Chupa: Carving 3D Clothed Humans from Skinned Shape Priors using 2D Diffusion Probabilistic Models (Supplementary Material)

Byungjun Kim[1*] Patrick Kwon[2*] Kwangho Lee[2] Myunggi Lee[2]
Sookwan Han[1] Daesik Kim[2] Hanbyul Joo[1]
[1]Seoul National University [2]Naver Webtoon AI
https://snuvclab.github.io/chupa/

A. Detailed formulation of Diffusion Models

We provide a detailed introduction to Gaussian-based diffusion models [5, 14]. Given the target data distribution $x_0 \sim q(x_0)$, the goal of diffusion models is to learn a model distribution p_θ that approximates q, while being easy to sample from. To achieve both objectives, diffusion models define a *forward process* that gradually introduces noise to the original data x_0 to generate a sequence of noised data $x_1, x_2, ..., x_T$. Additionally, a *reverse process* is defined, which aims to denoise the noised data x_t and produce less noisy data x_{t-1}. Once trained, Gaussian-based diffusion models sample data x_0 by first sampling x_T from a Gaussian distribution $\mathcal{N}(0, \mathbf{I})$ and iteratively sampling x_{t-1} from the previous step x_t. To ensure $x_T \sim \mathcal{N}(0, \mathbf{I})$, it is required for T to be sufficiently large.

The forward process is formulated as a Markov chain according to a variance schedule $\beta_1 < \beta_2 < ... < \beta_T$:

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}) \tag{1}$$

$$q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{2}$$

Note that to sample $x_t \sim q(x_t|x_0)$, it is not required to apply forward diffusion t times. Instead, using the notation $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$, we have a closed form expression:

$$q(x_t|x_0) := \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t\mathbf{I}) \tag{3}$$

Consequently, we can view x_t as a linear combination of x_0 and $\epsilon \sim \mathcal{N}.(0, \mathbf{I})(x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{(1-\bar{\alpha}_t)}\epsilon)$

Given the fixed forward process, p is designed to approximate the unknown true posterior $q(x_{t-1}|x_t)$. This is achieved through the use of a deep neural network with learnable parameters θ.

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{4}$$

$$p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{5}$$

Ho et al. [5] proposed a specific parameterization for $\mu_\theta(x_t, t)$ such that the neural network outputs the estimated noise ϵ_θ instead of predicting μ_θ.

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)) \tag{6}$$

For training, the variational lower bound is optimized and simplifies the following Eq. (7) that enables the model to learn how to predict the added noise.

$$L_{\text{simple}} = \mathbb{E}_{x_0,t,\epsilon\sim\mathcal{N}(0,I)}[||\epsilon - \epsilon_\theta(x_t, t)||^2] \tag{7}$$

In practice, Ho et al. [5] uses a U-Net backbone [12] to output the predicted noise ϵ_θ which has the same dimensionality as the input noisy sample x_t. To solve an image-to-image translation task, Saharia et al. [13] concatenates a spatial conditioning input y to x_t channel-wise and modifies the learning objective as Eq. (8).

$$L_{\text{simple}} = \mathbb{E}_{x_0,y,t,\epsilon\sim\mathcal{N}(0,I)}[||\epsilon - \epsilon_\theta(x_t, y, t)||^2] \tag{8}$$

A.1. Diffusion Training

B. Normal map-based mesh optimization

Camera parameters. In our normal map-based mesh optimization method, we require camera parameters to rasterize

*Equal contribution

(a) iter. 0 (b) iter. 500 (c) iter. 1000 (d) iter. 1500 (e) iter. 2000

Figure 1. **Coarse-to-fine optimization**. Starting from a decimated SMPL-X mesh, we perform optimization in a coarse-to-fine manner. By increasing the resolution of the mesh for every 500 iterations, we progressively deform the mesh to match the input normal maps, without losing high-frequency details.

the mesh into normal maps that are aligned with those generated from our dual-generation diffusion model. To generate the frontal normal map of the initial SMPL-X mesh (explained in Sec. 3.1), we utilize a weak perspective camera which shares the same parameters as our training data setup. For the second mesh refinement stage (explained in Sec. 3.3), we also employ weak perspective cameras that are defined in the same manner for both body and face rendering.

Coarse-to-fine optimization. We adopt the coarse-to-fine optimization strategy presented by NDS [17] for mesh optimization. Specifically, we begin with a coarse mesh and progressively increase the resolution through a remeshing technique, presented by Botsch and Kobbelt [1]. As demonstrated in [17], initializing optimization with a large number of vertices can lead to meshes with undesired geometry, such as degenerate triangles and self-intersections. Therefore, we start the optimization from a decimated version of our initial SMPL-X, which contains 3,000 vertices [4]. During optimization, for every 500 iterations, we apply remeshing [1] to increase the model resolution. It is worth noting that each iteration corresponds to a single gradient descent step, with respect to the loss based on a randomly sampled normal map. Following NDS [17], we perform optimization for a total of 2,000 iterations and decreased the gradient descent step size for the vertices by 25% after each remeshing. As Fig. 1 shows, we can handle the large deviation from the initial mesh without losing high-frequency details, due to the coarse-to-fine optimization scheme.

Loss weight scheduling. While we follow the individual loss objective terms and scheduling of NDS [17] for our mesh optimization loss in Sec. 3.2, we added our side loss term L_{sides} to the objective with weight term $\lambda_{\text{sides}} = 0.1$, which we decrease by 10% after each remeshing. We also set the loss weights for L_{normal} equivalent to L_{shading} in the original paper for NDS. During optimization, we progressively increase the geometric regularization term $L_{\text{laplacian}}, L^{reg}_{\text{normal}}$ to encourage the generation of smooth surfaces for the final mesh. For the second mesh refinement stage, which optimizes the earlier mesh based on the refined normal maps from multiple views (total of 36 views), we set $\lambda_{\text{sides}} = 0$ since the side views can now be well constrained without the sidewise loss.

Refine by resampling. To refine the mesh from dual normal map-based optimization, we render both full body and face normal maps and refine them with resampling technique (Sec. 3.3). Here, we render 36-view normal maps with 10° yaw interval, and set (t_0, K) to $(0.02, 2)$, respectively, both for body and face normal map refinement.

C. Qualitative Results

More generation results. Fig. 2 shows more random generation results from Chupa. We generate the human meshes based on SMPL-X parameters from the AGORA dataset [9], which includes SMPL, SMPL-X parameters fitted to $4,240$ 3D human scans. We can generate human scans with various identities and can be generalized to diverse poses.

Changing shape parameter $\boldsymbol{\beta}$. To control the shape of the generated mesh, we can control the shape parameter $\boldsymbol{\beta}$ of input SMPL-X mesh [8, 10]. Fig. 3, Fig. 4 shows the generated meshes according to the variation of β with fixed pose parameter Θ, where β_1, β_2 corresponds to the first and second component of the shape parameter respectively [8].

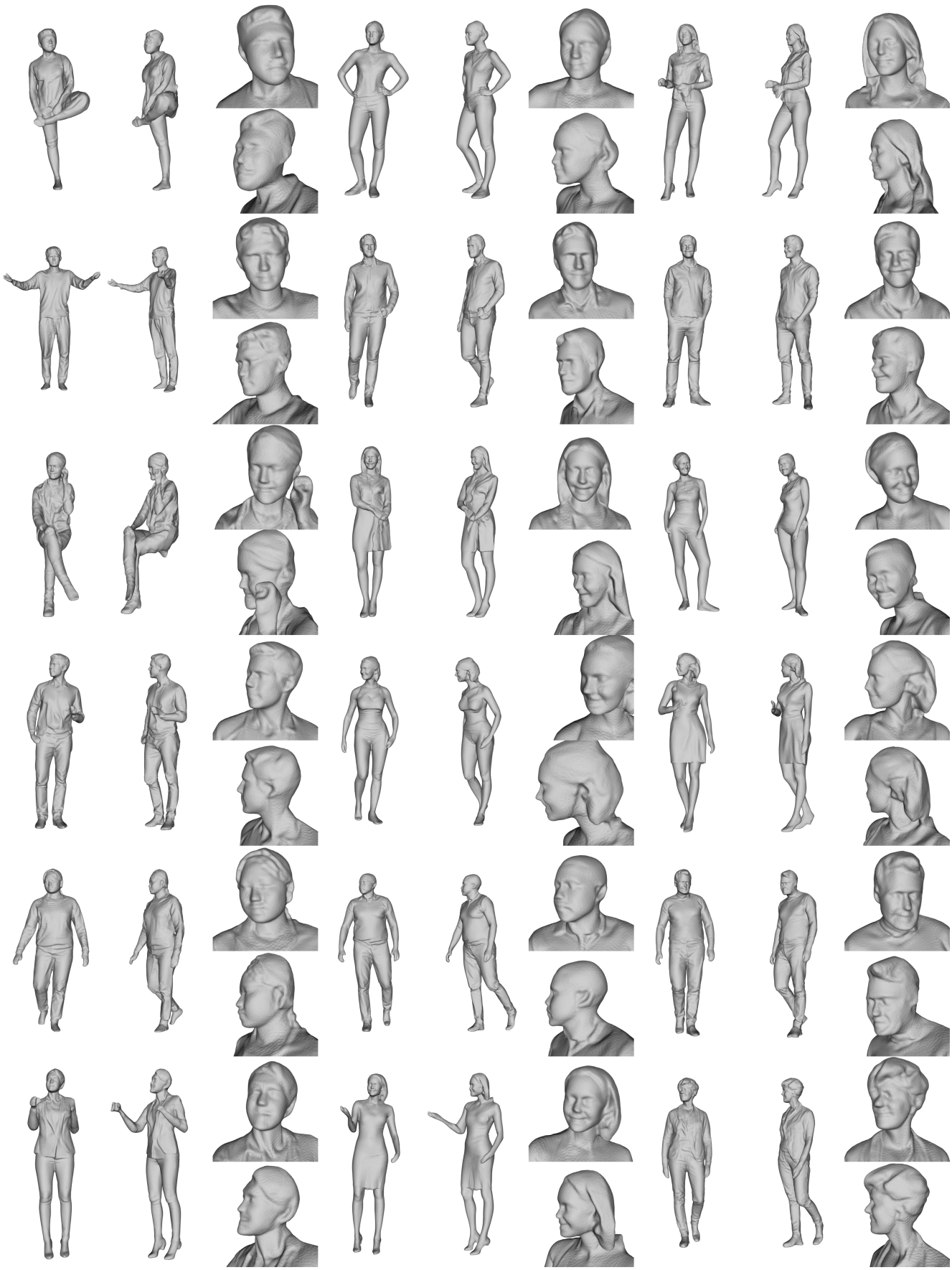
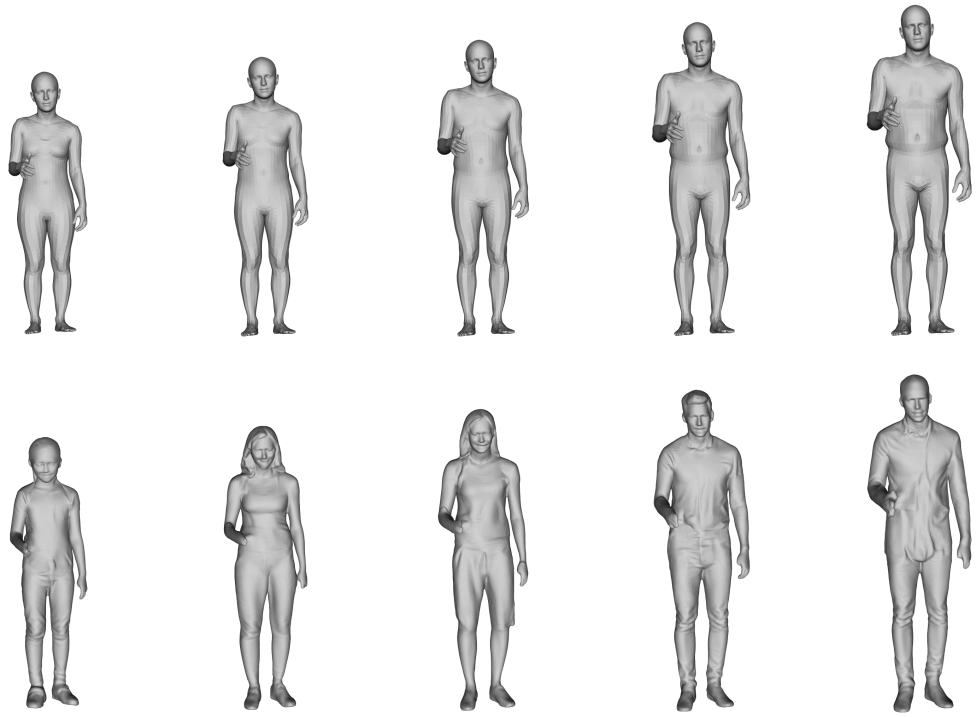
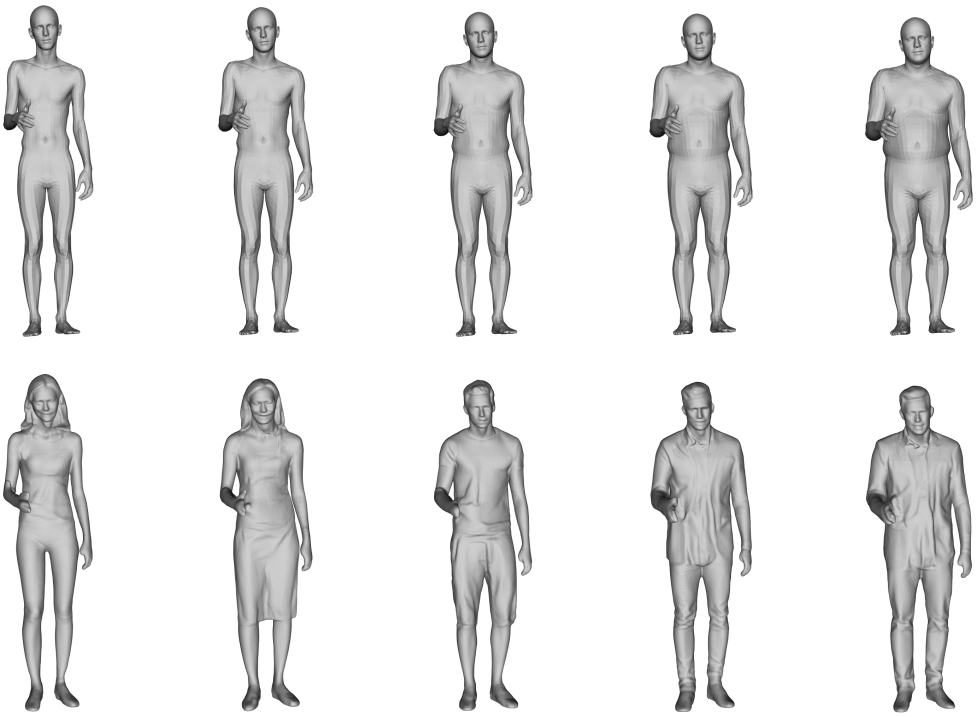


Figure 2. **More random generation results.**



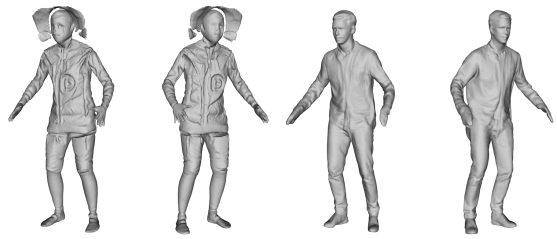
(a) $\beta_1 = -2$ (b) $\beta_1 = -1$ (c) $\beta_1 = 0$ (d) $\beta_1 = 1$ (e) $\beta_1 = 2$

Figure 3. **Changing shape parameter $\boldsymbol{\beta}_1$.**

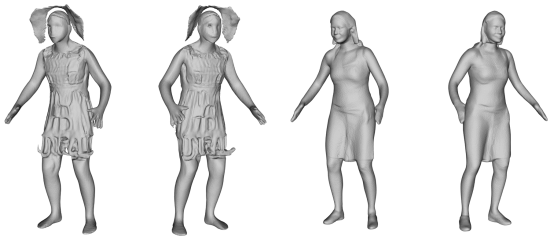


(a) $\beta_2 = -2$ (b) $\beta_2 = -1$ (c) $\beta_2 = 0$ (d) $\beta_2 = 1$ (e) $\beta_2 = 2$

Figure 4. **Changing shape parameter $\boldsymbol{\beta}_2$.**



(a) “a boy wearing a jacket”



(b) “a girl wearing a dress”

Figure 5. **Comparison with AvatarCLIP.** The left two columns are from AvatarCLIP, the right two columns are from Chupa (ours).

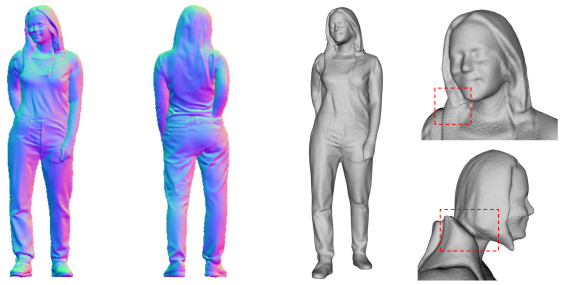


Figure 6. **Depth ambiguity problem.** Chupa may generate broken geometry, due to the depth ambiguity problem of our mesh reconstruction method(left: dual normal map, right: final mesh).

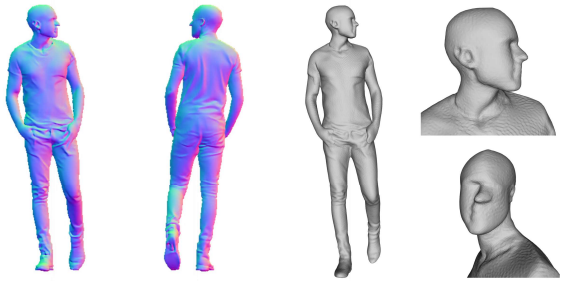


Figure 7. **Face direction matters.** Chupa may generate unnatural face geometry, when the face direction is not aligned with the input view (left: dual normal map, right: final mesh).

Figure 8. **Out-of-distribution pose.** Chupa may generate implausible geometry for some out-of-distribution pose (left: SMPL-X, middle: dual normal map, right: final mesh).

Comparison with AvatarCLIP. We compare our text-guided generation results with AvatarCLIP [6], a text-guided 3D avatar generation pipeline that also initializes its 3D implicit surface model [16] with a SMPL model. Once initialized, AvatarCLIP optimizes the 3D model based on a CLIP loss [11] on the rendered results, to match the 3D model according to the text description. Fig. 5 shows that Chupa can generate more realistic 3D human mesh while minimizing unnatural artifacts. Note that while AvatarCLIP takes more than 3 hours to generate a mesh, Chupa takes 3 minutes with a single RTX3090.

D. Failure Cases

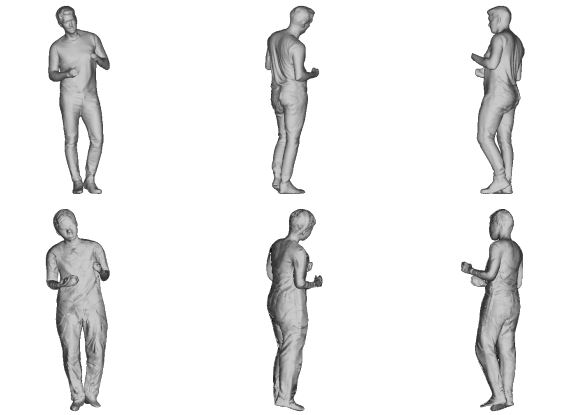
Depth ambiguity problem. Our dual normal map-based mesh reconstruction method (Sec. 3.2) has inherent depth ambiguity issues, as it only uses front and back-view normal maps for the initial optimization. When the given normal maps largely deviates from the initial SMPL model, *e.g.*, long hair, the vertices for both head and shoulder deforms to match the provided hairstyle, creating artifacts during deformation. Fig. 6 shows that while the hairstyle seems to be well-reconstructed in the front view, there exists unnatural seams and broken geometry at close view.

Face direction matters. When the input pose contains misaligned body and face direction, the final output might display unnatural face geometry. For example, when the face is turned to the side direction (Fig. 7), the diffusion models might fail to generate realistic faces for reconstruction. To make matters worse, the small distortion due to depth ambiguity during reconstruction (Sec. 3.2) can have huge impact on the perceptual quality of faces. Fig. 7 shows an example of such cases, where the resulting face mesh displays unnatural geometry.

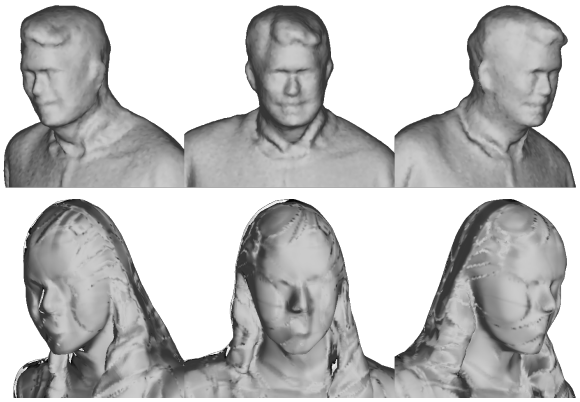
Out-of-distribution pose. While our method can be generalized for diverse poses, there exists out-of-distribution poses that the diffusion generative model fails to create plausible normal maps from. Fig. 8 shows such examples of unrealistic normal maps, which leads to 3D meshes with bad geometry.

E. User Study

We conduct a perceptual study asking user preference between the meshes from our method and gDNA [2]. We collect 100 participants through CloudResearch Connect [3] and get 78 valid answers out of them. Each participants are given 40 problems which consist of 20 problems for body and 20 problems for face. Fig. 9 shows the example problems.



(a) Body



(b) Face

Figure 9. **User study problem example**. The 3 views of mesh from our method and gDNA [2] with the same SMPL parameter are rendered as shading images. Each user is asked to choose more realistic shapes between two rows, where each row corresponds to the images from each method. Two rows are randomly shuffled.

F. Ablation Study

We present additional ablation study results on changing various hyperparameters such as resampling parameters, sampling angle, and the sampling scheme for dual generation. In Tab. 1 and Tab. 2, we present the effect of choosing different refinement parameters (t_0, K) and the sampling angle during the refinement stage for both shaded and normal maps of the resulting meshes. We also present the effect of using different diffusion samplers in Tab. 3.

Table 1. **Ablation study on resampling.** We see the effects of (t_0, K) both for body and face, with the number of views fixed as 36.

(t_0, K) Body	(t_0, K) Face	$\text{FID}_{\text{normal}} \downarrow$	$\text{FID}_{\text{shade}} \downarrow$
$(0.02, 2)$	-	22.61	37.13
$(0.02, 4)$	-	26.68	46.19
$(0.02, 6)$	-	31.39	51.98
$(0.04, 2)$	-	27.02	46.34
$(0.06, 2)$	-	31.71	52.65
$(0.02, 2)$	$(0.02, 2)$	**21.90**	**36.58**
$(0.02, 2)$	$(0.02, 4)$	22.42	37.57
$(0.02, 2)$	$(0.02, 6)$	22.65	38.11
$(0.02, 2)$	$(0.04, 2)$	22.41	37.64
$(0.02, 2)$	$(0.06, 2)$	22.65	37.94

Table 2. **Ablation on the number of views for refinement**. We see the effects of the number of views for refinement with $t_0 = 0.02, K = 2$ as fixed.

N_{views}	θ_{step}	$\text{FID}_{\text{normal}} \downarrow$	$\text{FID}_{\text{shade}} \downarrow$
4	90°	30.88	41.85
6	60°	29.01	41.30
12	30°	25.21	39.53
36	10°	**21.90**	**36.58**

Table 3. **Ablation on sampling scheme**. We ablate on the sampling scheme of our diffusion model for dual normal map generation. Here, we compute FID scores based on the results of dual normal map-based optimization without refinement.

Method	$\text{FID}_{\text{normal}} \downarrow$	$\text{FID}_{\text{shade}} \downarrow$
Euler [7]	28.84	37.36
DDIM [15]	26.76	34.79
DDPM [5]	26.31	37.13

Refine by resampling. Tab. 1 shows the effects of varying (t_0, K) for resampling. The first 6 rows show the results of varying (t_0, K) for body normal map refinement without face refinement. And the next 6 rows show the results of varying (t_0, K) for face normal map refinement with fixed (t_0, K) for body normal map refinement. For both body and face, the smaller forward time steps and fewer iterations show better performance since large forward steps or many iterations may lead to the normal map inconsistent with the original normal maps.

The number of views for mesh refinement. Tab. 2 shows the performance with the varying number of views used for the mesh refinement stage (Sec. 3.3), where $N_{\text{views}}, \theta_{\text{step}}$ correspond to the number of views and the yaw interval between views respectively. Here, the hyperparameters (t_0, K) for resampling are fixed as $(0.02, 2)$. It shows that increasing the number of views leads to better performance.

Sampling scheme of the diffusion model. As mentioned in Sec. 4.1, we generate dual normal maps with the same denoising steps used during training, which is the sampling scheme of DDPM [5]. Here, we ablate on the different sampling schemes for diffusion probabilistic models, with two additional samplers [7, 15] set to $t = 50$. Tab. 3 shows that the sampling scheme doesn't affect the performance significantly. Note that we compute the score without the mesh refinement stage (Sec. 3.3) to analyze the effects of the sampler since the refinement stage only involves a small number of denoising steps.

References

[1] M. Botsch and L. Kobbelt. A remeshing approach to multiresolution modeling. In *Proc. Eurographics*, 2004. 2

[2] X. Chen, T. Jiang, J. Song, J. Yang, M. J. Black, A. Geiger, and O. Hilliges. gdna: Towards generative detailed neural avatars. In *Proc. CVPR*, 2022. 6

[3] CloudResearch Connect. https://www.cloudresearch.com/products/connect-for-researchers/. 6

[4] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Proc. ACM SIGGRAPH*, 1997. 2

[5] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 1, 6, 7

[6] F. Hong, M. Zhang, L. Pan, Z. Cai, L. Yang, and Z. Liu. Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *ACM TOG*, 41(4):1–19, 2022. 5

[7] T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022. 6, 7

[8] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. *ACM TOG*, 34(6):1–16, 2015. 2

[9] P. Patel, C.-H. P. Huang, J. Tesch, D. T. Hoffmann, S. Tripathi, and M. J. Black. Agora: Avatars in geography optimized for regression analysis. In *Proc. CVPR*, 2021. 2

[10] 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 *Proc. CVPR*, 2019. 2

[11] 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 *Proc. ICML*, 2021. 5

[12] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. MICCAI*, 2015. 1

[13] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi. Image super-resolution via iterative refinement. *IEEE TPAMI*, 45(4):4713–4726, 2023. 1

[14] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proc. ICML*, 2015. 1

[15] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *Proc. ICLR*, 2021. 6, 7

[16] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *NeurIPS*, 2021. 5

[17] M. Worchel, R. Diaz, W. Hu, O. Schreer, I. Feldmann, and P. Eisert. Multi-view mesh reconstruction with neural deferred shading. In *Proc. CVPR*, 2022. 2