Snapmoji: Instant Generation of Animatable Dual-Stylized Avatars

Eric Ming Chen[1,2*] Di Liu[1,3*] Sizhuo Ma[1] Michael Vasilkovsky[1] Bing Zhou[1] Qiang Gao[1]
Wenzhou Wang[1] Jiahao Luo[1,4] Dimitris N. Metaxas[3] Vincent Sitzmann[2] Jian Wang[1]
[1]Snap Inc. [2]MIT [3]Rutgers University [4]University of California, Santa Cruz

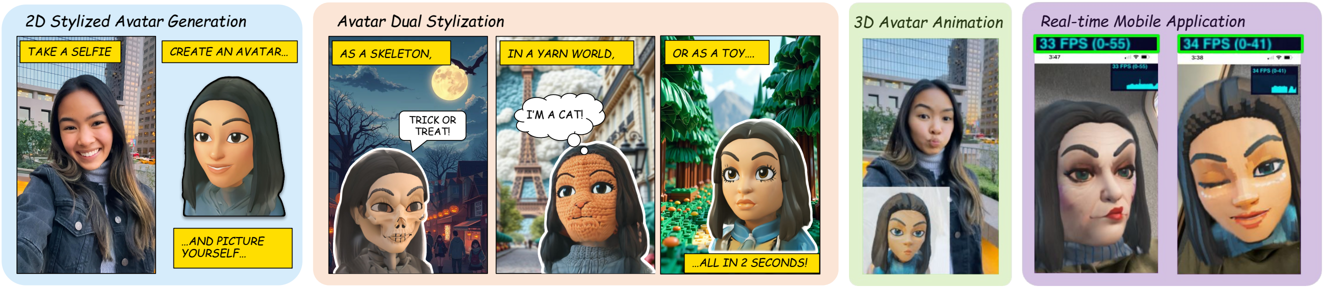

Figure 1. We introduce **Snapmoji**, a system that can instantly generate animatable dual-stylized avatars. Our dual stylization process reimagines a base avatar in various artistic styles, enabling users to visualize themselves in diverse scenarios and create personalized stories. Snapmoji accomplishes the selfie-to-avatar conversion in just 0.9 seconds. Equipped with a mobile face tracker, our approach also enables expression animation in real time, at 30 to 40 FPS on mobile devices. Project page.

Abstract

Despite the increasing popularity of avatar systems such as Snapchat Bitmojis, existing production avatar platforms face several limitations, such as a limited number of predefined assets, tedious customization processes, and inefficient rendering requirements. Addressing these shortcomings, we introduce Snapmoji, an avatar generation system that instantly creates 3D avatars, and enables customization in a process we call dual-stylization. Snapmoji first maps a selfie of a user to a primary avatar (e.g., Bitmoji style) using a new technique we name Gaussian Domain Adaptation (GDA), then applies a secondary style (e.g., skeleton, yarn, toy) to the primary avatar, all while preserving the user's identity. The generated 3D avatars can then be rendered an animated on mobile devices at 30–40 FPS.

1. Introduction

With platforms like Snapchat Bitmojis [5], Apple Memojis [1], and Meta avatars [31], personalized avatars have now become a cornerstone of social media. These platforms allow users to choose features from a range of cartoon-like traits, such as hairstyles, facial features, clothing, and accessories. However, despite their popularity, existing avatar platforms all share a central limitation: traits are limited to a list of predefined assets. Creating new traits requires a team of artists to create new assets from scratch, and this workflow becomes especially unsustainable as avatars are continuously updated. Take Snapchat for example, which hosts hundreds of avatar fashion assets, stickers, and animations. What happens if a user wants to picture their Bitmoji as if they were a superhero like Spiderman? On Snapchat, a typical marketing promotion for a movie or event will include 30 to 150 new clothing assets, requiring a significant amount of work from artists. So is there a way we can automatically stylize a user's avatar without this manual effort? This challenge underscores the need for an ability we call **dual-stylization**: the ability of a platform to not only enerate a single avatar for a user, but also to re-style it across multiple themes without manually building new 3D assets. In this example, we call the user's base Bitmoji the *primary avatar*, and their Bitmoji pictured as Spiderman their *dual-stylized avatar*.

Although recent research like StyleAvatar3D [54], TextToon [48], and DATID-3D [23] propose generative methods to create 3D avatars without requiring new assets, we find that these methods still do not fulfill the requirements that a production system should have. Their generated avatars

[*] Equal contribution

Table 1. Comparison among stylized avatar generation methods.

Method	Selfie Input	Mobile AR	Asset-free	Animatable	Dual Style
StyleAvatar3D [54]			✓		
DATID-3D [23]	✓		✓		
TextToon [48]			✓	✓	
EasyCraft [51]		✓		✓	
SwiftAvatar [50]	✓	✓		✓	
AgileAvatar [40]	✓	✓		✓	
Snapmoji (ours)	✓	✓	✓	✓	✓

are typically expensive to create, cannot be rendered in real time, and are not animatable (see Table 1 for a comparison). For applications like augmented reality (AR), they are unsuitable.

In search for an alternative, we propose *Snapmoji*, a novel framework for generating expressive and animatable avatars, represented in the form of 3D Gaussian Splats [22]. Using data collected from the public APIs of Bitmoji [46], Snapmoji is trained to map selfies of users to Bitmoji-styled avatars, and to support dual stylization: customizing the Bitmoji avatar in a user-specified style. The dual-styles are specified from text prompts, such as of "LEGO" or "Yoda", and the generated avatars preserve both the primary Bitmoji style, and the user's identity.

Our system is designed with three core objectives in mind:

1. **Dual Stylization:** The system should generate avatars in the Bitmoji art style, and a secondary style, such as of LEGO or Yoda, while preserving user identity.
2. **User Convenience:** For ease of use, the system should require only a single image as input and produce the avatar instantly.
3. **Efficiency:** The avatars should enable real-time rendering on mobile devices, supporting applications like AR.

To meet these goals, we introduce a two-stage pipeline. In the first stage, we present Gaussian Domain Adaptation (GDA), a domain translation method that leverages a learned 3D prior to map realistic selfies into the Bitmoji space, followed by a diffusion model that further customizes the style based on user-provided text prompts. In the second stage, the stylized 2D avatar is lifted to a 3D avatar, and can be animated using blendshapes in real time.

Although our system is demonstrated using Bitmojis, the pipeline is general and can be adapted to other avatar ecosystems, potentially enabling a wide range of creative applications across gaming, social media, virtual meetings, and education. In summary, our contributions are as follows:

- We introduce the concept of dual stylization, and propose an efficient system to convert a realistic image into a dual-stylized 3D animatable avatar
- We propose a novel domain adaptation method, Gaussian Domain Adaptation (GDA), to transfer a real image into a predefined avatar style.
- We develop a Javascript framework to animate our generated avatars on mobile phones at 30-40 FPS.

2. Related Work

Commercial Systems for Avatar Creation. Commercial avatar platforms from TikTok, Apple, etc. have begun to introduce automated processes for creating avatars by training classifiers that can predict avatar traits from a user's photograph. The users can then manually adjust the traits to their preference. However, these classifiers require paired data that link real faces to specific avatar traits, which is difficult to acquire at scale. To address this challenge, approaches like AgileAvatar [40] and SwiftAvatar [50] from TikTok; and F2P [42, 43] and EasyCraft [51] from NetEase; develop self-supervised learning techniques to map real photos to avatars. While these methods are efficient, they can only predict traits from existing 3D asset libraries. Our goal however is to generate entirely new avatars without requiring more 3D assets to be manually created.

2D Stylized Avatar Generation Also relevant to avatar creation are techniques in image-to-image translation. Neural networks like StyleGAN [21] have been used to map realistic images of faces to various domains, like of Disney cartoons, paintings, and vintage photos [7, 27, 35]. One reason why StyleGAN is so popular for this face stylization task is that it does not require paired images between the domains. Works such as SwiftAvatar [50] use this property of StyleGAN to map real faces to cartoon avatars.

Along with GANs, diffusion models are another prominent approach for 2D stylization. Unlike GANs which are trained on single-class datasets, diffusion models are typically trained on internet-scale data, giving them more stylistic diversity than their GAN counterparts. To control the generation process, models like Stable Diffusion [38] take a text prompt as input. Further tools, including SDEdit [30], ControlNet [55], and IP Adapter [53], provide additional control by allowing image prompts as input. Because of their complimentary strengths, our method uses both GANs and diffusion models for avatar generation.

3D Avatar and Object Generation. Recent advances in 3D representations like neural radiance fields [32] and Gaussian splats [22] can also be applied to avatar generation. For instance, DATID-3D [23] and StyleAvatar3D [54] employ 3D GANs to generate and stylize 3D face models. More recent developments utilize text-to-image diffusion models for avatar stylization [9, 14, 15, 29, 33, 48], though this process tends to be slow. Different from our task of cartoon avatar generation, much work also also been done in generating photorealistic avatars with Gaussian splats [26, 28, 36, 39] . The major difference from these photorealistic avatar methods and cartoon avatar methods

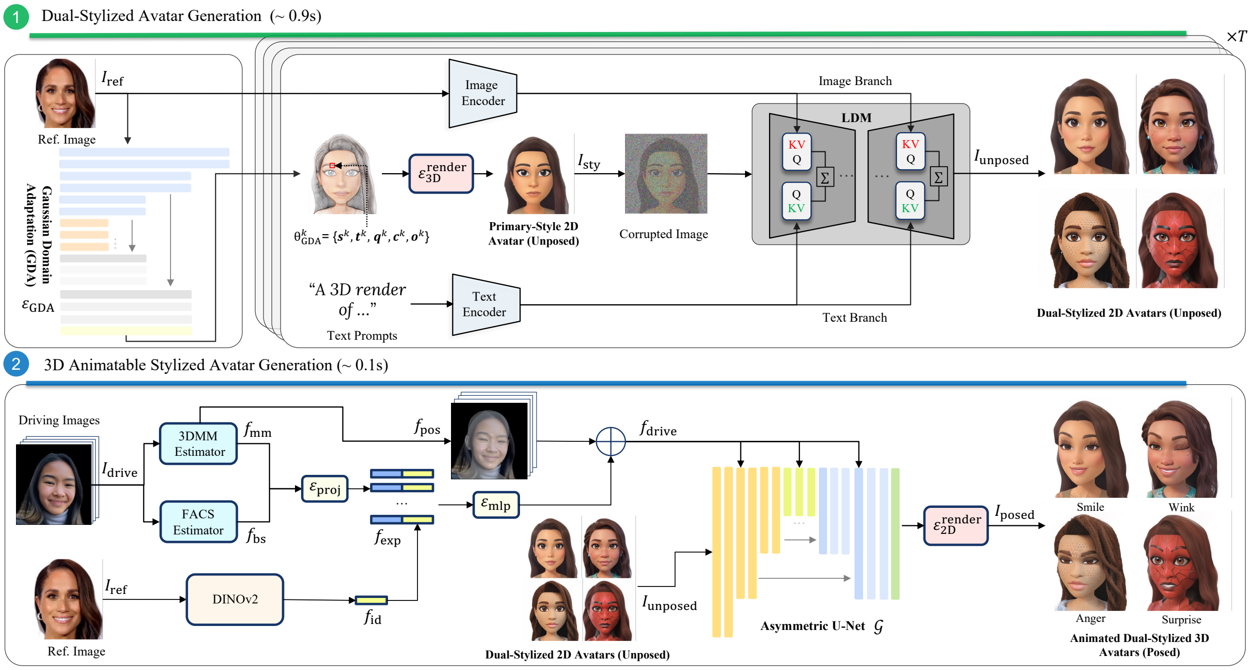
1 Dual-Stylized Avatar Generation (~ 0.9s)
×T
I_{ref}
Ref. Image
Gaussian Domain Adaptation (GDA)
$\mathcal{E}_{\text{GDA}}$
Image Encoder
Image Branch
LDM
KV
Q
Q
KV
Σ
$\mathcal{E}_{3D}^{\text{render}}$
$\theta^k_{\text{GDA}} = \{s^k, t^k, q^k, c^k, o^k\}$
Primary-Style 2D Avatar (Unposed)
I_{sty}
Corrupted Image
I_{unposed}
Dual-Stylized 2D Avatars (Unposed)
"A 3D render of ..."
Text Prompts
Text Encoder
Text Branch
2 3D Animatable Stylized Avatar Generation (~ 0.1s)
Driving Images
I_{drive}
3DMM Estimator
f_{mm}
FACS Estimator
f_{bs}
$\mathcal{E}_{\text{proj}}$
f_{exp}
f_{pos}
$\mathcal{E}_{\text{mlp}}$
f_{drive}
I_{ref}
Ref. Image
DINOv2
f_{id}
I_{unposed}
Dual-Stylized 2D Avatars (Unposed)
Asymmetric U-Net $\mathcal{G}$
$\mathcal{E}_{2D}^{\text{render}}$
I_{posed}
Smile
Wink
Anger
Surprise
Animated Dual-Stylized 3D Avatars (Posed)

Figure 2. **The Snapmoji Inference Pipeline.** The pipeline has two stages. First, the Gaussian Domain Adaptation network $\mathcal{E}_{\text{GDA}}$ converts a facial image into a primary avatar I_{sty}. This avatar undergoes further personalization using a text-guided diffusion process with T steps for additional stylization. Second, expression codes extracted via an 3DMM and FACS are combined with identity features f_{id} from a reference image I_{ref} and positional maps f_{pos} from a driving image I_{drive}. The unposed dual-stylized avatar I_{unposed} is then processed by an asymmetric UNet $\mathcal{G}(\cdot)$, conditioned on the driving codes f_{drive} through cross-attention, to generate animated, dual-stylized 3D avatars.

is that the former fits Gaussians on the surfaces of 3D Morphable Models (3DMMs) like FLAME [25]. Because 3DMMs only model realistic human geometry, this strategy cannot be applied to cartoon avatars, which often have large geometric differences like big eyes or big heads.

In addition to 3D avatars, our method is also inspired by work general 3D object generation. Notably, LRM [17] and LGM [49] are trained on Objaverse to map 2D images to 3D objects in a single network evaluation. As we show in Section 3.2, this data prior from Objaverse can surprisingly be repurposed for image-to-avatar generation.

3. Method

Our method begins with creating a synthetic dataset of real face images and their 2D primary avatars (Sec. 3.1), which is then used for GDA and dual stylization (Sec. 3.2). The dual-stylized avatars are then lifted to 3D and animated (Sec. 3.3). Training details are provided in Sec. 3.4.

3.1. Datasets

Training our image-to-avatar GDA model requires paired datasets of real faces and primary avatars, which are not available at scale. To overcome this, we use GAN inversion, similar to prior work in unsupervised domain adaptation [50], to create synthetic paired data. By aligning the latent spaces of a source GAN and a fine-tuned target GAN [7, 50, 52], we generate corresponding pairs of realistic and Bitmoji faces. Specifically, Bitmoji images are inverted into the target GAN's latent space to obtain latent codes, which are then applied to the source GAN to produce realistic counterparts:

$$w := \text{argmin}_{w \in \mathcal{W}} \|G_{\text{tgt}}(w) - I_{\text{tgt}}\|, I_{\text{src}} = G_{\text{src}}(w). \quad (1)$$

Using this method, we generated 13,000 synthetic image pairs from Bitmoji avatars, forming the basis for GDA training. Examples are shown in the supplementary material.

3.2. Image to 2D Avatar Generation

Gaussian Domain Adaptation $\mathcal{E}_{\text{GDA}}(\cdot)$. Our first step is to map real photos to their corresponding primary avatars (*i.e*., Bitmoji avatars). Surprisingly, we find that features learned by Large Multi-view Gaussian models (LGMs) [49] can be readily adapted for style transfer. We believe this is due to their ability to hold internet-scale information from multi-view training datasets such as Objaverse [10]. We repurpose LGM for this style transfer task, and call this technique *Gaussian Domain Adaptation*. In the supplementary material, we perform an ablation study demonstrating that GDA training struggles without Objaverse pre-training.

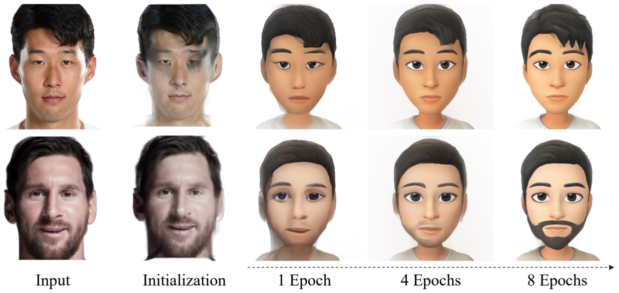

Figure 3. **Gaussian Domain Adaptation.** We show the outputs of the GDA network over several training epochs to visualize the domain shifts from natural images to cartoon avatars.

To adapt LGM for GDA, we fine tune the U-Net to map realistic faces to primary avatars using the data generated in Section 3.1. The training process is shown in Figure 3, where the realistic face domain evolves into the Bitmoji domain. At inference time, we pass a user's reference image $I_{\text{ref}} \in \mathbb{R}^{3\times512\times512}$ through the U-Net to map to the pixel-aligned Gaussian parameters of scaling $\boldsymbol{s}$, position $\boldsymbol{t}$, color $\boldsymbol{c}$, opacity $\boldsymbol{o}$, and orientation $\boldsymbol{q}$:

$$\theta_{\text{GDA}} = \left\{\boldsymbol{s}^k, \boldsymbol{t}^k, \boldsymbol{q}^k, \boldsymbol{c}^k, \boldsymbol{o}^k\right\}_{k=1}^{M} = \mathcal{E}_{\text{GDA}}(I_{\text{ref}}; \Phi_{\text{GDA}}). \quad (2)$$

M is the number of Gaussians and Φ_{GDA} is the learnable parameters. The 3D Gaussians are then rendered in the frontal view $I_{\text{sty}} = \mathcal{E}_{\text{3D}}^{\text{render}}(\theta_{\text{GDA}})$. This process transforms real face photos into the primary 2D avatar domain while preserving identity-related features.

Dual-Stylization. After GDA, we employ a Stable Diffusion [38] pipeline for dual stylization. The diffusion model takes the GDA output image I_{sty}, a text prompt, and the original user photo as input, and outputs a dual-stylized avatar. First, to preserve the coarse structure of the primary avatar, we use SDEdit [30] to start denoising from a noised GDA output. To further preserve the avatar's primary style, we feed the GDA image's Canny edges to ControlNet [55]. To maintain identity preservation, we input the original user photo to IP Adapter [53] to condition the generation on facial similarity embeddings. Using the DDIM scheduler [47], the entire process only uses $T = 10$ denoising steps, taking less than one second.

3.3. 2D to 3D Avatar Generation and Animation

Expression Encoder. Current avatar animation techniques using Gaussian Splats often solely depend on 3D Morphable Models (3DMM) [8, 19, 36, 41, 48], which limits generalization beyond realistic faces, especially for cartoon avatars. To overcome these constraints, we condition the 3D generation network on not only 3DMM features, but also blendshape weights derived from the Facial Action Coding System (FACS) [13]. These weights are widely used in cartoon animation to control facial features like eye position and mouth shape, and can generalize beyond realistic faces. As depicted in Fig. 2, for generating expressive avatars, we extract expression codes $f_{\text{mm}} \in \mathbb{R}^{100}$ from the driving image using a 3DMM estimator. These codes are concatenated with the blendshape vector $f_{\text{bs}} \in \mathbb{R}^{16}$, producing a comprehensive expression feature. A learnable projection layer $\mathcal{E}_{\text{proj}}$ then projects this combined feature into a 16-dimensional expression vector $f_{\text{exp}} = \mathcal{E}_{\text{proj}}([f_{\text{bs}}; f_{\text{mm}}])$, where $[\cdot]$ indicates feature concatenation. To integrate expressiveness with identity, the driving signal is formulated as:

$$f_{\text{drive}} = (\mathcal{E}_{\text{mlp}}([f_{\text{exp}}, f_{\text{id}}]), f_{\text{pos}}). \quad (3)$$

Here, f_{id} is the global identity feature extracted from a reference image I_r via a frozen DINOv2 backbone [34], and f_{pos} denotes the position map from 3DMM vertices.

3D Generation Network $\mathcal{G}(\cdot)$. Given the generated unposed avatars I_{unposed} and driving features f_{drive} from the expression encoder, we employ an asymmetric U-Net architecture akin to Large Multi-view Gaussian Models [49] and incorporate cross-attention layers to merge the driving features seamlessly:

$$I_{\text{posed}} = \mathcal{E}_{\text{2D}}^{\text{render}}(\mathcal{G}(I_{\text{unposed}}, f_{\text{drive}}; \Phi_g)), \quad (4)$$

where Φ_g is the learnable parameters of $\mathcal{G}(\cdot)$ and $\mathcal{E}_{\text{2D}}^{\text{render}}$ is a 2DGS renderer [20]. $\mathcal{G}(\cdot)$ consists of an encoder with five down-sampling blocks, a middle block, and a decoder with three up-sampling blocks. Cross-attention modules are strategically placed in the deeper layers of the network: the last two down-sampling blocks, the middle block, and the first two up-sampling blocks.

Mobile AR Application. Our framework is designed to facilitate real-time animation on mobile devices. Offline, we use the 3D Generation Network $\mathcal{G}(\cdot)$ from the pipeline shown in Fig. 2 to initially generate a base set of Gaussians for the avatar in a rest pose θ_{rest}, along with specific Gaussian sets corresponding to each component of the expression features f_{drive}. On the mobile device, we use a face tracker, such as *Mediapipe*'s BlazeFace tracker [4], to generate a list of blendshape weights $f_{\text{bs}} \in \mathbb{R}^{16}$. We leverage these weights to animate the avatar through linear interpolation between the parameters of each feature component:

$$\theta_{\text{mobile}} = \theta_{\text{rest}} + \sum_{i=1}^{K} f_{\text{drive}}^{i}(\theta_i - \theta_{\text{rest}}) \quad (5)$$

K represents the number of driving features, and can be tuned to balance expression detail and speed. For compatibility with Mediapipe, we choose $K = 16$. The final rendering of the Gaussians θ_{mobile} takes place in WebGL, offering efficient rendering while retaining high visual fidelity. To demonstrate this capability, we developed a JavaScript application that allows users to control their avatars directly in their browsers.

3.4. Training and Losses

Image to Avatar Generation. Our training process uses GDA to map the reference identity I_{ref} to Gaussian parameters θ_{GDA}, which are then used to render the unposed primary avatar I_{unposed} from the frontal view via a 3DGS renderer $\mathcal{E}_{3\text{D}}^{\text{render}}$. The rendered image is supervised using a combination of Mean Squared Error (MSE) and perceptual LPIPS [56] losses:

$$\mathcal{L}_{\text{GDA}} = \mathcal{L}_{\text{MSE}}(I_{\text{ref}}, I_{\text{sty}}) + \mathcal{L}_{\text{LPIPS}}(I_{\text{ref}}, I_{\text{sty}}). \quad (6)$$

Despite potential noise introduced by low-quality GAN inversion, the pre-training on 3D datasets including Objaverse equips our network with strong generalization capabilities, enabling effective real-to-avatar domain adaptation. As shown in Fig. 3, GDA efficiently transforms realistic faces into a primary style while preserving the subjects' identity and enhancing features, *e.g.*, eye size.

3D Animatable Stylized Avatar Generation. To improve the surface geometry of avatars, our model incorporates normal consistency and depth distortion losses. The normal consistency loss $\mathcal{L}_{\text{normal}}$ aligns the normals of 2D Gaussians [20] with surface normals determined through finite differences from rendered depths, thereby reducing noise. Meanwhile, the depth distortion loss $\mathcal{L}_{\text{dist}}$, implemented following [2, 3], encourages Gaussians to cluster closely along camera rays, effectively enhancing surface representation. This optimization allows our network to output avatars with detailed geometry, suitable for applications such as animation and relighting. The total loss function for the 3D generation network is defined as:

$$\mathcal{L}_{\text{3DGen}} = \mathcal{L}_{\text{render}} + \lambda_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}} + \lambda_{\text{n}}\mathcal{L}_{\text{normal}} + \lambda_{\text{d}}\mathcal{L}_{\text{dist}}, \quad (7)$$

where $\mathcal{L}_{\text{render}}$ combines RGB and alpha mask losses:

$$\mathcal{L}_{\text{render}} = \|I_{\text{posed}} - I_{\text{posed}}^{\text{gt}}\|_2^2 + \|\alpha^{\text{pred}} - \mathbf{M}^{\text{gt}}\|_2^2. \quad (8)$$

$\mathcal{L}_{\text{normal}}$ aligns predicted normals with surface normals:

$$\mathcal{L}_{\text{normal}} = 1 - (\mathbf{n}^{\text{pred}} \cdot \mathbf{n}^{\text{surf}}). \quad (9)$$

Here, $I^{\text{posed}}, I_{\text{gt}}^{\text{posed}}$ are the predicted and ground truth images; α^{pred} and $\mathbf{M}^{\text{gt}}$ are the predicted alpha mask and its ground truth counterpart; $\mathbf{n}^{\text{pred}}, \mathbf{n}^{\text{surf}}$ are predicted and surface normal vectors. $\lambda_{\text{lpips}}, \lambda_{\text{n}}, \lambda_{\text{d}}$ are weights for $\mathcal{L}_{\text{lpips}}$, $\mathcal{L}_{\text{normal}}$ and $\mathcal{L}_{\text{dist}}$, respectively. The normal and distortion losses commence after 20% of training to first establish basic appearance convergence.

Table 2. **Image to 2D Avatar Generation.** We compare different methods of generating 2D stylized avatars. Our GDA significantly outperforms GAN inversion and diffusion in terms of image quality (FID, KID), identity preservation (ID), and execution speed.

	FID ↓	KID ↓	ID ↑	Speed ↓
GAN Inversion	93.73	0.0603	0.16	98.14s
Diffusion	93.63	0.0457	0.19	3.54s
GDA (Ours)	**72.94**	**0.0346**	**0.25**	**0.080s**

4. Experiments

As no prior work follows the exact design goals as ours, we evaluate each component of the Snapmoji system individually, similar to an ablation study. Section 4.1 studies the image to 2D avatar generation step, Section 4.2 studies the 2D-to-3D avatar generation step, and Section 4.3 studies the 3D animation step. Evaluation is performed on a dataset of Bitmojis which we plan on open sourcing upon publication.

4.1. Image to 2D Avatar Generation

Baselines. We first evaluate GDA for image-to-avatar generation and compare with GAN inversion and diffusion-based methods, both fine-tuned on our Bitmoji dataset. For GAN inversion, invert real faces into the latent space of a fine-tuned SemanticStyleGAN [44] model. The diffusion baseline is a Stable Diffusion 1.5 model [38] fine-tuned with a Bitmoji LoRA [18] and BLIP-2 [24] captions. At inference time, a real face is input into an IP Adapter Plus Face [53] model for identity conditioning.

Evaluation. We conducted an evaluation using 100 randomly selected faces from the FFHQ dataset, assessing each method on visual quality, identity retention, and speed. Visual quality was measured through FID and KID scores between the transformed images to the Bitmoji dataset. Identity retention was evaluated using ArcFace [11], and speed was benchmarked on a Nvidia L4 GPU. As reported in Table 2, GDA outperforms the baselines across all metrics, achieving FID scores more than 20 points lower than those of GAN inversion and diffusion. Figure 4 visually highlights these quality differences: GAN inversion GAN inversion struggles to generate avatars with diverse hair styles, eye colors, and clothing. The diffusion approach fails to maintain a consistent style and often incorrectly introduces features like glasses, undermining both style and identity preservation. In contrast, GDA produces avatars that both have a consistent Bitmoji style and retain key identity features such as eye color, sunglasses, and hairstyles. Finally, since GDA only requires one forward pass through a UNet, it is two and four orders of magnitude faster than diffusion and GAN inversion respectively, translating images in less than 0.1 seconds. Surprisingly, even though GDA is trained on data generated from GAN inversion, because of GDA's Objaverse [10] prior, the resulting images are more detailed than from GAN inversion alone.

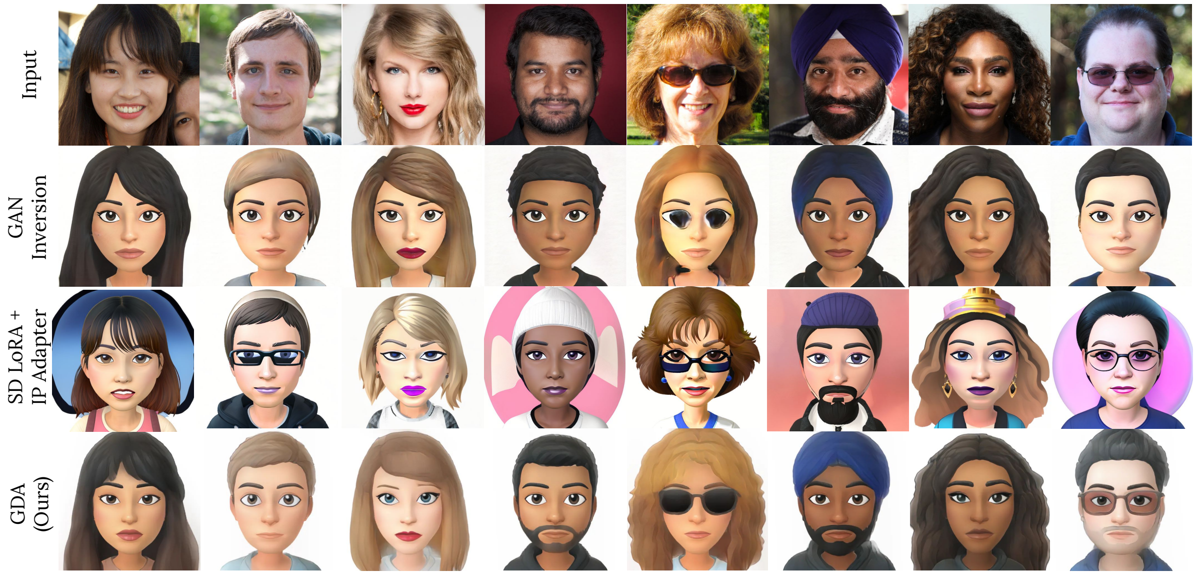

Figure 4. **Image to 2D Avatar Generation.** Showcased are photos from eight individuals transformed into the primary Bitmoji domain using various methods. GAN inversion produces overly generic avatars, struggling with unique features such as beards, glasses, and headwear. Diffusion-based models inaccurately add features, making them inconsistent for targeted styles. In contrast, our GDA method excels in creating high-quality avatars, effectively retaining the original identity features.

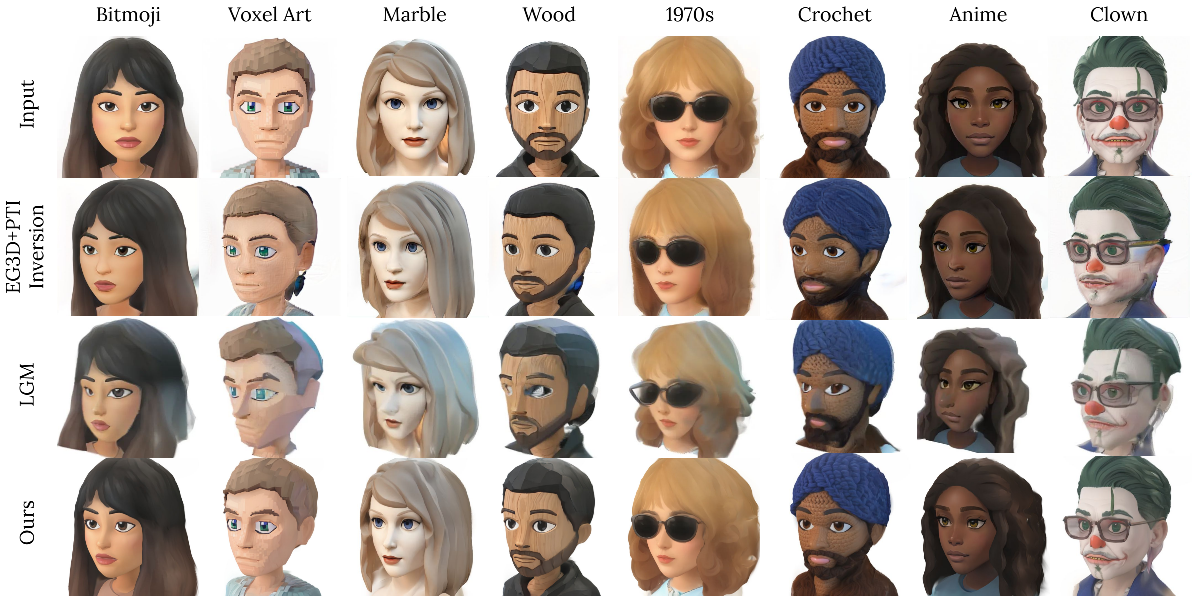

Figure 5. **2D to 3D Avatar Generation.** We demonstrate the process of converting dual-stylized avatar, derived from the primary avatars in Fig. 4, into 3D avatars. PTI inversion with EG3D [6, 37] struggles to accurately reproduce 3D geometry, while LGM [49] produces artifacts in both geometry and texture. Despite being trained exclusively on the Bitmoji style, our method successfully generates high-quality 3D avatars in previously unseen styles.

4.2. 3D Avatar Generation

2D to 3D Avatar Generation. After dual stylization, our method lifts the 2D avatar image to 3D. We compare our proposed technique against two other single-image 3D reconstruction techniques: a EG3D [6] model fine-tuned on Bitmojis, and LGM [49]. The EG3D baseline maps 2D images to radiance fields via pivotal tuning inversion [37], while LGM uses MVDream [45] to transform a single image into multi-view images, then maps that set of images to

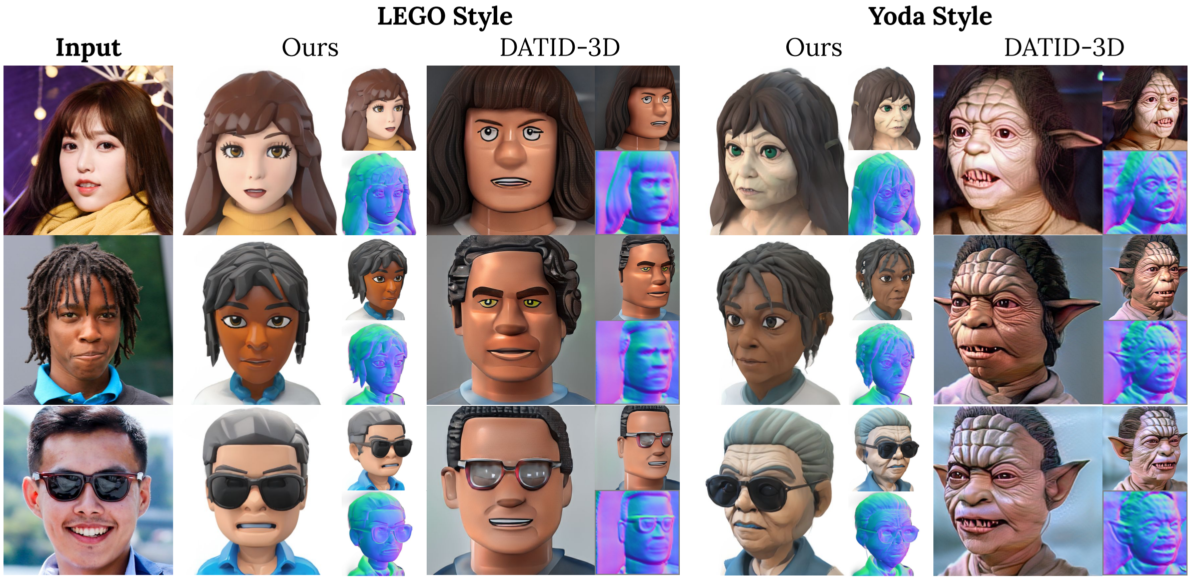

Figure 6. **Single Portrait to 3D Generation.** We compare Snapmoji with DATID-3D [23] in the context of dual stylization. For each method and style, we render outputs from two viewpoints alongside a normal map. DATID-3D exhibits typical GAN-related issues, such as poor identity preservation and limited stylistic diversity, resulting in similar outputs across different identities. Conversely, Snapmoji effectively maintains identity and produces distinct styles, showcasing superior image quality and sharper geometry.

Table 3. **2D to 3D Avatar Generation.** Our approach outperforms EG3D [6] and LGM [49] on all metrics, providing superior texture and geometry accuracy with faster processing.

	PSNR ↑	SSIM ↑	LPIPS ↓	Speed ↓
EG3D [6]	10.92	0.68	0.50	95.1s
LGM [49]	12.16	0.69	0.53	2.82s
Ours	**18.73**	**0.81**	**0.24**	**0.091s**

3D Gaussians. We assessed each method using 100 random 3D Bitmojis, rendered from ten views distributed spherically around the head. As shown in Table 3, our method surpasses all baselines, demonstrating superior capability in accurately converting 2D images to 3D, while being significantly faster, needing only a single U-Net pass.

Figure 5 provides visual comparisons on dual-stylized avatars. The top row features eight dual-stylized avatars generated from the identities in Fig. 4. EG3D struggles to generate high-fidelity geometry. Similarly, due to the diffusion process in MVDream, LGM is a slow, iterative process, and often produces incorrect 3D head geometries. In contrast, our method successfully creates high-quality textures and geometry, even for out-of-distribution accessories like turbans and sunglasses.

Single Portrait to 3D Generation. Next, we evaluate Snapmoji's full image-to-3D-avatar generation ability against DATID-3D [23]. Figure 6 shows that DATID-3D struggles to maintain the original identity in avatars. Snapmoji, however, achieves a robust balance of identity preservation and style versatility. Our approach produces sharp images with detailed geometries, and only creates each avatar in just 0.9 seconds, a significant improvement over DATID-3D's 90-second processing time.

Table 4. **User Preferences for Dual-Stylized Avatars.** Our method outperforms DATID-3D [23] in user preferences. Users believed our avatars better preserved idenity, had more diversity, and would be a better fit the aesthetics of a Bitmoji video game.

User Study (N=27)	DATID-3D	Ours
Better identity preservation?	8%	92%
Better variation in style and identity?	4%	96%
Better aesthetics in a Bitmoji video game?	4%	96%

Dual Stylization User Study. Recall that one of our major goals is to support dual stylization. The avatars should resemble the user, fit the art style of the Bitmoji world, and enable diversity in customization. We compare Snapmoji to DATID-3D with a user study on the 12 avatars in Figure 6. Across 27 participants, 92% said that our avatars better resemble the input users. The DATID-3D avatars are not diverse enough to preserve the appearance and skin tone of users, making them unacceptable for a production system. Regarding style, 96% said our avatars had more variation in style and identity, and 96% said they would prefer our avatar aesthetics in a video game set in the Bitmoji world. While the avatars generated from DATID-3D may look more like LEGO or Yoda at first glance, they are limited in diversity, and do not fit the primary Bitmoji style.

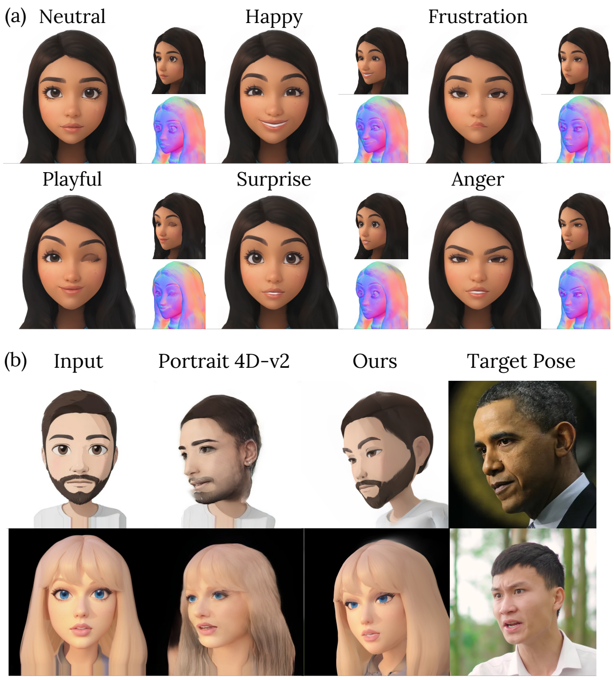

Figure 7. **Avatar Animation.** (a) An Snapmoji showcasing various emotions controlled by blendshape weights. (b) Snapmoji effectively transfers expressions from driving images, outperforming Portrait4D-v2 [12] in accuracy and visual appeal.

4.3. 3D Avatar Animation

Expression Animation. Snapmoji enables avatars to express a wide range of emotions, such as neutrality, happiness, frustration, playfulness, anger, and surprise, by using blendshape weights, as shown in Fig. 7(a). Additionally, Snapmoji can perform expression transfer from driving images, producing 3D-consistent and visually appealing avatars. Fig. 7(b) shows this capability, where Snapmoji outperforms Portrait4D-v2 [12] by generating avatars with more accurate expressions derived from the target image. 3DMMs, used by Portrait3D-v2 [11] and many other works [8, 16, 48], do not generalize to the geometry of cartoon heads.

Mobile AR. We provide a web-based AR app to efficiently render and animate avatars using a face tracker. As illustrated in Fig. 8, an avatar animated using a user's facial expressions can be rendered at 30–40 FPS on an iPhone 13 Pro. These avatars occupy only 3 MB of disk space, enabling the creation of dynamic filters and engaging AR effects directly within a mobile web browser. To highlight the advantages of our animation technique, we compare Snapmoji against TextToon [48]. Like many other avatar generation methods [8, 19, 36], TextToon requires a neural network to predict 3DMM features, limiting it to 15–18 FPS on an M1 MacBook. In contrast, our method consistently runs at 90-100 FPS on a laptop. Moreover, TextToon's dependence on 3DMMs limits its practicality on phones, whereas our cross-platform solution can still run at 30 FPS. Table 5 offers a detailed feature comparison. Other work like LAM [16] propose alternative ways to render avatars on mobile devices, but we do not compare against them because they are incapable of AR puppeting. Unlike our model and TextToon, they do not attempt to integrate their system with a face tracker at inference time.

Table 5. **Mobile AR Application Comparison.** We compare various features of our mobile AR application and TextToon [48].

Method	Frame Rate (FPS)		Cross-Platform	Driving Signal
	M1 MacBook	iPhone 13 Pro		
TextToon [48]	15–18	N/A	×	3DMM
Snapmoji (Ours)	90-100	30–40	✓	3DMM + Blendshapes

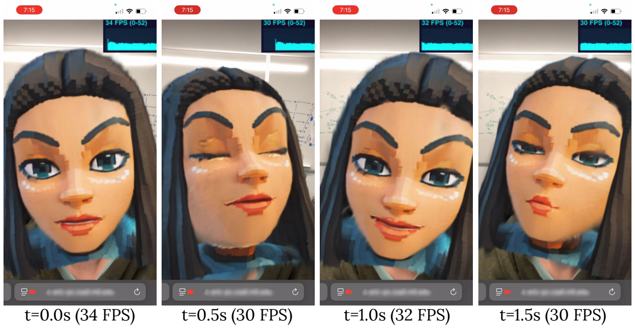

Figure 8. **Mobile AR.** Snapmoji enables a user to puppet their avatar in augmented reality. Our method runs entirely on a web client, at 30-40 FPS on a phone.

5. Conclusion

Although avatar generation is now one of the most popular research areas in computer vision, the gap between the existing research literature and production is still large. We aim to bridge this gap by introducing Snapmoji, a system for generating animatable, dual-stylized avatars from selfies almost instantly. Leveraging Gaussian Domain Adaptation, Snapmoji first converts selfies into primary stylized avatars, then applies a diffusion process for a secondary style while preserving identity integrity. The system achieves selfie-to-avatar conversion in just 0.9 seconds, and enabling real-time interactions at 30–40 FPS.

Limitations and Future Work. Snapmoji requires a large 3D avatar dataset for training. We perform all experiments on Bitmoji avatars, but future work could also apply these methods to other avatar platforms. We recognize that there is a lack of avatar data available for research, so we intend to publicly release our data as well. Future improvements could include improving the face tracker, which limits the quality of the animations. Mediapipe can produce noisy blendshapes, leading to jittery results.

References

[1] Apple. Use memoji on your iphone or ipad pro. https://support.apple.com/en-us/111115, 2018. Accessed: 2024-11-03. 1

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5855–5864, 2021. 5

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 5

[4] Valentin Bazarevsky, Yury Kartynnik, Andrey Vakunov, Karthik Raveendran, and Matthias Grundmann. Blazeface: Sub-millisecond neural face detection on mobile gpus. *ArXiv*, abs/1907.05047, 2019. 4

[5] Bitstrips. Bitmoji. https://www.bitmoji.com/, 2007. Accessed: 2024-11-03. 1

[6] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022. 6, 7

[7] Eric Chen, Jin Sun, Apoorv Khandelwal, Dani Lischinski, Noah Snavely, and Hadar Averbuch-Elor. What's in a decade? transforming faces through time. *Computer Graphics Forum*, 42, 2022. 2, 3

[8] Xuangeng Chu and Tatsuya Harada. Generalizable and animatable gaussian head avatar. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 4, 8

[9] Quan Dao, Khanh Doan, Di Liu, Trung Le, and Dimitris Metaxas. Improved training technique for latent consistency models. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. 2

[10] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13142–13153, 2022. 3, 5

[11] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 5, 8

[12] Yu Deng, Duomin Wang, and Baoyuan Wang. Portrait4d-v2: Pseudo multi-view data creates better 4d head synthesizer. In *European Conference on Computer Vision*, pages 316–333. Springer, 2024. 8

[13] Paul Ekman and Wallace V. Friesen. Facial action coding system: a technique for the measurement of facial movement. 1978. 4

[14] Ligong Han, Song Wen, Qi Chen, Zhixing Zhang, Kunpeng Song, Mengwei Ren, Ruijiang Gao, Anastasis Stathopoulos, Xiaoxiao He, Yuxiao Chen, et al. Proxedit: Improving tuning-free real image editing with proximal guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4291–4301, 2024. 2

[15] Xiaoxiao He, Ligong Han, Quan Dao, Song Wen, Minhao Bai, Di Liu, Han Zhang, Martin Renqiang Min, Felix Juefei-Xu, Chaowei Tan, et al. Dice: Discrete inversion enabling controllable editing for multinomial diffusion and masked generative models. 2025. 2

[16] Yisheng He, Xiaodong Gu, Xiaodan Ye, Chao Xu, Zhengyi Zhao, Yuan Dong, Weihao Yuan, Zilong Dong, and Liefeng Bo. Lam: Large avatar model for one-shot animatable gaussian head. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–13, 2025. 8

[17] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *ArXiv*, abs/2311.04400, 2023. 3

[18] J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021. 5

[19] Liangxiao Hu, Hongwen Zhang, Yuxiang Zhang, Boyao Zhou, Boning Liu, Shengping Zhang, and Liqiang Nie. Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 4, 8

[20] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. 4, 5

[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8107–8116, 2019. 2

[22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42:1 – 14, 2023. 2

[23] Gwanghyun Kim and Se Young Chun. Datid-3d: Diversity-preserved domain adaptation using text-to-image diffusion for 3d generative model. In *CVPR*, 2023. 1, 2, 7

[24] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, 2023. 5

[25] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017. 3

[26] Di Liu, Teng Deng, Giljoo Nam, Yu Rong, Stanislav Pidhorskyi, Junxuan Li, Jason Saragih, Dimitris N. Metaxas,

and Chen Cao. Lucas: Layered universal codec avatars, 2025. 2

[27] Xuan Luo, Xuaner Zhang, Paul Yoo, Ricardo Martin-Brualla, Jason Lawrence, and Steven M. Seitz. Time-travel rephotography. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2021)*, 40(6), 2021. 2

[28] Shengjie Ma, Yanlin Weng, Tianjia Shao, and Kun Zhou. 3d gaussian blendshapes for head avatar animation. In *ACM SIGGRAPH Conference Proceedings, Denver, CO, United States, July 28 - August 1, 2024*, 2024. 2

[29] Yifang Men, Hanxi Liu, Yuan Yao, Miaomiao Cui, Xuansong Xie, and Zhouhui Lian. 3dtoonify: Creating your high-fidelity 3d stylized avatar easily from 2d portrait images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10127–10137, 2024. 2

[30] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2021. 2, 4

[31] Meta. Express yourself with Meta avatars. https://www.meta.com/avatars/, 2024. Accessed: 2024-11-03. 1

[32] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[33] Thu Nguyen-Phuoc, Gabriel Schwartz, Yuting Ye, Stephen Lombardi, and Lei Xiao. Alteredavatar: Stylizing dynamic 3d avatars with fast style adaptation. *ArXiv*, abs/2305.19245, 2023. 2

[34] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 4

[35] Justin N. M. Pinkney and Doron Adler. Resolution dependent gan interpolation for controllable image synthesis between domains. *ArXiv*, abs/2010.05334, 2020. 2

[36] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. *arXiv preprint arXiv:2312.02069*, 2023. 2, 4, 8

[37] Daniel Roich, Ron Mokady, Amit H. Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on Graphics (TOG)*, 42:1 – 13, 2021. 6

[38] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2021. 2, 4, 5

[39] Shunsuke Saito, Gabriel Schwartz, Tomas Simon, Junxuan Li, and Giljoo Nam. Relightable gaussian codec avatars. In *CVPR*, 2024. 2

[40] Shen Sang, Tiancheng Zhi, Guoxian Song, Minghao Liu, Chun-Pong Lai, Jing Liu, Xiang Wen, James Davis, and Linjie Luo. Agileavatar: Stylized 3d avatar creation via cascaded domain bridging. *SIGGRAPH Asia 2022 Conference Papers*, 2022. 2

[41] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. SplattingAvatar: Realistic Real-Time Human Avatars with Mesh-Embedded Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 4

[42] Tianyang Shi, Yi Yuan, Changjie Fan, Zhengxia Zou, Zhen Xia Shi, and Yong Liu. Face-to-parameter translation for game character auto-creation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 161–170, 2019. 2

[43] Tianyang Shi, Yi Yuan, Changjie Fan, Zhengxia Zou, Zhenwei Shi, and Yong Liu. Fast and robust face-to-parameter translation for game character auto-creation. *ArXiv*, abs/2008.07132, 2020. 2

[44] Yichun Shi, Xiao Yang, Yangyue Wan, and Xiaohui Shen. Semanticstylegan: Learning compositional generative priors for controllable image synthesis and editing. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11244–11254, 2021. 5

[45] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv:2308.16512*, 2023. 6

[46] Snap Inc. Bitmoji 3d avatar platform solutions. https://developers.snap.com/lens-studio/platform-solutions/bitmoji-avatar/bitmoji-3d, 2024. 2

[47] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ArXiv*, abs/2010.02502, 2020. 4

[48] Luchuan Song, Lele Chen, Celong Liu, Pinxin Liu, and Chenliang Xu. Texttoon: Real-time text toonify head avatar from single video. 2024. 1, 2, 4, 8

[49] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, 2024. 3, 4, 6, 7

[50] Shizun Wang, Weihong Zeng, Xu Wang, Han Yang, Li Chen, Chuang Zhang, Ming Wu, Yi Yuan, Yunzhao Zeng, and Minghang Zheng. Swiftavatar: Efficient auto-creation of parameterized stylized character on arbitrary avatar engines. In *AAAI Conference on Artificial Intelligence*, 2023. 2, 3

[51] Suzhen Wang, Weijie Chen, Wei Zhang, Minda Zhao, Lincheng Li, Rongsheng Zhang, Zhipeng Hu, and Xin Yu. Easycraft: A robust and efficient framework for automatic avatar crafting, 2025. 2

[52] Zongze Wu, Yotam Nitzan, Eli Shechtman, and Dani Lischinski. Stylealign: Analysis and applications of aligned stylegan models. *ArXiv*, abs/2110.11323, 2021. 3

[53] Hu Ye, Jun Zhang, Siyi Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *ArXiv*, abs/2308.06721, 2023. 2, 4, 5

[54] Chi Zhang, Yiwen Chen, Yijun Fu, Zhenglin Zhou, Gang YU, Billzb Wang, Bin Fu, Tao Chen, Guosheng Lin, and

Chunhua Shen. Styleavatar3d: Leveraging image-text diffusion models for high-fidelity 3d avatar generation, 2023. 1, 2

[55] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3813–3824, 2023. 2, 4

[56] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 5