DOI: 10.1111/cgf.15177
ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2024
M. Skouras and H. Wang
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 8

Creating a 3D Mesh in A-pose from a Single Image for Character Rigging

Seunghwan Lee[1] and C. Karen Liu[1]

[1]Stanford University, USA

Figure 1: ***Left:*** *Our framework takes an input image depicting a character in an arbitrary pose.* ***Middle:*** *We transform the character in the input image to an A-pose and lift it to 3D.* ***Right:*** *The A-posed 3D mesh enables the standard automatic rigging process to create an animatable 3D character.*

Abstract

Learning-based methods for 3D content generation have shown great potential to create 3D characters from text prompts, videos, and images. However, current methods primarily focus on generating static 3D meshes, overlooking the crucial aspect of creating an animatable 3D meshes. Directly using 3D meshes generated by existing methods to create underlying skeletons for animation presents many challenges because the generated mesh might exhibit geometry artifacts or assume arbitrary poses that complicate the subsequent rigging process. This work proposes a new framework for generating a 3D animatable mesh from a single 2D image depicting the character. We do so by enforcing the generated 3D mesh to assume an A-pose, which can mitigate the geometry artifacts and facilitate the use of existing automatic rigging methods. Our approach aims to leverage the generative power of existing models across modalities without the need for new data or large-scale training. We evaluate the effectiveness of our framework with qualitative results, as well as ablation studies and quantitative comparisons with existing 3D mesh generation models.

CCS Concepts

• ***Computing methodologies*** → ***Computer vision; Rendering;***

1. introduction

Recent advances in learning-based 3D content generation have shown great potential to create 3D characters from text prompts, videos, and images [XZW*23; ZZZ*23; LYX*24; SZS*23; PJBM22; MPE*23]. Using these simple yet expressive 2D or textual input formats to create 3D characters is highly effective for downstream applications such as games, movies, and mixed reality. However, current methods primarily focus on generating static 3D meshes, overlooking the crucial aspect of creating an *animatable* 3D meshes, which limits their application in computer animation.

An *animatable* mesh is a 3D mesh that depicts the appearance and the shape of the character, while being able to deform based on the pose of its underlying skeleton. To make an arbitrary 3D mesh animatable, a common approach is to create a skeleton that drives the deformation of the 3D mesh, known as the rigging process. Unfortunately, 3D meshes generated by existing methods often exhibit artifacts such as asymmetric body parts or improper merging of body segments (e.g., partial merging of upper arms and torso). Additionally, the generated mesh may assume a pose that complicates or renders the subsequent rigging process impossible (e.g., a pose with crossed arms).

This work proposes a new framework for generating a 3D mesh from a single 2D image depicting the character, designed to facilitate the rigging process for animation. To overcome the abovemen-

tioned challenges, we enforce the generated 3D mesh to assume an *A-pose*, wherein the limbs are separate from the torso except for the area near the shoulder and hip joints. This approach mitigates the geometry artifacts and supports the use of existing automatic rigging methods [XZK*20; BP07], which expects the input mesh in an A-pose or a T-pose. As such, our problem can be redefined as follows: given an image of a bipedal character in an arbitrary pose, create a 3D mesh of that character in the A-pose.

While there are several image and 3D mesh generation models [RBL*22; BDK*23; PJBM22; QMH*23; ZZZ*23], none of them can be directly applied to our specific task. Before committing to training yet another model for this new task, we explore the idea of synergistically combining existing pretrained models effectively. This approach aims to leverage the generative power of these models across modalities without the need for developing new architectures, embeddings, losses, or, importantly, acquiring new data. To this end, we propose a framework that consists of three stages (Figure 1). First, we transform the pose in the input image to a synthesized A-pose image. Second, we construct a 3D geometry using the A-pose image. Finally, we employ an image generation model from text prompts trained on high-resolution images to enhance the texture quality of our 3D avatar. These steps involve the innovative integration and utilization of various existing pretrained image generation models from text prompts (text-to-image) or from images (image-to-image), 3D mesh generation models from images (image-to-3D), and off-the-shelf image feature prediction models, without requiring additional training data or large-scale training. We evaluate the effectiveness of our framework with qualitative results, as well as ablation studies and quantitative side-by-side comparisons with existing image-to-3d models. Animated characters can be found in supplementary videos.

2. Related Work

2.1. Neural Radiance Field for 3D Contents Generation

In the field of machine learning, notable research has been conducted on 3D content generation based on point clouds [ADMG18], signed distance fields [PFS*19], voxels [WZX*16], and neural radiance fields (NeRF) [MST*21; YYTK21; TY21]. NeRF, initially aimed at creating detailed and photorealistic 3D scenes from 2D images, has evolved into a tool for generating 3D models from sparse inputs such as single images and text prompts. To overcome the original need for numerous calibrated views, large-scale image generation models [RBL*22; SCS*22; RPG*21] have been employed to synthesize novel views replacing the calibrated views. Liu et al. [LWV*23] introduce an image-to-image diffusion model that is conditioned on camera views. Due to the inherent randomness of the diffusion process, multiple novel view synthesis methods ensuring 3D consistency have been explored [LLZ*23; LHG*23; SWY*23; LXJ*23]. These methods develop their own network models for training, which requires training datasets of 3D content, such as Objaverse [DSS*23]. In practice, learning new network models for such specific tasks often requires laborious hyper parameter tuning, the curation of training dataset, and expensive training resources. Conversely, our framework does not require training new network models for novel tasks. Instead, we employ pretrained 2D image generation models that have been trained on large-scale datasets [SBV*22; DSS*23].

The original concept of using pretrained text-to-image generation models for 3D content creations was proposed by Poole et al. [PJBM22]. They introduce score distillation sampling (SDS), an iterative process that refines an initial random NeRF to match 2D reference images depicting the target object from various angles. These reference images are not actual photos but are synthetically generated by the 2D text-to-image model. In this way, a dataset of 3D models is not required for generating 3D content, thereby opening up new possibilities for 3D generative models [WLW*23; SZS*23; GAA*22]. Our 3D model creation pipeline is aligned with the above-mentioned methods, and it achieves high-quality results with our customized diffusion models and two-phase process that focuses on geometry construction and texture refinement, respectively.

2.2. Diffusion Model Customization and Control

Ever since large-scale diffusion models have demonstrated their performance in text-to-image generation [RPG*21; RBL*22; SCS*22], the customization of the diffusion models to generate synthesized images of specific subjects in various textual contexts emerges as an important topics. A naive way to first obtain text prompts from existing image-to-text models such as CLIP [RKH*21a], and then use these text prompts for text-to-image generations. However, it often fails to get the desired output primarily because the text prompts may not be as specific as required for text-to-image models to accurately synthesize images that depict the subjects. Gal et al. [GAA*22] propose to learn text embeddings in the latent space in text encoder, which offers more degrees of freedom to the text-to-image model than those available in the natural language prompts. Fine-tuning the diffusion models for customization with a few shot of images [RLJ*23; HSW*21; VHGS19], with single images [WZJ*23], with images having multiple concepts [AAF*23; KZZ*23] has been explored. Our pipeline leverages these methods for A-posed image generation as well as 3D model creations. Furthermore, we specialize in character-specific enhancements, thereby improving the quality of synthesized A-pose images and 3D models.

In addition to the customization for generating specific subjects, research has also focused on adding controllability to the generated outputs. Zhang et al. [ZRA23] introduce network architectures integrated with diffusion models to condition the denoising process using an additional image input. Text-driven control methods, combined with in-painting techniques, have been explored to maintain originality while still exerting influence over the final image results [MHS*21; BHE23; KZL*23]. Our framework employ both the image control techniques to consistently generate A-pose images and the text-driven in-painting techniques to refine facial complexity.

2.3. Generative Animatable Characters

Research in the generation of animatable characters has predominantly focused on creating human-like avatars based on parametric body model such as SMPL [LMR*23]. Saito et al. [SSSJ20; SYMB21] introduce a framework generating custom animatable

avatars from 3D scans of humans. MP-NeRF generates multi-person novel view synthesis using sparse cameras and a multi-person SMPL template, addressing occlusion and interaction issues in dynamic scenes [CL22]. HDHumans integrates a classical deforming character template with NeRF to generate high-resolution human characters with accurate novel views and motions [HLX*23]. Text-driven animatable character generations has been explored using text-to-image diffusion model and the SDS approach [HSZ*23; KAZ*23; KLTT23; LYX*24].

In the domain of facial animations, DreamFace proposed personalized animatable faces generated from text descriptions [ZQL*23]. Qin et al [QSA*23] propose an end-to-end deep-learning approach for automatic rigging and retargeting of 3D human face models. Fundamental difference between our framework and these above-mentioned methods is that we don't rely on assuming parametric models underlying the meshes. By enabling the generation of 3D meshes without underlying parameteric models, our method can create animatable characters ranging from human-like characters to super-deformed characters at the cost of an additional rigging process [BP07; XZK*20; Mix].

3. Method

Our framework takes as input a single image of the character of interest and produces a 3D animatable character. The input image is assumed to depict a single full-body bipedal character with legs, arms, and a head visible in the image.

We propose a framework that consists of three stages. First, we transform the pose in the input image to a canonical A-pose using an image-to-image translation model. We utilize text-to-image model [RBL*22] based on diffusion models for our system's generative capabilities. Second, we construct a 3D geometry using the A-pose image. A pretrained image-to-image model capable of synthesizing novel view images is used to create a 3D mesh using score distillation sampling (SDS). The texture synthesized by the image-to-image model often lacks details and good quality. As such, we employ a text-to-image diffusion model trained on high-resolution images to boost the quality of our 3D mesh. Once we obtain a high quality mesh for the character in the A-pose, we can directly use existing rigging method to make the mesh animatable.

3.1. Image Canonicalization

Given an input image of a character in an arbitrary pose, viewed from an arbitrary camera angle in front of an arbitrary background, image canonicalization process translates the input image to a new image with the character in an A-pose, viewed from the frontal view with the background removed [LSH21].

This image-to-image translation task requires a generative model capable of synthesizing a target image that resembles the appearance of the character in the source image, conditioned on a body pose. One straightforward approach converts the input image to text using an off-the-shelf image-to-text model [RKH*21a] and then put this text to the text-to-image model, including a prompt such as "A-pose" to control the output pose. However, this method often fails to synthesize accurate A-pose images because the text prompt cannot fully capture the original image's appearance and loses crucial body proportions needed for precise A-pose generation. Instead, we employ DreamBooth [RLJ*23], which customize the pretrained text-to-image model to mimic the input image. Additionally, the body proportions are extracted using Openpose [CSWS17] features to utilize ControlNet [ZRA23], which can generate images conditioned on these pose features.

3.1.1. Text-to-image model customization using a single 2D image

We use a diffusion-based text-to-image model that learns to produce high-quality images conditioned on text prompts y by sequentially denoising a sample, starting from random noise $\epsilon \sim \mathcal{N}(0,1)$. Specifically, the model $\epsilon_\phi(\boldsymbol{x}_t, y, t)$ learns to predict noises with network parameters ϕ at diffusion step $t \in [0, 1000]$, where $\boldsymbol{x}_t$ is the noised image at the noise level t. Our goal is to finetune a pretrained text-to-image model ϵ_ϕ based on the input image $\boldsymbol{x}_0$, in order to obtain a customized diffusion model $\epsilon_{\phi_{\text{cus}}}$. We use a score estimation loss [Ryu] for finetuning:

$$L_{\text{diff}} = \mathbb{E}_{y,b,t,\epsilon} ||(\epsilon_\phi(\boldsymbol{x}_t, y, t) - \epsilon) \odot \boldsymbol{m}_b||^2. \quad (1)$$

To improve the quality of the image at areas with high complexity, we partition the character into two regions, $b = \{\text{body}, \text{face}\}$ with two individual masks $\boldsymbol{m}_b$.

The learnable variables include the parameters of the denoising network ϕ and the textual embeddings <b> corresponding to the body or face. The <b> is an embedding vector that encodes the text b. We use CLIP [RKH*21a] to encode the prompt y, constructed as "A photo of <b>" [GAA*22] and learn the textual embeddings <b> to optimize L_{diff}.

To ensure the textual concepts of <face> is aligned with the concept of <body>, we use an union-sampling technique inspired by Avrahami et al. [AAF*23]. During finetuning, we concatenate two concepts to create a merged mask $\boldsymbol{m}_b = \boldsymbol{m}_{body} \cup \boldsymbol{m}_{\text{face}}$ and a combined prompt, "A photo of <body> and <face>". The new prompt and the new mask are used to compute Equation (1) to learn the combinations of two textual concepts. We additionally use spatial cross-attention loss to ensure that each concept learns from the corresponding mask when union-sampling is applied.

For the A-pose image generation, we finetune the text-to-image model ϵ_ϕ in a two steps, suggested by [AAF*23]. We first train textual embeddings <b> with a high learning rate 5e-4, followed by finetuning the whole ϕ in low learning rate 2e-6, including the CLIP text encoder and the UNet denosing networks [RKH*21b; RFB15].

3.1.2. Kinematics extraction and reposing

We use two off-the-shelf models to predict the 2D joint locations [CSWS17] on $\boldsymbol{x}_0$ and the depth map of $\boldsymbol{x}_0$ [RLH*20]. Each 3D joint position is reconstructed by concatenating its predicted 2D joint position and the corresponding depth. From the 18 reconstructed 3D joint positions, we compute the body segment lengths and facial feature locations (Figure 3). Body symmetry is enforced by taking the average of the limb lengths and the joint angles on both side. Once a 3D stick figure is constructed in such a way, we repose the stick figure to an A-pose by forward kinematics.

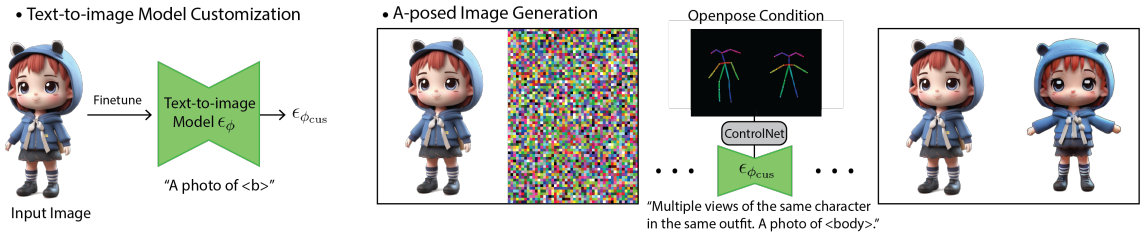


Figure 2: ***Left:*** *Text-to-Image Model Customization. We customize a pretrained text-to-image model to the input image by optimizing the model parameters and the textual embeddings.* ***Right:*** *A-pose Image Generation. We utilize pretrained pose-conditioned ControlNet to control the customized text-to-image model. We ask ControlNet to generate the character in two views at once, utilizing the input image and inpainting technique.*

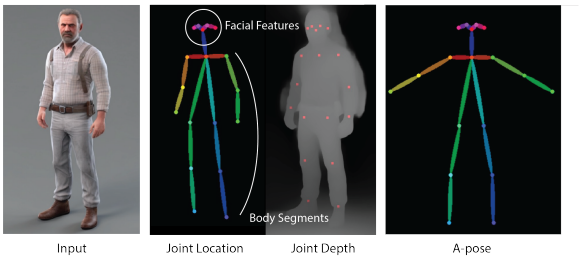


Figure 3: *We use off-the-shelf methods to estimate 2D joint locations and depth map from the input image. Concatenating these estimates, 3D joint positions, body segment lengths and facial feature locations of a 3D stick figure are reconstructed. We repose the stick figure to an A-pose using forward kinematics.*

Since the model that predicts 2D joint locations was trained on the real human data [CSWS17], it occasionally fails when applied on nonhuman characters. We manually label the 2D joint locations for those cases. Surprisingly, the A-pose image generation process described in Section 3.1.3 can still handle nonhuman characters robustly.

3.1.3. A-posed image generation

With the customized text-to-image model $\epsilon_{\phi_{cus}}$ and a stick figure in an A-pose, we use ControlNet [ZRA23] to generate a new image $\boldsymbol{x}_A$ with the character in A-pose. ControlNet is a neural network model attached to a backbone diffusion model to enable additional controllablity for text-to-image generation. We could simply use the existing ControlNet for 3D pose conditioning and swap the backbone diffusion model with our customized $\epsilon_{\phi_{cus}}$. However, it is challenging to come up with a text prompt such that ControlNet would generate consistent appearance of the desired character, even with our customized backbone diffuion model $\epsilon_{\phi_{cus}}$.

We overcome the issue using the idea of in-painting with a two-stage process (Figure 2). In the first stage, we put side-by-side the original input $\boldsymbol{x}_0 \in \mathbb{R}^{H\times W\times 3}$ and an image of noise $\epsilon \sim \mathcal{N}(0,1)$ to form a wider image $[\boldsymbol{x}_0, \epsilon] \in \mathbb{R}^{H\times 2W\times 3}$, where H and W are the height and the width of $\boldsymbol{x}_0$ respectively. Similarly, we put two stick figures side-by-side, one with original pose and one with the A-pose, and project them to an image as a condition to the ControlNet. We then ask ControlNet to in-paint the right side of the wide image with the text prompt, "Multiple views of the same character in the same outfit. A photo of <body>". With the guidance provided by the left side of the image along with the pose condition, ControlNet dutifully generates an image of the character on the right side in A-pose with the same appearance similar to $\boldsymbol{x}_0$. This trick works well for the full body, but the result lacks details in the face and hand areas, hence the second stage. We generate circle masks around the face and hands to further refine these areas. The center and radius of the masks are determined by the corresponding 2D joint positions and limb length. Similar to the work by Meng et al. [MHS*21], the refinement process is done by adding noise at the mask regions at the noise level $t = 500$ and denoising them down to $t = 0$ with the same text prompt but swapping textual embedding <body> with <face>.

3.2. Geometry Construction

The next step is to create a 3D triangle mesh from the generated A-posed image $\boldsymbol{x}_A$ (Figure 4). Recent work such as Zero123 [LWV*23] has made great advances in creating novel views from a single image. In this stage, we leverage a pretrained image-to-image, view-conditioned diffusion model ϵ_ψ to construct the 3D geometry from $\boldsymbol{x}_A$. We further exploit our customized text-to-image model $\epsilon_{\phi_{cus}}$, and off-the-shelf image feature prediction models to improve detail reconstruction of the 3D geometry.

We use a NeRF representation with learnable implicit density and albedo function [MST*21] to construct the 3D mesh. The NeRF parameters $\boldsymbol{\theta}$ are optimized with the following two loss functions.

3.2.1. Front view loss

Although we do not have a sufficient set of images that cover different camera views to train a NeRF representation, we can at least leverage the one image from the frontal view we have, $\boldsymbol{x}_A$. Given the rendered image $\boldsymbol{x}$ of the 3D representation $\boldsymbol{\theta}$ from the camera position $\boldsymbol{c} = \boldsymbol{c}_{\text{front}}$ and the differentiable rendering function $\boldsymbol{g}(\theta, \boldsymbol{c}) = \boldsymbol{x}$, we define a loss function such that the 3D geometry seen from the front view matches $\boldsymbol{x}_A$:

$$L_{\text{front}}(\boldsymbol{x}) = L_{\text{rgb}} + L_{\text{opac}} + L_{\text{depth}} + L_{\text{normal}} \tag{2}$$

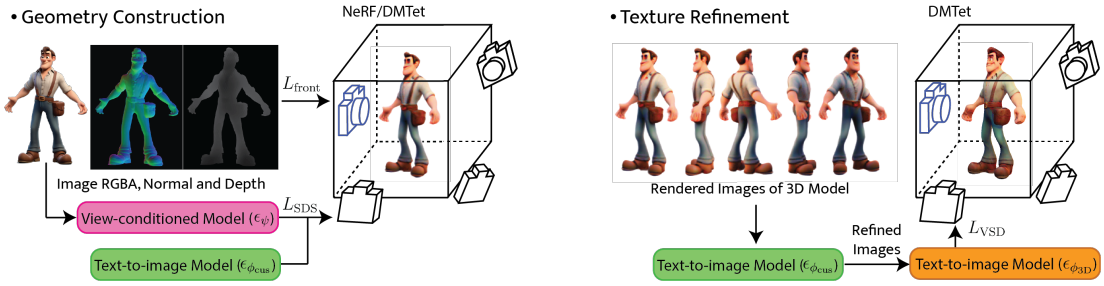


Figure 4: ***Left:*** *Geometry Construction. In addition to the information provided by the input image in the frontal view, we utilize a pretrained view-conditioned diffusion model and our customized text-to-image model to provide score distillation sampling (SDS) loss.* ***Right:*** *Texture Refinement. We fix the geometry of the reconstructed 3D model and continue to improve its texture. Images from multiple views are rendered from the 3D model and refined by our customized text-to-image model. These refined images are used to customize another 3D-aware text-to-image model, which is then used to further optimize the texture of the 3D model.*

where L_{rgb} and L_{opac} penalize pixel-wise rgb and opacity deviations between $\boldsymbol{x}$ and $\boldsymbol{x}_A$, respectively. With the predicted depth and normal maps from the off-the-shelf feature prediction models [RLH*20], L_{depth} encourages higher Pearson correlation coefficients between the depths of $\boldsymbol{x}$ and the predicted depths of $\boldsymbol{x}_A$. Similarly, L_{normal} encourages cosine similarities between the normals of $\boldsymbol{x}$ and the predicted normals of $\boldsymbol{x}_A$. The design of L_{front} closely follows the work [QMH*23; SZS*23].

3.2.2. Novel view loss

For other views which we do not have any reference images, we leverage a pretrained view-conditioned diffusion model $\epsilon_\psi(\boldsymbol{x}_t, \boldsymbol{x}_A, \boldsymbol{c}, t)$ [LWV*23] and our customized text-to-image model $\epsilon_{\phi_{\text{cus}}}(\boldsymbol{x}_t, y, t)$ A straightforward approach samples numerous synthesized novel view images using ϵ_ψ in random arbitrary views with standard methods like DDPM or DDIM [HJA20; SME20]. However, the NeRF θ fails to converge due to 3D inconsistency throughout the synthesized images. This occurs because the images are independently sampled before updating θ for 3D consistency. Instead, we use score distillation sampling (SDS), which updates θ during sampling. Consider sampling an image $\boldsymbol{x}$ from the camera position $\boldsymbol{c}$ by minimizing the loss $L(\epsilon_\psi, \boldsymbol{x}) = \mathbb{E}t, \epsilon||\epsilon\psi(\boldsymbol{x}_t, \boldsymbol{x}_A, \boldsymbol{c}, t) - \epsilon||^2$. By following the gradient of the loss with respect to $\boldsymbol{x}$, the minimization aligns with the score of the data distribution formed by ϵ_ψ to obtain an optimal image $\boldsymbol{x}^*$ that meets the conditions $\boldsymbol{x}_A$ and $\boldsymbol{c}$. In our problem, the image $\boldsymbol{x} = g(\theta, \boldsymbol{c})$ is the rendering of the NeRF with the volumetric renderer g, allowing us to update θ by the chain rule:

$$\nabla_\theta L_{\text{SDS}}(\epsilon_\psi, \boldsymbol{x}) = \mathbb{E}_{\boldsymbol{c},t,\epsilon}\left[w(t)[\epsilon_\psi(\boldsymbol{x}_t, \boldsymbol{x}_A, \boldsymbol{c}, t) - \epsilon]\frac{\partial \epsilon_\psi}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial \theta}\right] \quad (3)$$

where $w(t)$ is the weight depending on the noise level t. The noise residual $\epsilon_\psi(\boldsymbol{x}_t, \boldsymbol{x}_A, \boldsymbol{c}, t) - \epsilon$ estimates the update direction for the current rendered image $\boldsymbol{x}$ toward higher data density region of the frozen network ϵ_ψ. In practice, we omit computing the U-Net Jacobian term $\frac{\partial \epsilon_\psi}{\partial \boldsymbol{x}}$ for computational efficiency. For more details on SDS, please refer to the original paper [PJBM22].

We also exploit the customized text-to-image model $\epsilon_{\phi_{\text{cus}}}(\boldsymbol{x}_t, y, t)$ (Section 3.1). Our SDS loss linearly combines both pretrained diffusion models with a weight w_{cus}. As such, our final loss function is defined as follows:

$$L = \begin{cases} L_{\text{front}}(\boldsymbol{x}) & \text{if } \boldsymbol{c} = \boldsymbol{c}_{\text{front}}, \\ L_{\text{SDS}}(\epsilon_\psi, \boldsymbol{x}) + w_{\text{cus}} L_{\text{SDS}}(\epsilon_{\phi_{\text{cus}}}, \boldsymbol{x}) & \text{otherwise} \end{cases} \quad (4)$$

During training, we sample $\boldsymbol{c}$ randomly from a predefined range of angles and elevations. Additionally, we use classifier free guidance (CFG) to steer the update direction toward conditioned samples by adding $w_{\text{CFG}}(\epsilon_\phi(\boldsymbol{x}_t, y, t) - \epsilon_\phi(\boldsymbol{x}_t, \varnothing, t))$ to $\epsilon_\phi(\boldsymbol{x}_t, y, t)$, where $w_{\text{CFG}} = 100$ is the weight determining how much we emphasize the conditioning y [HS22].

Finally, we improve the geometry quality by switching to DMTet [SGY*21] from NeRF in the middle of the optimization process. Extracting explicit triangle meshes directly from an implicit neural field using marching cube algorithm [CDH*87] results in poor mesh quality. On the other hand, DMTet, which is based on tetrahedralized grids with nodes containing signed distances, is capable of generating explicit iso-surface triangle mesh. With a differentiable renderer [LHK*20] for the triangle meshes, we can continue to optimize the 3D shape in DMTet representation. While DMTet improves the final mesh quality, we cannot use it at the beginning of the optimization because DMTet requires an initial geometry with proper topology [CCJJ23].

3.3. Texture Refinement

The method described above generates 3D models with high quality geometry, but the texture is noticeably blurry due to several reasons. First, we use low-resolution for rendering NeRF models because high-resolution NeRF is computationally costly and requires excessive GPU memory. Second, the view-conditioned model ϵ_ψ is trained on low-resolution image dataset (256×256). Lastly, the SDS loss empirically requires higher CFG weights for convergence than the weight ranges (3.0 to 7.5) used for ancestral sampling [HJA20; SME20]. To overcome these issues, this final refinement step only optimizes appearance with fixed geometry, using a text-to-image model to distill scores on high-resolution images (512×512). In addition, we utilize variational score distillation (VSD) to allow for smaller CFG weights [WLW*23].

We would like to remove the dependency to ϵ_ψ which is trained on low-resolution images, but depending solely on the customized model $\epsilon_{\phi_{cus}}$ to provide SDS loss without the view-conditioned ϵ_ψ can be harmful since $\epsilon_{\phi_{cus}}$ is customized on the single front view 2D image; we found that optimizing θ without 3D priors creates multiple faces on the character even though the geometry is fixed. As such, we opt to customize another text-to-image diffusion model that is 3D-aware for providing distillation loss to improve the texture of the 3D model (Figure 4).

3.3.1. Customizing a 3D-aware diffusion model

Inspired by the work [RKP*23], we learned a customized 3D-awear text-to-image model $\phi = \phi_{3D}$ by finetuning a pretrained text-to-image model to a set of high-quality images with multiple views. We render images of our current 3D model from 18 random camera views and refine them using our customized text-to-image diffusion model $\epsilon_{\phi_{cus}}$. The image refinement is done by setting the noise level to $t = 500$ [MHS*21] and ask $\epsilon_{\phi_{cus}}$ to denoise low-quality rendered images. When generating these images, we also exploit the normal-map-conditioned ControlNet to help preserve original geometry. The normal maps can be extracted using the existing model [RLH*20]. The refined images are used to train the customized $\epsilon_{\phi_{3D}}$ using standard diffusion loss $L = \mathbb{E}_{t,y,\epsilon}||\epsilon_\phi(\boldsymbol{x}_t, \boldsymbol{c}, y, t) - \epsilon||^2$. This render-and-refine process allows to learn 3D-aware text-to-image diffusion model while keeping the quality as high as ϕ_{cus}.

3.3.2. Optimizing texture of the 3D models

To utilize $\epsilon_{\phi_{3D}}$ for texture optimizaiton, we construct a VSD loss as proposed in the work [WLW*23], which has shown to be able to use standard range of w_{CFG} for image generation. The VSD loss suggests learning the distribution of the 3D models $\mu(\theta|y)$, instead of learning θ. The goal is to minimize KL divergence between the prior data distribution $p_{\phi_{3D}}(\boldsymbol{x}|y)$ and the data distributions among the 3D models $q_\mu(\boldsymbol{x}|\boldsymbol{c})$. With the model $\epsilon_\mu(\boldsymbol{x}_t, \boldsymbol{c}, y, t)$ that learns q_μ by standard diffusion loss, the proposed update rule for θ uses the gradient of the VSD loss as follows:

$$\nabla_\theta L_{VSD}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{c},t,\epsilon}\left[w(t)[\epsilon_{\phi_{3D}}(\boldsymbol{x}_t, y, t) - \epsilon_\mu(\boldsymbol{x}_t, \boldsymbol{c}, y, t)]\frac{\partial \boldsymbol{x}}{\partial \theta}\right]. \quad (5)$$

which replace ϵ in the Equation (3) to ϵ_μ, providing more elaborate information for the current 3D model distributions than just pure noise. Considering ancestral samplings [HJA20; SME20], where the sample $\boldsymbol{x}_0$ is sampled from the sequence of Markov chain $p_\phi(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y)$, the VSD loss acts in similar way in a sense that the score is distilled between target prior data distribution and current distribution. Hence, we are able to use standard CFG weights used for standard image samplings. In our experiments, we use $w_{CFG} = 3.0$ for the VSD loss.

The VSD loss acts in similar way as ancestral samplings [HJA20; SME20], in a sense that the score is estimated not from the gaussian noise, but from the current data distribution. Hence it allows standard range of the CFG weights in $\epsilon_{\phi_{3D}}$.

Table 1: *Pretrained models used in our framework.*

Pretrained Model	URL (https://huggingface.co)
Stable Diffusion 1.5	runwayml/stable-diffusion-v1-5
Stable Diffusion 2.1	stabilityai/stable-diffusion-2-1
Stable Diffusion XL	stabilityai/stable-diffusion-xl-base-1.0
ControlNet	lllyasviel/ControlNet-v1-1
Stable Zero123	stabilityai/stable-zero123

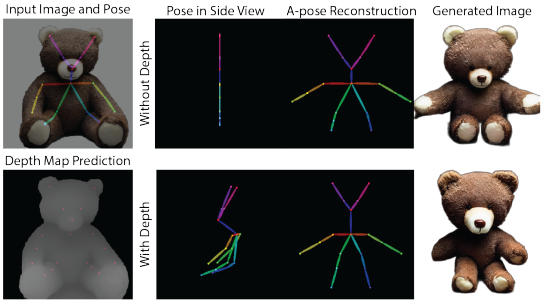


Figure 5: *Ablation study of depth map for A-pose reconstruction.*

3.4. Character Rig Generation

Now that we obtain a 3D mesh of the character in the A-pose, we can enjoy a streamlined rigging process provided by commercial software. We use Adobe Mixamo^{TM}, a free software that requires minimal intervention from the user for generating a skeleton and skinning weights for the A-posed mesh. Each rigging process takes under two minutes because the user only needs to provide a few clicks on the mesh to indicate the locations of key joints. See the accompanying video for qualitative evaluation.

4. Implementation

Our framework relies on many pretrained models listed in Table 1. We use hugging face diffusers to load pretrained network weights ϕ and ψ and their architectures [vPPL*22]. xformers [LML*22] is used to accelerate the inference time of transformer architectures and 3D position encoding proposed by InstanceNGP [MESK22] is used to accelerate the optimization of the 3D model parameters θ. threestudio, a unified framework for 3D content generations [GLS*23] is used for implementing the loss functions in Equations (2) and (3).

We use NVidia A5000 24GB GPU with batch size 1 in all our experiments. The optimization and finetuning is done by the Adam optimizer [KB14]. For each example, A-pose image generation roughly takes 7 minutes to finetune $\epsilon_{\phi_{cus}}$, followed by an additional 10 minutes for the image generation. We generate 32 samples using standard ancestral sampling [SME20], and choose one of them manually. For geometry construction, we iterate 10000 epochs with NeRF and additional 5000 epochs with DMTet, taking 70 minutes on average to produce a triangle mesh. For texture refinement, we iterate 10000 epochs which takes on average 60 minutes. The final rigging step only takes 2 minutes on average, thanks to the near-automatic process provided by Adobe Mixamo^{TM}.

Figure 6: *Without concatenating input image when in-painting the A-pose image, DB-CN produces images with significant differences in appearance from the input images.*

Table 2: *Analyses on A-pose generation.*

	CLIP Similarity	CLIP Score
Ours	0.9639 ± 0.0065	**24.8217** ± 0.4251
Ours Without Face Refinement	**0.9708** ± 0.0066	23.5377 ± 0.6733
DB-CN	0.9376 ± 0.0185	22.7274 ± 1.1282

5. Evaluation

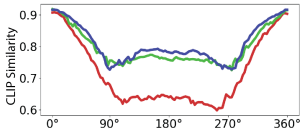
We evaluate our system with 17 character images varying in body proportions, anatomy, and poses. Our system is agnostic to the source of images. We have tested our system on images from internet [GLS*23] or from open-sourced image generation models [PEL*23].

Below we describe the quantitative evaluation on each stage of the method with different metrics and baselines. For qualitative evaluation, please see the accompanying video and the supplementary document.

5.1. Evaluation of A-pose image generation

We compare our A-pose image generation with a baseline that combines DreamBooth and ControlNet, since either of them alone cannot achieve our task. The baseline, DB-CN, customizes a pretrained Stable Diffusion model to the input image $\boldsymbol{x}_0$ using the technique proposed by DreamBooth and use this customized diffusion model as the backbone image generator for the pose-conditioned ControlNet. The main difference between DB-CN to our method is that DB-CN directly generate the A-pose image using the customized text prompt, while our method asks the ControlNet to generate a concatenated image with the same characters from two views

We evaluate our A-pose image generations with an off-the-shelf image-to-text CLIP model [RKH*21a]. The model provides us an image embedding useful for semantic evaluation because they are trained on a large-scale image-text paired dataset. We compare the cosine similarity of the embedding between the input and the generated images to quantitatively evaluate the semantic similarity.



	Similarity	Saturation
Ours	**0.808**±0.061	16.343±5.907
M123	0.796±0.058	**20.936**±6.256
DG	0.715±0.098	16.549±6.647

Figure 7: ***left****: CLIP similarity.* ***right*** *Average similarity and saturation of the rendered images of 3D models.*

Table 2 shows that our method, with or without face refinement option, outperforms DB-CN baseline. This quantitative result is consistent with our visual inspection as shown in Figure 6. However we also observe a small amount of decrease in similarity when face refinement is applied. For some characters, the face refinement might change accessories around the head, such as necklaces, resulting in lower CLIP similarity scores. However, we still opt to include the face refinement as it dramatically increase the quality of faces.

We also compare the general quality of the images generated by our method and by DB-CN. We compute the CLIP score that measures the closeness between an image and the prompt "A photo of a high-quality bipedal character.". The result in Table 2 shows that our method generates images with higher quality according to the CLIP model.

We also conduct ablation study to assess the importance of extracting depth maps for constructing a stick figure (Figure 5). We find that depth map extracted from an off-the-shelf feature prediction model [RLH*20] is especially beneficial for the characters with non-trivial poses. For example, without the depth map, an image with a sitting teddy bear results in incorrect body proportions. As such, ControlNet would generate a sitting pose instead of an A-pose as the length of the lower body is shorter than the length of the upper body. Using the depth map to reconstruct the stick figure, we are able to robustly generate A-pose images for all of our 17 characters.

5.2. Evaluation of 3D model generation

5.2.1. Qualitative Comparisons

We compare the 3D models generated by our method with those generated by previous works: Magic123 (M123), Wonder3D (W3D), DreamGaussian (DG) and commercial assets from a company called CSM [QMH*23; LGL*23; TRZ*23; CSM]. In general, our method works well for a wide range of characters, with no obvious misalignment in shapes, colors, and semantics.

The textures and geometry generated by W3D tend to lack details. This might be caused by the fact that W3D does not use score distillation techniques. DG uses Gaussian splatting as the 3D representation [KKLD23]. We observe that DG often generates artifacts in geometry (most obviously in the firefighter character). Possible cause might be that only the view-conditioned diffusion model is

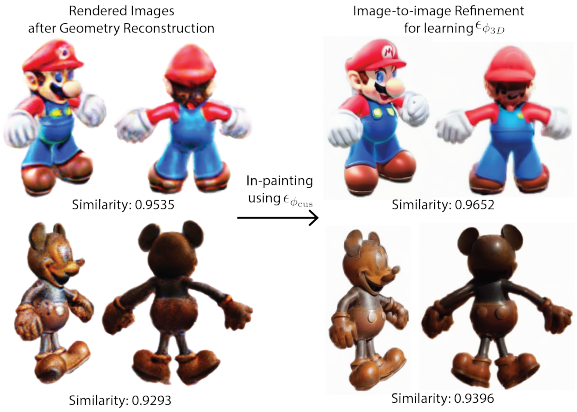


Figure 8: *The quality of rendered images from the 3D model before texture refinement is poor. However, after refinement using $\epsilon_{\phi_{cus}}$, the images look a lot sharper with slightly higher CLIP similarity scores.*

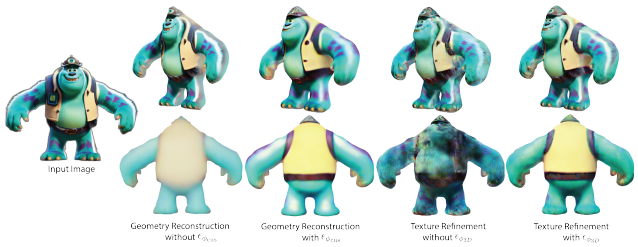


Figure 9: *Ablation studies on the impact of customized text-to-image diffusion models during geometry construction and texture refinement*

used in the SDS loss, as opposed to adding the text-to-image diffusion model, shown in our Equation 4. M123 by far generates the best textures because they utilize both view-conditioned and text-to-image diffusion models with the text embedding learned from textual inversion techniques [GAA*22]. However, we notice misalignment in geometry, such as multiple faces shown on the tree-like character and eyes merging into nose shown on the bear and the mouse characters. The reasons might be that only text embedding was learned without finetuning the text-to-image model. Additionally, we notice that M123 produces textures that are overly saturated (Figure 7). We quantitatively evaluate the level of saturation by converting RGB of rendered images to HSV. The results distinctively show that M123 is overly saturated. We suspect that they use higher CFG weights for the SDS loss.

5.2.2. Quantitative Comparisons

We evaluate our results with the CLIP [RKH*21a] to measure semantic similarity between the 3D models and the input 2D images (Figure 7). We do so by rendering 128 images varying in view angles for each 3D model and compare the cosine similarity of their CLIP embeddings with that of the input image. Our method outperforms M123 and DG on average and in all viewing directions, though the difference between M123 and ours is small. However, we notice that M123 sometimes creates multiple faces on the characters. The erroneous faces may inadvertently increase their semantic similarity values.

5.2.3. Ablation Study

We first study the importance of adding our customized text-to-diffusion model $\epsilon_{\phi_{cus}}$ in the SDS loss (Equation 4). Figure 9 shows that creating geometry only with the view-conditioned model ϵ_{ψ} results in blurred texture and adding $\epsilon_{\phi_{cus}}$ dramatically improves the texture quality even with a small weight $w_{cus} = 0.001$.

Texture refinement without training another model $\epsilon_{\phi_{3D}}$ often creates multiple faces on the back view even with a fixed geometry. Figure 9 shows that with a 3D-aware text-to-image model, our framework is able to learn 3D models with highly-detailed textures.

We also evaluate the effectiveness of the render-and-refine technique to generate higher quality images for training $\epsilon_{\phi_{3D}}$. Figure 8 shows that the texture quality improves noticeably after image refinement using $\epsilon_{\phi_{cus}}$. The average CLIP similarity before and after the refinement also increases by a small amount.

6. Conclusion

We present a framework, capable of creating a 3D animatable character from a single image. We exploit the generative power of pretrained models to reconstruct A-pose figures, generate A-pose images, and create 3D models ready for the rigging process, without the need for new loss functions, new architectures, or most importantly, additional data.

Our framework has several limitations. Firstly, the selection of A-pose images is a manual process from a batch of candidates. This necessitates human intervention before 3D model creation. Secondly, while leveraging pretrained models is advantageous, it can also inherit biases or weaknesses. For example, we found that using ControlNet trained on the OpenPose dataset, which is based on real human data, often fails to accurately control poses when the input characters have body proportions that significantly differ from typical human proportions. Moreover, the generated images frequently display textures influenced by lighting and ambient occlusions. Such undesired shading effects appear in texture maps and can be potentially fixed by new methods that extract true colors from images. Lastly, our framework struggles with skeletal rigging in characters with long hair, often resulting in the hair merging into the torso. The movements of the hairs complicate the standard skeletal-driven rigging process. A potential future direction is to separate generation of 3D hair and full-body models and use a simulation-based rigging process for the hair.

References

[AAF*23] AVRAHAMI, OMRI, ABERMAN, KFIR, FRIED, OHAD, et al. "Break-A-Scene: Extracting Multiple Concepts from a Single Image". *SIGGRAPH Asia 2023 Conference Papers*. SA '23. , Sydney, NSW, Australia, Association for Computing Machinery, 2023. ISBN: 9798400703157. DOI: 10.1145/3610548.3618154. URL: https://doi.org/10.1145/3610548.3618154 2, 3.

[ADMG18] ACHLIOPTAS, PANOS, DIAMANTI, OLGA, MITLIAGKAS, IOANNIS, and GUIBAS, LEONIDAS. "Learning representations and generative models for 3d point clouds". *International conference on machine learning*. PMLR. 2018, 40–49 2.

[BDK*23] BLATTMANN, ANDREAS, DOCKHORN, TIM, KULAL, SUMITH, et al. "Stable video diffusion: Scaling latent video diffusion models to large datasets". *arXiv preprint arXiv:2311.15127* (2023) 2.

[BHE23] BROOKS, TIM, HOLYNSKI, ALEKSANDER, and EFROS, ALEXEI A. "Instructpix2pix: Learning to follow image editing instructions". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, 18392–18402 2.

[BP07] BARAN, ILYA and POPOVIĆ, JOVAN. "Automatic rigging and animation of 3d characters". *ACM Transactions on graphics (TOG)* 26.3 (2007), 72–es 2, 3.

[CCJJ23] CHEN, RUI, CHEN, YONGWEI, JIAO, NINGXIN, and JIA, KUI. "Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation". *arXiv preprint arXiv:2303.13873* (2023) 5.

[CDH*87] CLINE, HARVEY E, DUMOULIN, CL, HART JR, HR, et al. "3D reconstruction of the brain from magnetic resonance images using a connectivity algorithm". *Magnetic Resonance Imaging* 5.5 (1987), 345–352 5.

[CL22] CHAO, XIAN JIN and LEUNG, HOWARD. "MP-NeRF: Neural Radiance Fields for Dynamic Multi-person synthesis from Sparse Views". *Computer Graphics Forum*. Vol. 41. 8. Wiley Online Library. 2022, 317–325 3.

[CSM] CSM. *CSM*. URL: https://www.csm.ai 7.

[CSWS17] CAO, ZHE, SIMON, TOMAS, WEI, SHIH-EN, and SHEIKH, YASER. "Realtime multi-person 2d pose estimation using part affinity fields". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, 7291–7299 3, 4.

[DSS*23] DEITKE, MATT, SCHWENK, DUSTIN, SALVADOR, JORDI, et al. "Objaverse: A universe of annotated 3d objects". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, 13142–13153 2.

[GAA*22] GAL, RINON, ALALUF, YUVAL, ATZMON, YUVAL, et al. "An image is worth one word: Personalizing text-to-image generation using textual inversion". *arXiv preprint arXiv:2208.01618* (2022) 2, 3, 8.

[GLS*23] GUO, YUAN-CHEN, LIU, YING-TIAN, SHAO, RUIZHI, et al. *threestudio: A unified framework for 3D content generation*. https://github.com/threestudio-project/threestudio. 2023 6, 7.

[HJA20] HO, JONATHAN, JAIN, AJAY, and ABBEEL, PIETER. "Denoising diffusion probabilistic models". *Advances in neural information processing systems* 33 (2020), 6840–6851 5, 6.

[HLX*23] HABERMANN, MARC, LIU, LINGJIE, XU, WEIPENG, et al. "Hdhumans: A hybrid approach for high-fidelity digital humans". *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 6.3 (2023), 1–23 3.

[HS22] HO, JONATHAN and SALIMANS, TIM. "Classifier-free diffusion guidance". *arXiv preprint arXiv:2207.12598* (2022) 5.

[HSW*21] HU, EDWARD J, SHEN, YELONG, WALLIS, PHILLIP, et al. "Lora: Low-rank adaptation of large language models". *arXiv preprint arXiv:2106.09685* (2021) 2.

[HSZ*23] HUANG, XIN, SHAO, RUIZHI, ZHANG, QI, et al. "Humannorm: Learning normal diffusion model for high-quality and realistic 3d human generation". *arXiv preprint arXiv:2310.01406* (2023) 3.

[KAZ*23] KOLOTOUROS, NIKOS, ALLDIECK, THIEMO, ZANFIR, ANDREI, et al. "DreamHuman: Animatable 3D Avatars from Text". *arXiv preprint arXiv:2306.09329* (2023) 3.

[KB14] KINGMA, DIEDERIK P and BA, JIMMY. "Adam: A method for stochastic optimization". *arXiv preprint arXiv:1412.6980* (2014) 6.

[KKLD23] KERBL, BERNHARD, KOPANAS, GEORGIOS, LEIMKÜHLER, THOMAS, and DRETTAKIS, GEORGE. "3D Gaussian Splatting for Real-Time Radiance Field Rendering". *ACM Transactions on Graphics* 42.4 (July 2023). URL: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/ 7.

[KLTT23] KAO, SHIU-HONG, LIU, XINHANG, TAI, YU-WING, and TANG, CHI-KEUNG. "Deceptive-Human: Prompt-to-NeRF 3D Human Generation with 3D-Consistent Synthetic Images". *arXiv preprint arXiv:2311.16499* (2023) 3.

[KZL*23] KAWAR, BAHJAT, ZADA, SHIRAN, LANG, ORAN, et al. "Imagic: Text-based real image editing with diffusion models". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, 6007–6017 2.

[KZZ*23] KUMARI, NUPUR, ZHANG, BINGLIANG, ZHANG, RICHARD, et al. "Multi-concept customization of text-to-image diffusion". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, 1931–1941 2.

[LGL*23] LONG, XIAOXIAO, GUO, YUAN-CHEN, LIN, CHENG, et al. "Wonder3d: Single image to 3d using cross-domain diffusion". *arXiv preprint arXiv:2310.15008* (2023) 7.

[LHG*23] LIN, YUKANG, HAN, HAONAN, GONG, CHAOQUN, et al. "Consistent123: One image to highly consistent 3d asset using case-aware diffusion priors". *arXiv preprint arXiv:2309.17261* (2023) 2.

[LHK*20] LAINE, SAMULI, HELLSTEN, JANNE, KARRAS, TERO, et al. "Modular Primitives for High-Performance Differentiable Rendering". *ACM Transactions on Graphics* 39.6 (2020) 5.

[LLZ*23] LIU, YUAN, LIN, CHENG, ZENG, ZIJIAO, et al. "SyncDreamer: Generating Multiview-consistent Images from a Single-view Image". *arXiv preprint arXiv:2309.03453* (2023) 2.

[LML*22] LEFAUDEUX, BENJAMIN, MASSA, FRANCISCO, LISKOVICH, DIANA, et al. *xFormers: A modular and hackable Transformer modelling library*. https://github.com/facebookresearch/xformers. 2022 6.

[LMR*23] LOPER, MATTHEW, MAHMOOD, NAUREEN, ROMERO, JAVIER, et al. "SMPL: A skinned multi-person linear model". *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 2023, 851–866 2.

[LSH21] LEE, MIN SEOK, SHIN, WOOSEOK, and HAN, SUNG WON. "TRACER: Extreme Attention Guided Salient Object Tracing Network". *arXiv preprint arXiv:2112.07380* (2021) 3.

[LWV*23] LIU, RUOSHI, WU, RUNDI, VAN HOORICK, BASILE, et al. "Zero-1-to-3: Zero-shot one image to 3d object". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, 9298–9309 2, 4, 5.

[LXJ*23] LIU, MINGHUA, XU, CHAO, JIN, HAIAN, et al. "One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization". *arXiv preprint arXiv:2306.16928* (2023) 2.

[LYX*24] LIAO, TINGTING, YI, HONGWEI, XIU, YULIANG, et al. "TADA! Text to Animatable Digital Avatars". *International Conference on 3D Vision (3DV)*. 2024 1, 3.

[MESK22] MÜLLER, THOMAS, EVANS, ALEX, SCHIED, CHRISTOPH, and KELLER, ALEXANDER. "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding". *ACM Trans. Graph.* 41.4 (July 2022), 102:1–102:15. DOI: 10.1145/3528223.3530127. URL: https://doi.org/10.1145/3528223.3530127 6.

[MHS*21] MENG, CHENLIN, HE, YUTONG, SONG, YANG, et al. "Sdedit: Guided image synthesis and editing with stochastic differential equations". *arXiv preprint arXiv:2108.01073* (2021) 2, 4, 6.

[Mix] MIXAMO. *Mixamo*. URL: https://www.mixamo.com 3.

[MPE*23] MENDIRATTA, MOHIT, PAN, XINGANG, ELGHARIB, MOHAMED, et al. "Avatarstudio: Text-driven editing of 3d dynamic human head avatars". *ACM Transactions on Graphics (TOG)* 42.6 (2023), 1–18 1.

[MST*21] MILDENHALL, BEN, SRINIVASAN, PRATUL P, TANCIK, MATTHEW, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis". *Communications of the ACM* 65.1 (2021), 99–106 2, 4.

[PEL*23] PODELL, DUSTIN, ENGLISH, ZION, LACEY, KYLE, et al. "Sdxl: Improving latent diffusion models for high-resolution image synthesis". *arXiv preprint arXiv:2307.01952* (2023) 7.

[PFS*19] PARK, JEONG JOON, FLORENCE, PETER, STRAUB, JULIAN, et al. "Deepsdf: Learning continuous signed distance functions for shape representation". *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, 165–174 2.

[PJBM22] POOLE, BEN, JAIN, AJAY, BARRON, JONATHAN T, and MILDENHALL, BEN. "DreamFusion: Text-to-3D using 2D Diffusion". *The Eleventh International Conference on Learning Representations*. 2022 1, 2, 5.

[QMH*23] QIAN, GUOCHENG, MAI, JINJIE, HAMDI, ABDULLAH, et al. "Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors". *arXiv preprint arXiv:2306.17843* (2023) 2, 5, 7.

[QSA*23] QIN, DAFEI, SAITO, JUN, AIGERMAN, NOAM, et al. "Neural face rigging for animating and retargeting facial meshes in the wild". *ACM SIGGRAPH 2023 Conference Proceedings*. 2023, 1–11 3.

[RBL*22] ROMBACH, ROBIN, BLATTMANN, ANDREAS, LORENZ, DOMINIK, et al. "High-Resolution Image Synthesis With Latent Diffusion Models". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, 10684–10695 2, 3.

[RFB15] RONNEBERGER, OLAF, FISCHER, PHILIPP, and BROX, THOMAS. "U-net: Convolutional networks for biomedical image segmentation". *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, 234–241 3.

[RKH*21a] RADFORD, ALEC, KIM, JONG WOOK, HALLACY, CHRIS, et al. "Learning transferable visual models from natural language supervision". *International conference on machine learning*. PMLR. 2021, 8748–8763 2, 3, 7, 8.

[RKH*21b] RADFORD, ALEC, KIM, JONG WOOK, HALLACY, CHRIS, et al. "Learning transferable visual models from natural language supervision". *International conference on machine learning*. PMLR. 2021, 8748–8763 3.

[RKP*23] RAJ, AMIT, KAZA, SRINIVAS, POOLE, BEN, et al. "Dreambooth3d: Subject-driven text-to-3d generation". *arXiv preprint arXiv:2303.13508* (2023) 6.

[RLH*20] RANFTL, RENÉ, LASINGER, KATRIN, HAFNER, DAVID, et al. "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer". *IEEE transactions on pattern analysis and machine intelligence* 44.3 (2020), 1623–1637 3, 5–7.

[RLJ*23] RUIZ, NATANIEL, LI, YUANZHEN, JAMPANI, VARUN, et al. "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, 22500–22510 2, 3.

[RPG*21] RAMESH, ADITYA, PAVLOV, MIKHAIL, GOH, GABRIEL, et al. "Zero-shot text-to-image generation". *International Conference on Machine Learning*. PMLR. 2021, 8821–8831 2.

[Ryu] RYU, SIMO. *Low-rank adaptation for fast text-to-image diffusion fine-tuning*. URL: https://github.com/cloneofsimo/lora 3.

[SBV*22] SCHUHMANN, CHRISTOPH, BEAUMONT, ROMAIN, VENCU, RICHARD, et al. "Laion-5b: An open large-scale dataset for training next generation image-text models". *Advances in Neural Information Processing Systems* 35 (2022), 25278–25294 2.

[SCS*22] SAHARIA, CHITWAN, CHAN, WILLIAM, SAXENA, SAURABH, et al. "Photorealistic text-to-image diffusion models with deep language understanding". *Advances in Neural Information Processing Systems* 35 (2022), 36479–36494 2.

[SGY*21] SHEN, TIANCHANG, GAO, JUN, YIN, KANGXUE, et al. "Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis". *Advances in Neural Information Processing Systems* 34 (2021), 6087–6101 5.

[SME20] SONG, JIAMING, MENG, CHENLIN, and ERMON, STEFANO. "Denoising diffusion implicit models". *arXiv preprint arXiv:2010.02502* (2020) 5, 6.

[SSSJ20] SAITO, SHUNSUKE, SIMON, TOMAS, SARAGIH, JASON, and JOO, HANBYUL. "Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, 84–93 2.

[SWY*23] SHI, YICHUN, WANG, PENG, YE, JIANGLONG, et al. "Mvdream: Multi-view diffusion for 3d generation". *arXiv preprint arXiv:2308.16512* (2023) 2.

[SYMB21] SAITO, SHUNSUKE, YANG, JINLONG, MA, QIANLI, and BLACK, MICHAEL J. "SCANimate: Weakly supervised learning of skinned clothed avatar networks". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, 2886–2897 2.

[SZS*23] SUN, JINGXIANG, ZHANG, BO, SHAO, RUIZHI, et al. "Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior". *arXiv preprint arXiv:2310.16818* (2023) 1, 2, 5.

[TRZ*23] TANG, JIAXIANG, REN, JIAWEI, ZHOU, HANG, et al. "Dreamgaussian: Generative gaussian splatting for efficient 3d content creation". *arXiv preprint arXiv:2309.16653* (2023) 7.

[TY21] TREVITHICK, ALEX and YANG, BO. "Grf: Learning a general radiance field for 3d representation and rendering". *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 15182–15192 2.

[VHGS19] VON OSWALD, JOHANNES, HENNING, CHRISTIAN, GREWE, BENJAMIN F, and SACRAMENTO, JOÃO. "Continual learning with hypernetworks". *arXiv preprint arXiv:1906.00695* (2019) 2.

[vPPL*22] Von PLATEN, PATRICK, PATIL, SURAJ, LOZHKOV, ANTON, et al. *Diffusers: State-of-the-art diffusion models*. https://github.com/huggingface/diffusers. 2022 6.

[WLW*23] WANG, ZHENGYI, LU, CHENG, WANG, YIKAI, et al. "ProlificDreamer: High-Fidelity and Diverse Text-to-3D Generation with Variational Score Distillation". *arXiv preprint arXiv:2305.16213* (2023) 2, 5, 6.

[WZJ*23] WEI, YUXIANG, ZHANG, YABO, JI, ZHILONG, et al. "Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation". *arXiv preprint arXiv:2302.13848* (2023) 2.

[WZX*16] WU, JIAJUN, ZHANG, CHENGKAI, XUE, TIANFAN, et al. "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling". *Advances in neural information processing systems* 29 (2016) 2.

[XZK*20] XU, ZHAN, ZHOU, YANG, KALOGERAKIS, EVANGELOS, et al. "Rignet: Neural rigging for articulated characters". *arXiv preprint arXiv:2005.00559* (2020) 2, 3.

[XZW*23] XU, YUELANG, ZHANG, HONGWEN, WANG, LIZHEN, et al. "LatentAvatar: Learning Latent Expression Code for Expressive Neural Head Avatar". *ACM SIGGRAPH 2023 Conference Proceedings*. 2023 1.

[YYTK21] YU, ALEX, YE, VICKIE, TANCIK, MATTHEW, and KANAZAWA, ANGJOO. "pixelnerf: Neural radiance fields from one or few images". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, 4578–4587 2.

[ZQL*23] ZHANG, LONGWEN, QIU, QIWEI, LIN, HONGYANG, et al. "DreamFace: Progressive Generation of Animatable 3D Faces under Text Guidance". *ACM Transactions on Graphics (TOG)* 42.4 (2023), 1–16 3.

[ZRA23] ZHANG, LVMIN, RAO, ANYI, and AGRAWALA, MANEESH. *Adding Conditional Control to Text-to-Image Diffusion Models*. 2023 2–4.

[ZZZ*23] ZHENG, ZERONG, ZHAO, XIAOCHEN, ZHANG, HONGWEN, et al. "AvatarRex: Real-time Expressive Full-body Avatars". *ACM Transactions on Graphics (TOG)* 42.4 (2023) 1, 2.