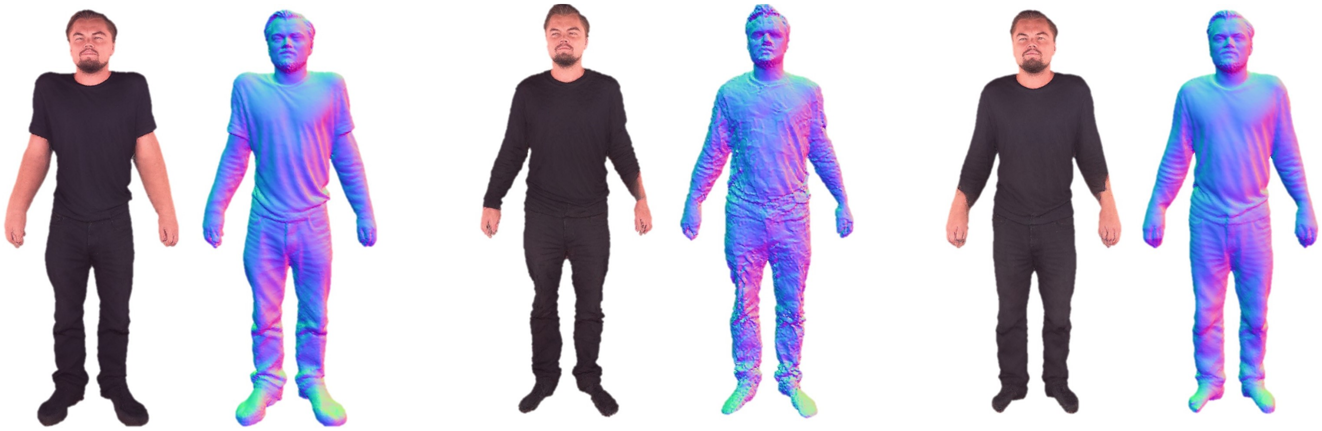
Noise and guidance scale of the diffusion model. In the geometry stage, both our normal-adapted and depth-adapted diffusion models have a guidance scale of 50. Similar to the strategy employed in progressive geometry generation, we introduce noise progressively during the geometry stage. In the first 5,000 steps, the timestep t of noise follows the distribution $\mathcal{U}(0.02, 0.8)$. Between 5,000 and 8,000 steps, the timestep t of noise follows the distribution $\mathcal{U}(0.02, k)$ with parameter $k = 0.2 + (0.8 - 0.2)\frac{8000-step}{8000-5000}$. After 8,000 steps, the timestep t of noise follows the distribution $\mathcal{U}(0.02, 0.2)$. In the texture stage, our geometry-guided diffusion model has a guidance scale of 7.5, and the controlled condition scale is set to 1.0. During the coarse level of texture generation, the timestep t of noise follows the distribution $\mathcal{U}(0.02, 0.98)$. In the fine level, the timestep t of noise follows the distribution $\mathcal{U}(0.02, 0.5)$.

Learning rate and the weight of losses in 3D generation. We adopt the AdamW optimizer in 3D generation. The learning rate of θ_g is set to 2×10^{-5} and the learning rate of θ_c is set to 1×10^{-3}. In the geometry generation, the weight of the normal SDS loss is set to 1.0, and the weight of the depth SDS loss is 1.0. In the texture generation, the weight of perceptual loss is set to 1.0.

Part-based optimization. We primarily divide the human body into four parts for generation: head, upper body, lower body, and the full body. To ensure that the rendered images cover each of these four parts separately, we predefine the camera positions and focal lengths accordingly. During the generation process, the probability of sampling from these four camera positions varies based on the optimization objective. When generating only the head, we sample from the camera capturing the head alone. When generating the upper body of the human, we assign a sampling probability of 0.7 to the upper body and 0.3 to the head. When generating the entire human body, we adjust the sampling strategy progressively. In the first 10,000 iterations, we assign a sampling probability of 0.7 to the entire body and 0.1 to each of the head, upper body, and lower body. In the subsequent 5,000 iterations, we assign a sampling probability of 0.1 to the entire body and 0.3 to each of the head, upper body, and lower body.

3. User Study

Following TADA [5] and DreamHuman [4], we conducted a user study to further assess the quality of the 3D human models generated by our method. Our approach was compared with five state-of-the-art methods across 30 prompts. For each prompt, 50 volunteers (comprising 40 students specializing in computer vision and graphics, and 10 members of the general public) evaluated the color and normal map videos rendered from the generated 3D humans. They



(a) W/O progressive SDF loss (b) W/O progressive positional encoding (c) Full method

Figure 1. **Importance of progressive SDF loss and progressive positional encoding.**

Table 1. **Results of user study.** The table reports the user preference percentages in detail.

	Q1 (%)		Q2 (%)		Q3 (%)	
	Best	Second best	Best	Second best	Most	Second most
DreamFusion	5.36	22.27	4.73	20.55	9.27	22.55
LatentNeRF	3.09	11.82	6.64	8.45	8.45	12.91
TEXTure	3.64	10.27	3.91	6.64	4.91	9.09
Fantasia3D	9.91	**41.45**	10.45	**50.55**	12.64	**39.00**
Ours	**78.00**	14.18	**74.27**	13.82	**64.73**	16.45

	Q1 (%)	Q2 (%)	Q3 (%)
DreamHuman	8.79	18.20	25.80
TADA	16.91	11.25	15.20
Ours	**74.30**	**70.55**	**59.00**

voted on three questions:

- Q1: Which 3D human model exhibits the best (and second best) texture quality?
- Q2: Which 3D human model displays the best (and second best) geometric quality?
- Q3: Which 3D human model aligns most closely (and second most closely) with the given prompt?

Since the source code of DreamHuman [4] is not publicly accessible, we sourced the results from its project page. The results of LatentNeRF [7], TEXTure [10], Fantasia3D [1], and TADA [5] are produced using their official code with default settings. Meanwhile, the results of DreamFusion [8] are generated using an unofficial implementation in ThreeStudio, a unified framework for 3D content creation (https://github.com/threestudio-project/threestudio). We all collect 1,500 pairwise comparisons. The results are shown in Tab. 1. One can see that our method surpasses the performance of the text-to-3D content methods and text-to-3D human methods, particularly in terms of geometry and texture quality. These results underscore the superior performance of our approach.

4. More Comparisons

We offer further qualitative comparisons with the four text-to-3D content methods and the two text-to-3D human methods. As depicted in Fig. 6 and Fig. 7, Fantasia3D may generate textures that are not aligned with the geometry (as seen in the second row of Fig. 6). However, the textures produced by our method are accurately aligned with the generated geometry. When compared to the four text-to-3D content methods, our method can generate head-only and upper-body 3D humans with more detailed geometry and a more realistic appearance. In Fig. 8, we present full-body results in comparison with DreamHuman and TADA. It is evident that the results produced by baselines contain over-saturated textures and smooth geometry, whereas our method yields a more natural appearance and geometric details. Additionally, we add a comparison with AvatarVerse [15], as shown in Fig. 5. The 3D humans by AvatarVerse are over-saturated. In contrast, HumanNorm produces results with appearances that are more lifelike.

5. More Ablation Studies

Effectiveness of multi-step SDS loss and perceptual loss. As shown in Fig. 3, the multi-step SDS loss guarantees a lifelike appearance without over-saturation. The perceptual loss enhances the texture details but struggles to produce reasonable color without the multi-step SDS loss. The multi-step SDS loss is critical for solving the over-saturation issue.

Effectiveness of progressive SDF loss. In Fig. 1 (a), we display the results obtained in the absence of progressive SDF loss. The 3D human exhibits a distorted body shape. However, the introduction of progressive SDF loss effectively constrains the wrong growth of the human body, thereby avoiding unreasonable body shapes.

Effectiveness of progressive positional encoding. In Fig. 1 (b), we conduct an experiment where the frequency of hash encoding is fixed. The results reveal extensive noise on the surface of the geometry, which can be attributed to the high-frequency content learned during the initial training phase. A contrasting case is presented in Fig. 1 (c) when a progressive positional encoding is employed. Our strategy reduces the learning of high-frequency information during the initial training phase, resulting in a stable geometry devoid of geometric noise.

6. Applications

Our method offers the capability to edit both the texture and geometry of the generated 3D humans by adjusting the input prompt. As demonstrated in Fig. 4, we modify the color and style of Messi's clothing, as well as his hairstyle, all while maintaining his identity. While geometry editing poses a greater challenge than texture editing, our method exhibits precise control over geometry generation, even allowing us to generate Messi wearing a hat. Furthermore, the edited geometry is rich in detail, as evidenced by the intricate details in the sweater. **More applications can be viewed on our attached project page.**

7. Failure cases

Our method may fail to generate 3D humans with challenging poses or loose clothes, as illustrated in Fig. 2. For example, we encountered difficulty in accurately rendering the loose dress of the dancer. Additionally, we observed that the hat of the cowboy appeared broken in some instances.

8. Ethics statement

The objective of HumanNorm is to equip users with a powerful tool for creating realistic 3D Human models. Our method allows users to generate 3D Humans based on their specific prompts. However, there is a potential risk that these generated models could be misused to deceive viewers. This problem is not unique to our approach but is prevalent in other generative model methodologies. Moreover, it is of paramount importance to give precedence to diversity in terms of gender, race, and culture. As such, it is essential for current and future research in the field of generative modeling to consistently address and reassess these considerations.

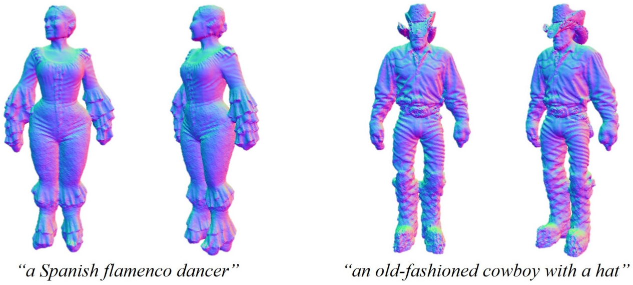

"a Spanish flamenco dancer"
"an old-fashioned cowboy with a hat"

Figure 2. **Failure cases.** Our method may fail to generate 3D humans with challenging poses or loose clothes.

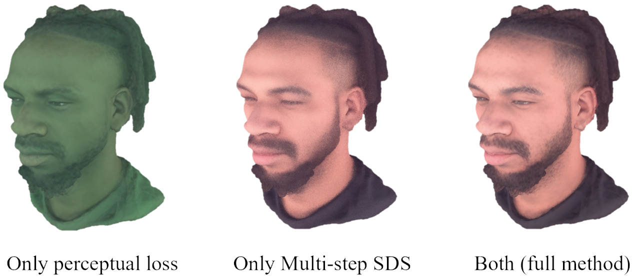
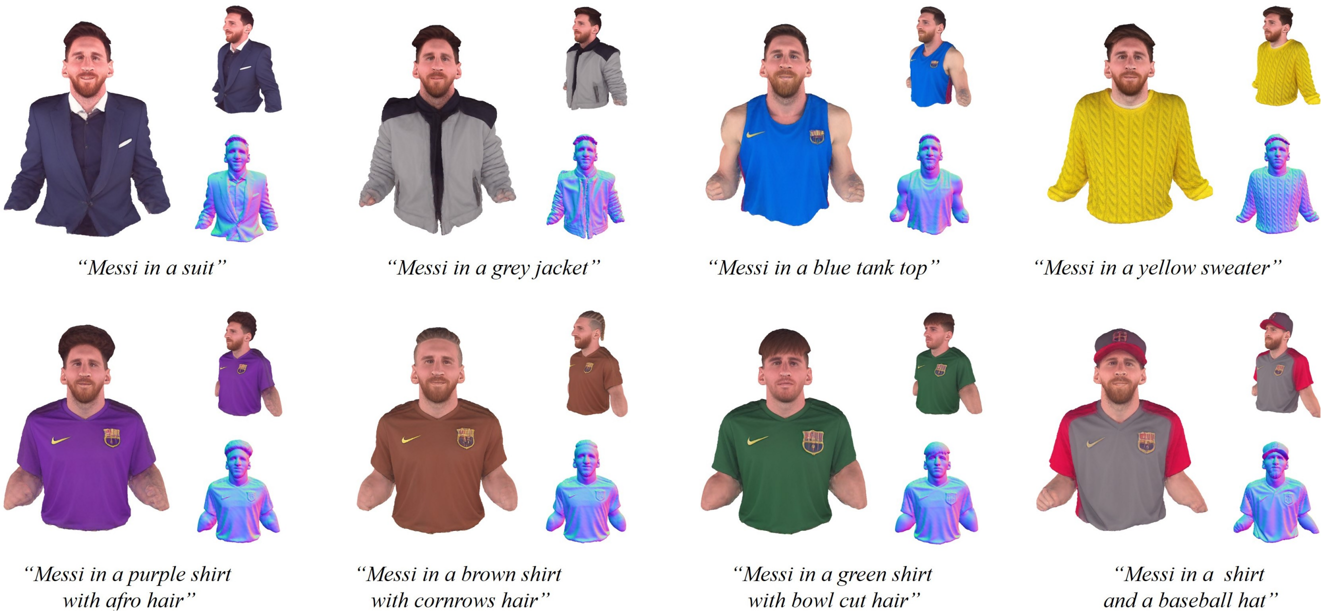


Figure 3. **Effectiveness of multi-step SDS loss and perceptual loss.**

References

[1] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. *arXiv preprint arXiv:2303.13873*, 2023. 3

[2] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. *arXiv preprint arXiv:2303.12789*, 2023. 2


"Messi in a suit"

"Messi in a grey jacket"

"Messi in a blue tank top"

"Messi in a yellow sweater"

"Messi in a purple shirt with afro hair"

"Messi in a brown shirt with cornrows hair"

"Messi in a green shirt with bowl cut hair"

"Messi in a shirt and a baseball hat"

Figure 4. **Text-based editing.** Our method provides the ability to modify both the texture and geometry of the generated 3D humans by simply altering the input prompt.

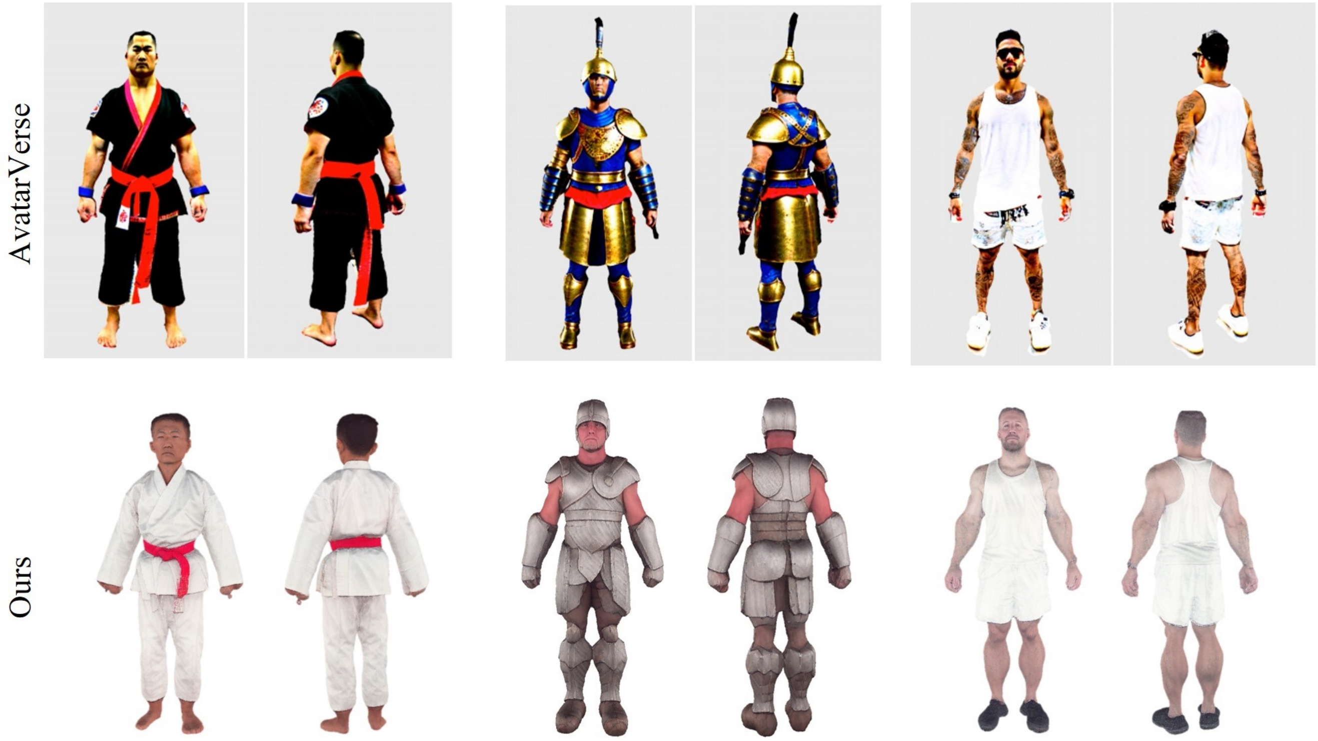


Figure 5. **Comparisons with AvatarVerse.** The results of AvatarVerse are copied from its paper.

[3] Hsuan-I Ho, Lixin Xue, Jie Song, and Otmar Hilliges. Learning locally editable virtual humans. In *CVPR*, pages 21024–21035, 2023. 1

[4] Nikos Kolotouros, Thiemo Alldieck, Andrei Zanfir, Eduard Gabriel Bazavan, Mihai Fieraru, and Cristian Sminchisescu. Dreamhuman: Animatable 3d avatars from text. *arXiv preprint arXiv:2306.09329*, 2023. 2, 3

[5] Tingting Liao, Hongwei Yi, Yuliang Xiu, Jiaxaing Tang,

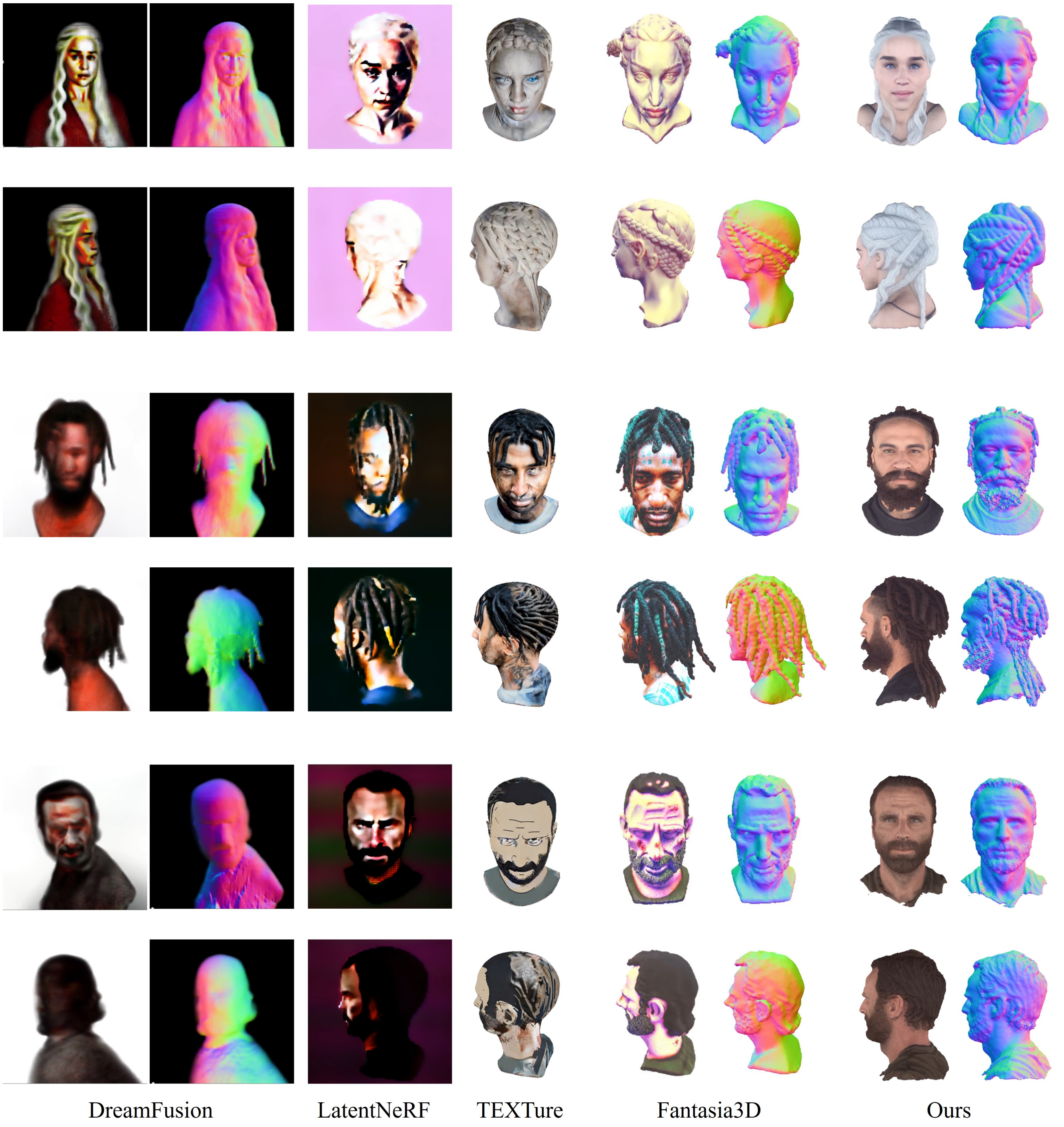


Figure 6. **Comparison with text-to-3D content methods on the head-only 3D human generation.**

Yangyi Huang, Justus Thies, and Michael J Black. Tada! text to animatable digital avatars. In *3DV*, 2024. 2, 3

[6] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022. 2

[7] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *CVPR*, pages 12663–12673, 2023. 3

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

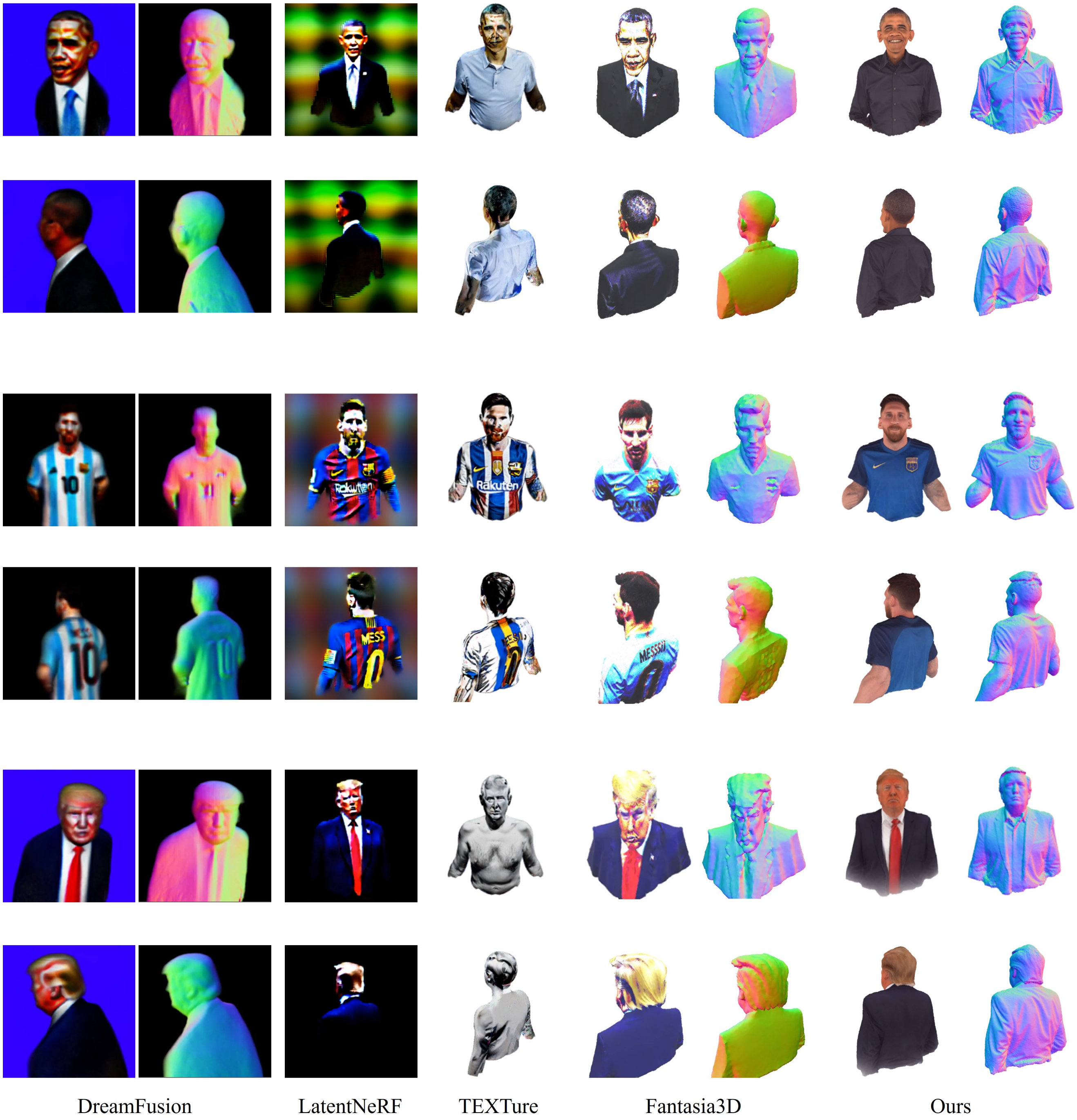


Figure 7. **Comparison with text-to-3D content methods on the upper-body 3D human generation.**

2023. 3

[9] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 1

[10] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. In *ACM SIGGRAPH Conference Proceedings*, 2023. 3

[11] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1

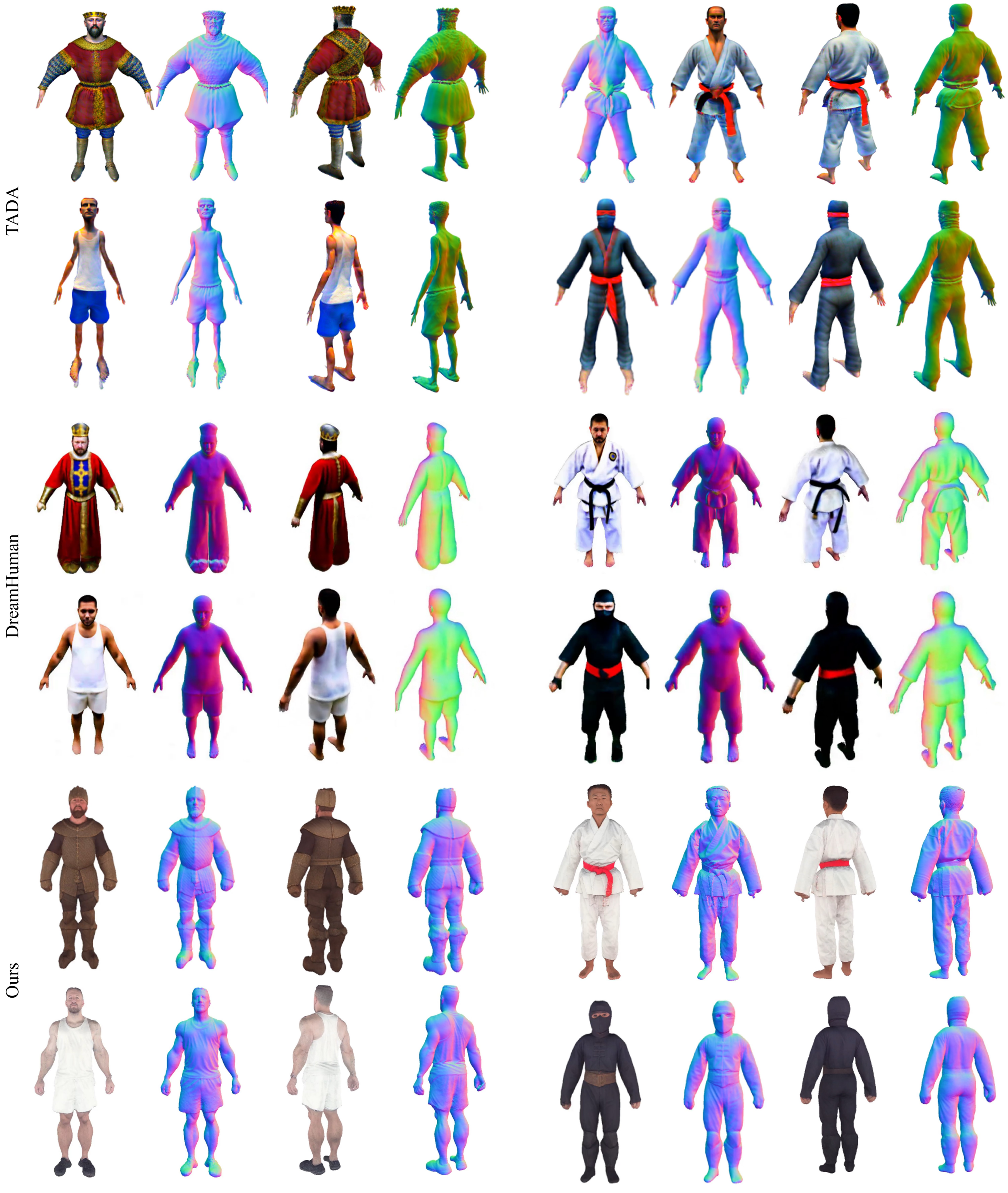


Figure 8. **Comparison with text-to-3D human methods on the full-body 3D human generation.**

[12] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *NeurIPS*, 34: 6087–6101, 2021. 1

[13] TwinDom. Twindom. https://web.twindom.com/. Accessed: 2023-09-28. 1

[14] 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 *CVPR*, pages 5746–5756, 2021. 1

[15] Huichao Zhang, Bowen Chen, Hao Yang, Liao Qu, Xu Wang, Li Chen, Chao Long, Feida Zhu, Kang Du, and Min Zheng. Avatarverse: High-quality & stable 3d avatar creation from text and pose. *arXiv preprint arXiv:2308.03610*, 2023. 4

[16] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 1