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

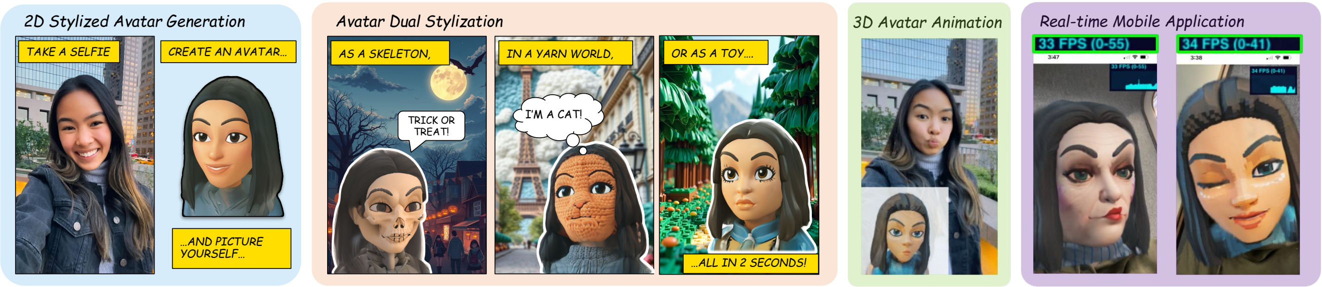

Figure 1. We introduce **Snapmoji**, a system that can instantly generate animatable dual-stylized avatars. Our dual stylization process reimagines avatars in various artistic styles, enabling users to visualize themselves in diverse scenarios and create personalized stories. Our approach also enables 3D stylized gaussian avatars generation and expression animation. Snapmoji accomplishes the selfie-to-avatar conversion in just 0.9 seconds, and offers real-time functionality for mobile applications. Project page.

Abstract

*The increasing popularity of personalized avatar systems, such as Snapchat Bitmojis and Apple Memojis, highlights the growing demand for digital self-representation. Despite their widespread use, existing avatar platforms face significant limitations, including restricted expressivity due to predefined assets, tedious customization processes, or inefficient rendering requirements. Addressing these shortcomings, we introduce Snapmoji, an avatar generation system that instantly creates animatable, dual-stylized avatars from a selfie. We propose Gaussian Domain Adaptation (GDA), which is pre-trained on large-scale Gaussian models using 3D data from sources like Objaverse and fine-tuned with 2D style transfer tasks, endowing it with a rich 3D prior. This enables Snapmoji to transform a selfie into a primary stylized avatar (*e.g.*, Bitmoji style) and apply a secondary style (*e.g.*, Plastic Toy or Alien), all while preserving the user's identity and the primary style's integrity. Our system is capable of producing 3D Gaussian avatars that support dynamic animation, including accurate facial expression transfer. Designed for efficiency, Snapmoji achieves selfie-to-avatar conversion in a mere 0.9 seconds and supports real-time interactions on mobile devices at 30–40 FPS. Extensive testing confirms that Snapmoji outperforms existing methods in versatility and speed, making it a convenient tool for automatic avatar creation in various styles.*

1. Introduction

Personalized cartoon avatars such as *Snapchat Bitmojis* [5], *Apple Memojis* [1], and *Meta Avatars* [35] have become popular digital self-representations. The broader digital avatar market, encompassing both stylized and photorealistic avatars, was valued at over $18 billion in 2023 [41]. Current stylized avatar platforms, although offering some level of customization, are often restricted by predefined traits, which makes it difficult to adapt avatars to varied styles without developing new 3D assets [29, 46, 56, 57]. Moreover, navigating extensive trait lists can be tedious, and efficiency demands frequently lead to compromises in texture detail and polygon count. Asset-free methods, such as StyleAvatar3D [61], TextToon [54] and DATID-3D [23], lack support for real-time operation or animatability for mobile augmented reality (AR). Table 1 provides a comparative overview of existing stylized avatar generation methods. While photorealistic avatar techniques [30, 32, 40, 44] excel in creating realistic representations and animation, they fall short in adapting to the uniquely stylized geometries of cartoon avatars.

In pursuit of a more expressive stylized avatar creation platform, we introduce *Snapmoji*, a system to generate 3D avatars, represented by 3D Gaussian Splats [22], in only 0.9 seconds. Snapmoji is built upon the Bitmoji platform, leveraging its public API and robust developer support [52]. Unlike traditional avatars limited to predefined assets, our

[*]Equal contribution

Method	Selfie Input	Mobile AR	Asset-free	Animatable	Dual Style
StyleAvatar3D [61]	×	×	✓	×	×
DATID-3D [23]	✓	×	✓	×	×
TextToon [54]	×	×	✓	✓	×
EasyCraft [57]	×	✓	×	✓	×
SwiftAvatar [56]	✓	✓	×	✓	×
AgileAvatar [46]	✓	✓	×	✓	×
Snapmoji (ours)	✓	✓	✓	✓	✓

Table 1. Feature comparison among various stylized avatar generation methods.

solution allows for stylization through text prompts, offering greater flexibility and creativity. 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 Plastic Toys or Aliens, while preserving user identity; 2) *User Convenience:* For ease of use, the system should require only a single image input and produce results instantly; 3) *Efficiency:* The avatars should be optimized for real-time rendering on mobile devices, supporting applications like AR, with minimal compute requirements.

Following these design objectives, we propose a two-stage pipeline for Snapmoji. First, Gaussian Domain Adaptation (GDA) transforms realistic selfies into 2D Bitmoji-style images, and then a diffusion-based model further stylizes these images based on user-specified text prompts. Second, the image is lifted to an animatable 3D Gaussian avatar that faithfully captures the user's identity and chosen styles. To facilitate AR applications, these avatars can be animated in real-time using facial parameter estimators. Although showcased with Bitmojis, our approach is applicable to other avatar platforms as well. In summary, our contributions include:

- Introducing an advanced avatar generation system that produces dual-stylized avatars instantly from a selfie.
- Developing Gaussian Domain Adaptation for enriching Snapmoji with a 3D prior, enabling dual-style transformations of selfies while preserving identity and style.
- Creating an animatable model by leveraging driving signals from 3DMM and blendshape priors combined with 3D Gaussians, enabling efficient, real-time rendering of dual-stylized avatars.
- Demonstrating, through extensive experiments, that our method outperforms existing solutions in both generation and animation performance, enabling real-time applications on mobile devices.

2. Related Work

2D Stylized Avatar Generation. In the realm of 2D avatar generation, neural networks like StyleGAN [21] are renowned for producing realistic images with interpretable latent spaces, as explored in works like [15, 58]. StyleGAN's versatility enables transformations into various styles, including Disney cartoons, paintings, and vintage photos [7, 31, 39]. A significant advantage of StyleGAN is its capability to perform these transformations without requiring paired domain images, a feature also utilized in SwiftAvatar [56], which creates paired data between realistic and stylized avatars. Diffusion models represent another prominent approach for 2D stylization, known for their larger architectures and enhanced diversity in generated content. Models such as Stable Diffusion [43] allow for image generation conditioned on text prompts, thereby increasing user control. Further advancements, including SDEdit [34], ControlNet [63], and IP Adapter [60], provide additional control through noise introduction or conditioning on structural inputs. Both GANs and diffusion models effectively convert real images into target styles, such as Bitmojis.

3D Content and Avatar Generation. Recent advances in 3D content creation have also significantly impacted avatar generation [26–28, 62]. 3D representations like NeRFs [36] and Gaussian Splats [22] have been integrated with generative models to automate the creation of 3D assets. For instance, DATID-3D [23] and StyleAvatar3D [61] employ 3D GANs to generate and stylize 3D facial models. More recent developments utilize text-to-image diffusion models for avatar stylization [9, 14, 16, 33, 37, 54, 65], though this process tends to be slow, taking approximately 10 minutes. In the realm of photorealistic avatar creation, techniques using Gaussian Splats [30, 32, 40, 44] fit real faces to 3D Morphable Models (3DMMs) like FLAME [25], but these do not adapt well to the geometry of cartoon avatars. Beyond avatars, work in general 3D object generation includes models like LRM [17] and LGM [55], which train end-to-end neural networks for mapping 2D images to 3D objects. These models provide much faster inference compared to diffusion models but rely on extensive internet-scale multi-view datasets, such as Objaverse [10], for training. This approach has yet to be applied to the specific challenge of 3D stylized avatar generation, which is a gap we seek to address.

Production Systems for Stylized 3D Avatar Creation. Developments in avatar creation platforms have introduced automated processes for selecting avatars by training classifiers that can predict avatar traits from a user's photograph [45]. Initially, these systems generate a basic version of an avatar, which users can then personalize by adjusting various traits to their preference. A significant challenge in training these classifiers is the need for paired data that links real faces to specific avatar traits, a requirement that is difficult to meet on a large scale. To address this challenge, approaches like AgileAvatar [46], F2P [48, 49], EasyCraft [57] and SwiftAvatar [56] have been developed, utilizing self-supervised learning techniques. While these methods are efficient, they are currently limited to creating

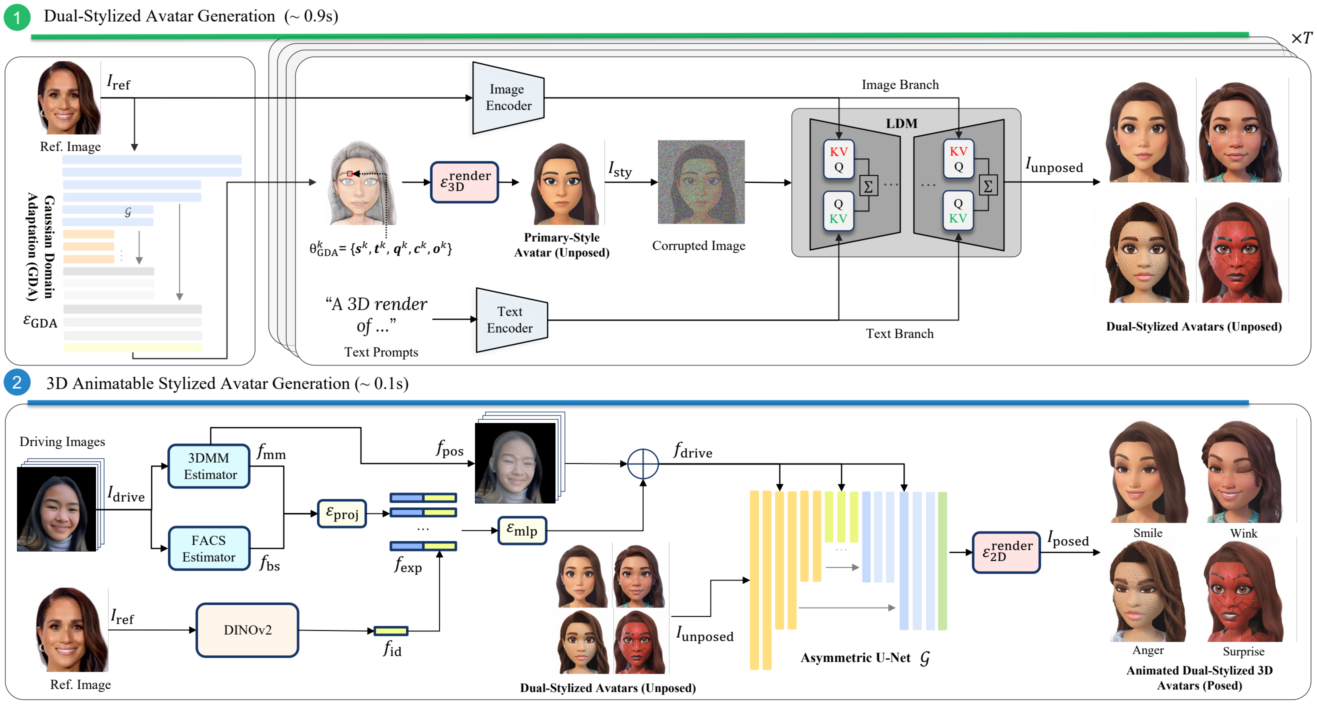
1
Dual-Stylized Avatar Generation (~ 0.9s)
×T
I_{ref}
Ref. Image
Gaussian Domain Adaptation (GDA)
$\mathcal{G}$
$\mathcal{E}_{\text{GDA}}$
Image Encoder
Image Branch
$\theta^k_{\text{GDA}} = \{s^k, t^k, q^k, c^k, o^k\}$
$\mathcal{E}^{\text{render}}_{\text{3D}}$
Primary-Style Avatar (Unposed)
I_{sty}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

avatars from pre-existing 3D asset libraries, rather than generating entirely new styles.

3. Method

Our method begins with the creation of datasets for real face images and their primary-style avatars (Sec. 3.1), facilitating avatar dual-stylization via Gaussian Domain Adaptation (Sec. 3.2). This framework supports 3D generation and animation of dual-stylized avatars (Sec. 3.3). Loss functions and 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 avatars in a primary style, which are not available at scale. To overcome this, we employ GAN inversion techniques inspired by unsupervised domain adaptation [56] to create synthetic paired data. By aligning the latent spaces of a source GAN and a fine-tuned target GAN [7, 56, 59], we generate corresponding pairs of realistic and primary-stylized images. Specifically, avatar 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-face 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. (See Fig. 9 and Fig. 10 in *Suppl.* for details).

3.2. 2D Dual-Stylized Avatar Generation

Gaussian Domain Adaptation $\mathcal{E}_{\text{GDA}}(\cdot)$. To bridge the domain gap between real photos and 3D-aware cartoonish avatars (*i.e.*, primary-style avatars), we first propose *Gaussian Domain Adaptation (GDA)*. Surprisingly, we find that features learned by Large Multi-view Gaussian models (LGMs) [55] can be quickly and robustly 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 first begin with a U-Net backbone from LGM trained on 3D objects. Then, we perform GDA by finetuning the network to map real face photos to primary-style avatar images in the frontal view. At inference time, for an input selfie, we first apply face alignment and background removal preprocessing. The preprocessed image is then passed through the asymmetric U-Net that takes the input reference image $I_{\text{ref}} \in \mathbb{R}^{3\times512\times512}$ and maps it to pixel-aligned Gaussian parameters, including scaling $\boldsymbol{s}$, position $\boldsymbol{t}$, color $\boldsymbol{c}$, opacity $\boldsymbol{o}$, and orientation $\boldsymbol{q}$:

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

where 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 avatar domain while preserving identity-related features, enabling seamless 3D-aware avatar generation and animation.

Avatar Dual-Stylization. While GDA efficiently generates avatars in a single primary style, our dual-stylization approach allows for additional customization. We employ a diffusion-based pipeline using SDEdit [34] to refine the GDA output image I_{sty} with minimal noise and guided denoising based on text prompts to add features like art style and accessories. To preserve the avatar's primary style, we use ControlNet [63] with Canny edges to maintain geometric integrity. Additionally, IP Adapter [60] is integrated with facial similarity embeddings to ensure resemblance to the original user. The cross-attention outputs for each layer, f_{out}^d, combine features from the reference image, text prompts, and the primary-stylized avatar:

$$\begin{aligned} f_{\text{out}}^d = \text{softmax}\left(\frac{(f_{\text{sty}}W_Q^d)(f_{\text{ref}}W_K^d)^T}{\sqrt{d_k^d}}\right)(f_{\text{ref}}W_V^d) \\ + \text{softmax}\left(\frac{(f_{\text{sty}}W_Q^d)(f_{\text{txt}}W_K^d)^T}{\sqrt{d_k^d}}\right)(f_{\text{txt}}W_V^d), \end{aligned} \quad (3)$$

where f_{ref}, f_{txt}, and f_{sty} correspond to the features of the reference image, text prompts, and the unstylized avatar, respectively. W_Q^d, W_K^d, and W_V^d are the weight matrices. Each cross-attention layer includes a residual connection (scaled by $\sqrt{d_k^d}$) for stable gradient flow. Using the DDIM scheduler [53], we perform only $T = 10$ denoising steps to rapidly integrate the secondary style in about one second.

3.3. 3D Animatable Stylized Avatar Generation

Expression Encoder. Current avatar animation techniques using Gaussian Splats often solely depend on 3D Morphable Models (3DMM) [8, 19, 40, 47, 54], which limits generalization beyond realistic faces, especially for stylized or cartoon avatars. To overcome these constraints, we condition the 3D generation network on a blend of 3DMM features and blendshape weights derived from the Facial Action Coding System (FACS) [13], which is widely utilized in cartoon animation to control facial features like eye position and mouth shape. 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:

$$f_{\text{drive}} = (\mathcal{E}_{\text{mlp}}([f_{\text{exp}}, f_{\text{id}}]), f_{\text{pos}}). \quad (4)$$

Here, f_{id} is the global identity feature extracted from a reference image I_r via a frozen DINOv2 backbone [38], 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 [55] and incorporate cross-attention layers to merge the driving features seamlessly:

$$I_{\text{posed}} = \mathcal{E}_{\text{2D}}^{\text{render}}(\mathcal{G}(I_{\text{unposed}}, f_{\text{drive}}; \Phi_g)), \quad (5)$$

where Φ_g is the network learnable parameter 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. Each block contains two ResNet layers with group normalization and SiLU activation. 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. The cross-attention is defined as:

$$f_{\text{out}}^g = \text{softmax}\left(\frac{(f_{\text{in}}^g W_Q^g)(f_{\text{drive}}W_K^g)^T}{\sqrt{d_k^g}}\right)(f_{\text{drive}}W_V^g), \quad (6)$$

where $f_{\text{in}}^g, f_{\text{out}}^g$ are the input and output features of the cross attention module in $\mathcal{G}(\cdot)$, respectively. W_Q^g, W_K^g, W_V^g are the learnable weight matrices and $\sqrt{d_k^g}$ the scale factor.

Mobile AR Application. Our model is also designed to facilitate real-time animation on mobile devices, striking a balance between advanced animation techniques and efficient rendering. 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:

$$\theta_{\text{mobile}} = \theta_{\text{rest}} + \sum_{i=1}^{K} f_{\text{drive}}^i(\theta_i - \theta_{\text{rest}}) \quad (7)$$

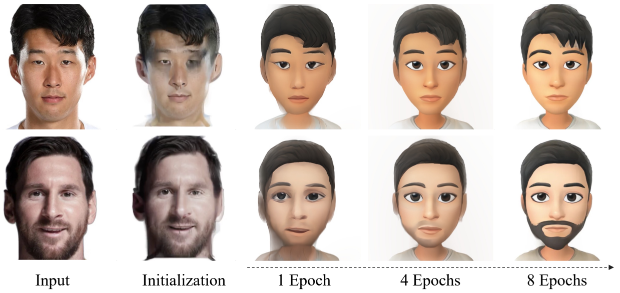

Figure 3. **Gaussian Domain Adaptation.** We show the outputs of the GDA network over several training epochs to visualize the domain shifts from natural images to cartoon avatars.

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

2D Dual-Stylized Avatar Generation. Our training process involves using GDA to map the input reference image I_{ref} to Gaussian parameters θ_{GDA}, which are then used to render the unposed primary-style avatar I_{unposed} from the frontal view via a 3DGS renderer $\mathcal{E}_{3D}^{\text{render}}$. The rendered image is supervised using a combination of Mean Squared Error (MSE) and perceptual LPIPS [64] losses:

$$\mathcal{L}_{\text{GDA}} = \mathcal{L}_{\text{mse}}(I_{\text{ref}}, I_{\text{sty}}) + \mathcal{L}_{\text{LPIPS}}(I_{\text{ref}}, I_{\text{sty}}). \tag{8}$$

Despite potential noise introduced by low-quality GAN inversion, the extensive 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, such as 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:

$$\mathcal{L}_{\text{3DGen}} = \mathcal{L}_{\text{render}} + \lambda_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}} + \lambda_{\text{n}}\mathcal{L}_{\text{normal}} + \lambda_{\text{d}}\mathcal{L}_{\text{dist}}, \tag{9}$$

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

$$\mathcal{L}_{\text{render}} = \|I_{\text{posed}} - I_{\text{posed}}^{\text{gt}}\|_2^2 + \|\alpha^{\text{pred}} - \mathbf{M}^{\text{gt}}\|_2^2. \tag{10}$$

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

$$\mathcal{L}_{\text{normal}} = 1 - (\mathbf{n}^{\text{pred}} \cdot \mathbf{n}^{\text{surf}}). \tag{11}$$

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.

4. Experiments

4.1. 2D Stylized Avatar Generation

Baselines. We evaluate GDA for 2D stylized avatar generation and compare with GAN inversion and diffusion-based methods, both fine-tuned on our Bitmoji dataset. For GAN inversion, we use a SemanticStyleGAN [50] model, translating real faces into avatars by inverting them into the latent space of a fine-tuned model. The diffusion-based method employs Stable Diffusion 1.5 [43] fine-tuned with LoRA [18], using BLIP-2 [24] for avatar captioning and IP Adapter Plus Face [60] for identity conditioning. Both methods aim to efficiently create primary-style avatars that retain the identity of the input images.

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, comparing the transformed images to the Bitmoji dataset. Identity retention was evaluated using ArcFace [11], while speed performance was benchmarked on an Nvidia L4 GPU. As shown in Table 2, GDA outperforms the other methods across all metrics, achieving FID scores more than 20 points lower than those of GAN inversion and diffusion. Fig. 4 visually highlights these quality differences: GAN inversion produces avatars with limited diversity, struggling to avoid generic outputs due to challenges in generating out-of-distribution images. 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 excels at producing avatars with a consistent style that retain key identity features such as eye color, sunglasses, and hairstyles. Its efficiency is noteworthy, requiring only a single forward pass through a U-Net, making it two and four orders of magnitude faster

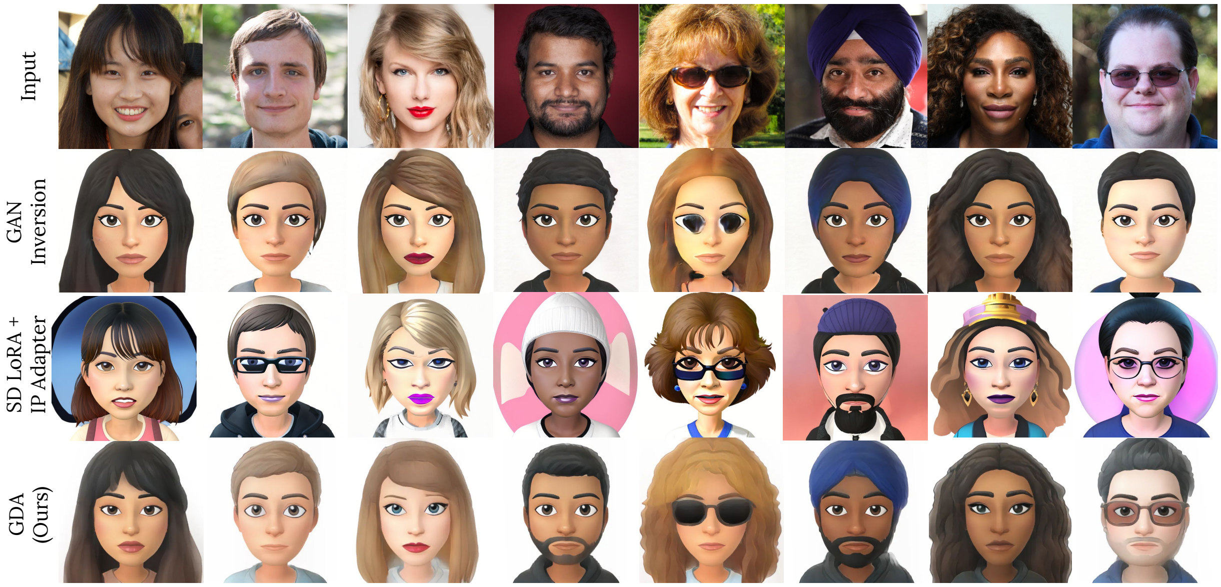

Figure 4. **2D Stylized Avatar Generation.** This figure showcases the transformation of photos from eight individuals into the Bitmoji domain using various methods. GAN inversion produces overly generic avatars, struggling with unique features such as beards, glasses, and headwear. Diffusion-based models 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.

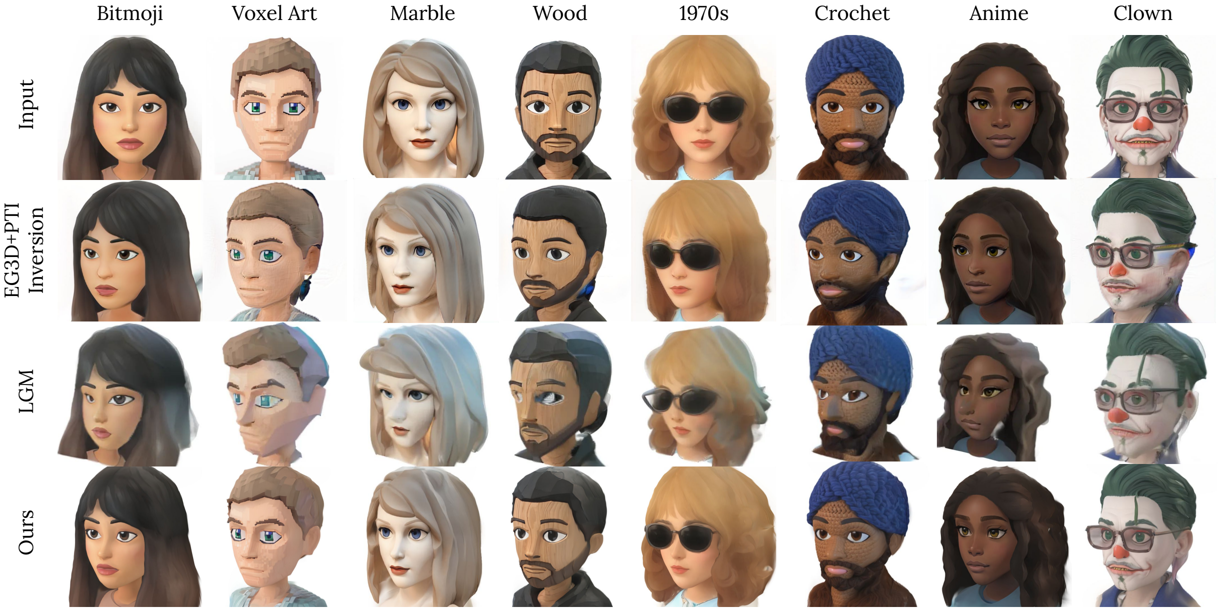

Figure 5. **2D Stylized Avatar to 3D Generation.** We demonstrate the process of converting dual-stylized avatar images, derived from the single-stylized avatars in Fig. 4, into 3D avatars. PTI inversion with EG3D [6, 42] struggles to accurately reproduce 3D geometry, while LGM [55] produces artifacts 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.

than diffusion and GAN inversion, respectively, with translations completed in under 0.1 seconds. Surprisingly, even though GDA uses data generated from GAN inversion for training, it produces images that are more detailed due to the learned 3D prior from the Objaverse [10] dataset, which enhances its generalization capability.

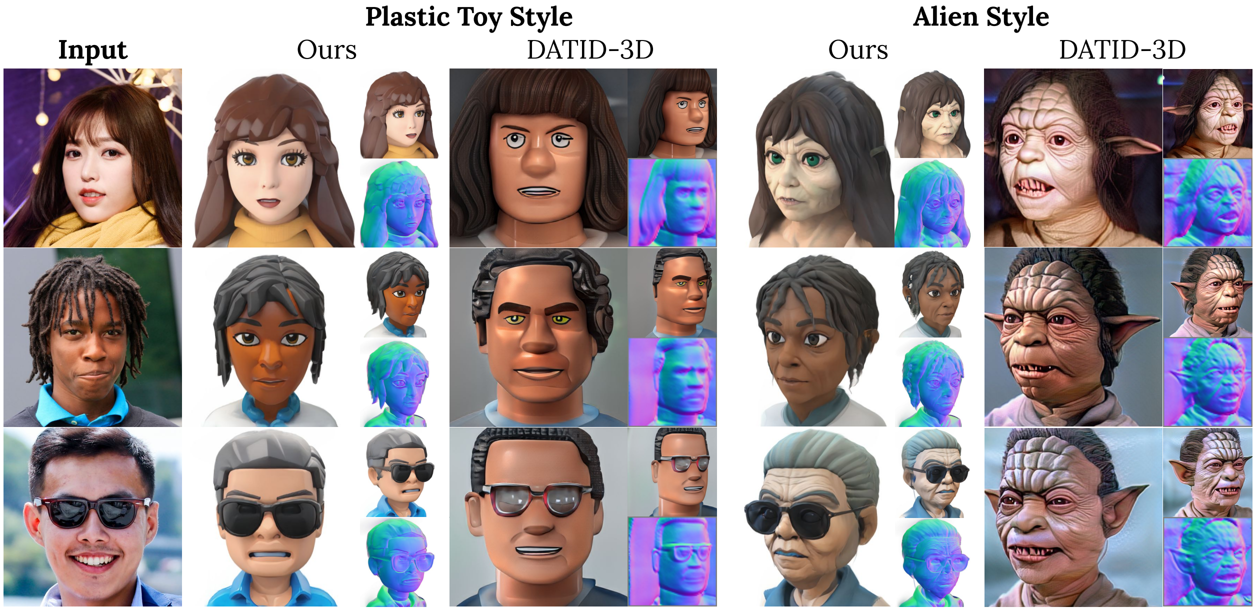

Figure 6. **Single Portrait to 3D Generation.** We compare Snapmoji with DATID-3D [23] in the context of 3D toonification. For each method and style, we render outputs from two viewpoints alongside a normal map. DATID-3D exhibits typical GAN-related issues, such as poor identity preservation and limited 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.

	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**

Table 2. **2D Stylized 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.

	PSNR ↑	SSIM ↑	LPIPS ↓	Speed ↓
EG3D [6]	10.92	0.68	0.50	95.1s
LGM [55]	12.16	0.69	0.53	2.82s
Ours	**18.73**	**0.81**	**0.24**	**0.091s**

Table 3. **2D Stylized Avatar to 3D Generation.** Our approach outperforms EG3D [6] and LGM [55] on all metrics, providing superior texture and geometry accuracy with faster processing.

4.2. 3D Dual-Stylized Avatar Generation

2D Stylized Avatar to 3D Generation. To evaluate the performance of generating a 3D avatar from a 2D stylized image, we compare our method against two other single-image 3D reconstruction techniques: EG3D [6] and LGMs [55]. EG3D, a 3D GAN based on the StyleGAN framework, generates a 3D neural radiance field and is fine-tuned on a multi-view 3D avatar dataset. It inverts a front-facing image into the GAN's $\mathcal{W}+$ space to render outputs from various viewpoints. LGM uses MVDream [51] to transform a single image into multiple viewpoints for input into a U-Net, which outputs 3D Gaussians. We assessed each method using 100 random 3D Bitmojis, each rendered from one front-facing view and ten additional views distributed spherically around the head. By inputting the front-facing view into each model, we calculated PSNR, SSIM, LPIPS [64], and speed metrics. 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. Fig. 5 provides visual comparisons on 3D *dual-stylized* avatars that fall outside the training distribution. The top row features eight stylized avatars generated using diffusion stylization from the identities in Fig. 4, with different style prompts as described in Sec. 3.2. Subsequent rows show each method's performance in translating these images to 3D. Even when employing PTI [42] for out-of-distribution images, EG3D struggles with high-fidelity geometry generation. Similarly, due to the diffusion process in MVDream, LGM often produces incorrect 3D head geometries. In contrast, our method successfully creates high-quality textures and geometry, even accommodating out-of-distribution accessories like turbans and sunglasses.

Single Portrait to 3D Generation. We evaluate Snapmoji's capability to generate a 3D avatar from a single portrait, with a comparison against DATID-3D [23]. DATID-3D uses a GAN to derive a latent code from the user's por-

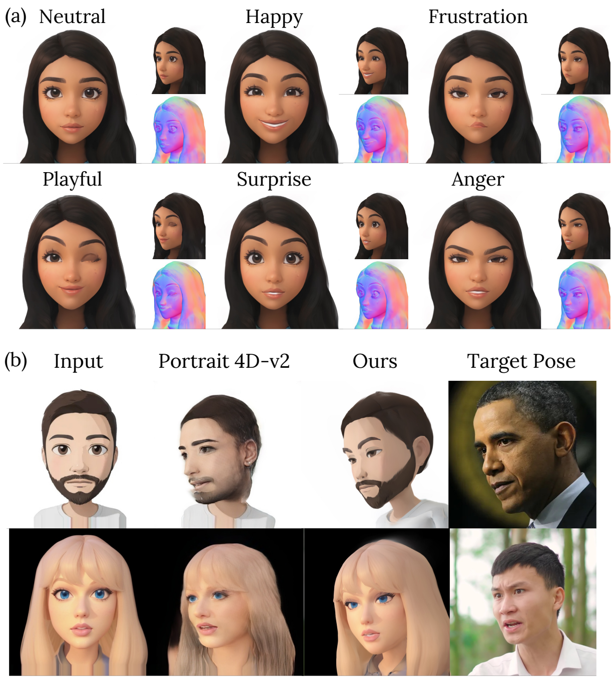

Figure 7. **3D Stylized Avatar Animation.** (a) An Snapmoji showcasing various emotions using blendshape weights. (b) Snapmoji effectively transfers expressions from driving images, outperforming Portrait4D-v2 [12] in accuracy and visual appeal.

trait, followed by domain adaptation for styling. Fig. 6 shows that DATID-3D struggles to maintain the original identity in avatars with distinct styles, like Plastic Toy and Alien. Snapmoji, however, achieves a robust balance of identity preservation and style versatility, allowing for enhanced user customization. Our approach produces sharp images and detailed geometries, processing each image in just 0.9 seconds, a significant improvement over DATID-3D's 90-second processing time for GAN inversion.

4.3. 3D Stylized Avatar Animation

Expression Animation. Snapmojienables 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, Snapmojican perform expression transfer from driving images, producing 3D-consistent and visually appealing avatars. Fig. 7(b) shows this capability, where Snapmojioutperforms Portrait4D-v2 [12] by generating avatars with more accurate expressions derived from the target image.

Mobile AR Application. We showcase a web-based AR app that demonstrates the efficient rendering of avatars on mobile devices. As illustrated in Fig. 8, an avatar animated using a user's facial expressions achieves rendering speeds of 30–40 FPS on an iPhone 13 Pro. These avatars are highly compact, occupying 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 [54]. Like many other avatar generation methods [8, 19, 40], TextToon relies solely on 3DMM features, achieving only 15–18 FPS on an M1 MacBook. In contrast, our method consistently runs at 90-100 FPS. Moreover, TextToon's dependence on 3DMMs limits its practicality on phones, whereas our cross-platform solution retains performance over 30 FPS. Table 4 offers a detailed feature comparison.

Method	Frame Rate (FPS)		Cross-Platform	Driving Signal
	M1 MacBook	iPhone 13 Pro		
TextToon [54]	15–18	N/A	×	3DMM
Snapmoji(Ours)	90-100	30–40	✓	3DMM + Blendshapes

Table 4. **Mobile AR Application Comparison.** We compare various features of our mobile AR application and TextToon [54].

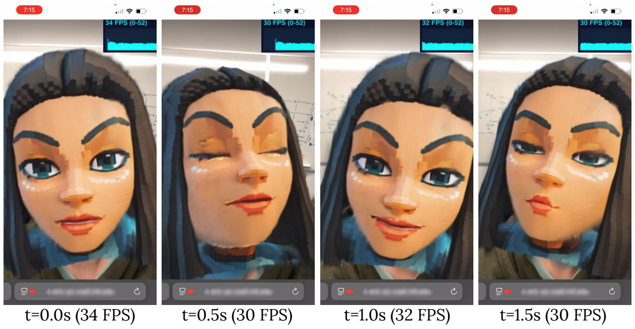

Figure 8. **Mobile AR Application.** Snapmoji's efficient mobile animation method enables a user to puppet their avatar in augmented reality.

5. Conclusion

We introduce Snapmoji, an easy-to-use system for generating animatable, dual-stylized avatars from selfies 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 supports 3D Gaussian avatars with dynamic animations and precise facial expression transfer, achieving selfie-to-avatar conversion in just 0.9 seconds, with real-time interactions at 30–40 FPS. Extensive testing confirms Snapmoji's versatility and speed, highlighting its value in creating diverse avatar styles.

Limitations and Future Work. Snapmoji relies on paired data from GAN inversion, which can sometimes yield low-quality images, and requires extensive 3D avatar datasets and text prompt engineering. Future improvements could include using multiple images for more accurate user head geometry and enabling post-stylization edits to specific facial features, like eye color or eyeglasses.

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. 2, 3, 6, 1

[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

[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] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *ArXiv*, abs/2004.02546, 2020. 2

[16] 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

[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. 2

[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. 1, 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. 2

[26] Di Liu, Xiang Yu, Meng Ye, Qilong Zhangli, Zhuowei Li, Zhixing Zhang, and Dimitris N Metaxas. Deformer: Integrating transformers with deformable models for 3d shape abstraction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14236–14246, 2023. 2

[27] Di Liu, Long Zhao, Qilong Zhangli, Yunhe Gao, Ting Liu, and Dimitris N Metaxas. Deep deformable models: Learning 3d shape abstractions with part consistency. *arXiv preprint arXiv:2309.01035*, 2023.

[28] Di Liu, Qilong Zhangli, Yunhe Gao, and Dimitris Metaxas. Lepard: Learning explicit part discovery for 3d articulated shape reconstruction. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[29] Di Liu, Bingbing Zhuang, Dimitris N. Metaxas, and Manmohan Chandraker. Instantaneous perception of moving objects in 3d. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024. 1

[30] 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. 1, 2

[31] 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

[32] 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. 1, 2

[33] 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

[34] 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

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

[36] 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

[37] 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

[38] 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

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

[40] 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. 1, 2, 4, 8

[41] Grand View Research. Digital avatar market size, share and growth report, 2030. https://www.grandviewresearch.com/industry-analysis/digital-avatar-market-report/, 2023. 1

[42] 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, 7

[43] 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, 5

[44] Shunsuke Saito, Gabriel Schwartz, Tomas Simon, Junxuan Li, and Giljoo Nam. Relightable gaussian codec avatars. In *CVPR*, 2024. 1, 2

[45] Samsung. How to use the AR Emoji feature on your Galaxy phone. https://www.samsung.com/sg/support/mobile-devices/how-to-use-the-emoji-feature-on-your-galaxy-phone/, 2018. Accessed: 2024-11-03. 2

[46] 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. 1, 2

[47] 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

[48] 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

[49] 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

[50] 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

[51] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv:2308.16512*, 2023. 7

[52] Snap Inc. Bitmoji 3d avatar platform solutions. https://developers.snap.com/lens-studio/platform-solutions/bitmoji-avatar/bitmoji-3d. 1

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

[54] 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

[55] 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. 2, 3, 4, 6, 7

[56] 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. 1, 2, 3

[57] 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. 1, 2

[58] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12858–12867, 2020. 2

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

[60] 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

[61] 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

[62] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. *ArXiv*, abs/2404.19702, 2024. 2

[63] 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

[64] 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, 7

[65] Qilong Zhangli, Jindong Jiang, Di Liu, Licheng Yu, Xiaoliang Dai, Ankit Ramchandani, Guan Pang, Dimitris N Metaxas, and Praveen Krishnan. Layout-agnostic scene text image synthesis with diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7496–7506. IEEE Computer Society, 2024. 2

Snapmoji: Instant Generation of Animatable Dual-Stylized Avatars

Supplementary Material

A. Implementation Details

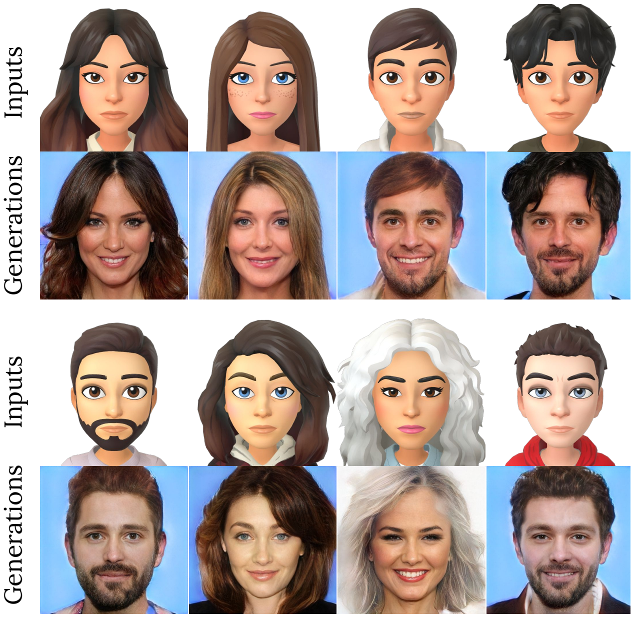

Figure 9. **GDA Training Data.** Visualization of data pairs used to train the GDA network. Each avatar is inverted into the latent space of a GAN, and a generator trained on realistic faces creates a corresponding realistic face, preserving features such as hairstyles, hair colors, and general facial characteristics.

A.1. Bitmoji Training Data

GDA Training Data. To train our Gaussian Domain Adaptation network, we start with a dataset of random Bitmojis and use GAN inversion to generate corresponding realistic images. Fig. 9 showcases some examples of the training data generated through this process. The resulting faces mirror the hairstyles, hair colors, and facial features of the original avatars. While the generated images may contain artifacts and exhibit limited diversity, our GDA model benefits from pre-training on Objaverse [10], enabling it to leverage prior knowledge and produce more detailed reconstructions than GAN inversion alone. This approach enhances the accuracy and expressiveness of the domain adaptation process.

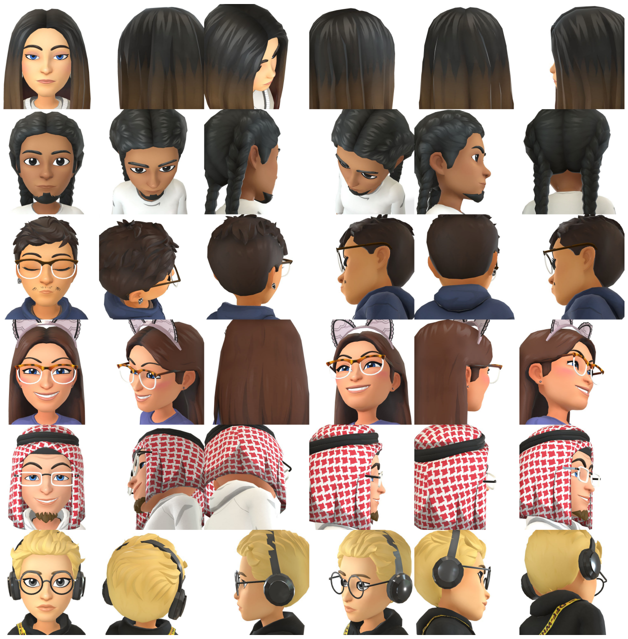

Figure 10. **Multi-view Bitmoji Training Data.** Samples from the Bitmoji dataset used in training the 3D Generation Network. Avatars are rendered from the front and multiple random angles around the head, with random blendshapes applied to simulate various expressions.

Multi-view Training Data. Fig. 10 visualizes training samples from the Bitmoji dataset used for training our 3D Generation Network. Each avatar is rendered from 10 spherically distributed viewpoints around the head and is posed with random blendshape weights to simulate diverse facial expressions. The dataset features a wide range of hairstyles, skin tones, and accessories such as glasses, hats, and earrings. Although the U-Net is trained exclusively on Bitmoji-style avatars, it effectively reconstructs dual-stylized avatars that exhibit distinct appearances and textures, demonstrating the network's versatility and generalization capability.

A.2. Facial Action Coding System

We implement the following 16 blendshapes from the Facial Action Coding System. These blendshapes are compatible with most facial blendshape predictors like *Apple ARKit*[*] or *Google Mediapipe*[*].

- `browDownLeft`
- `browDownRight`
- `browUpLeft`
- `browUpRight`
- `eyeBlinkLeft`

[*] https : / / arkit - face - blendshapes . com/

[*] https://ai.google.dev/edge/mediapipe/solutions/vision/face_land

- eyeBlinkRight
- jawOpen
- jawLeft
- jawRight
- lipsPucker
- mouthFrownLeft
- mouthFrownRight
- mouthSmileLeft
- MouthSmileRight
- mouthStretchLeft
- mouthStretchRight

To animate the Bitmoji avatars for training data, we used the publicly available Bitmoji rig available here: https://developers.snap.com/lens-studio/features/bitmoji-avatar/animating-bitmoji-3d.

At inference time, we can use a real-time blendshape predictor like *ARKit* or *Mediapipe* to puppet the avatars from a real video. Please see the attached HTML gallery for a demonstration.

A.3. User Interfaces

To showcase the intuitive features of the Snapmoji system, we present videos of the interface interactions available in the HTML gallery. The avatar generation interface, crafted with *Gradio*, enables users to effortlessly create dual-stylized avatars from their own photos. In addition, the blendshape editor, developed using *Viser*, allows users to pose their avatars in 3D by adjusting blendshape weights, thereby controlling facial expressions.

During the diffusion stylization process, we provide users with key parameters to balance identity preservation with style diversity. These controls are listed by their significance: 1) Style Transition Strength; 2) Edge Preservation Level 3) Identity Consistency Factor.

Style Transition Strength: This parameter, inspired by methods similar to SDEdit, regulates the extent of the stylization transition. Lower values enable the dual-stylized avatar to retain more details from the original single-styled avatar input.

Edge Preservation Level: This setting influences how accurately the system maintains the structure of the avatar by preserving the edges from the single-styled input.

Identity Consistency Factor: This controls the strength of identity features from the initial input photo, ensuring that essential facial characteristics remain recognizable.

We encourage users to view the videos in the HTML gallery to observe how these parameters affect avatar generation, enhancing both creativity and user experience.

B. Additional Results

B.1. Results Gallery

We invite you to explore the HTML gallery, which features videos of Snapmojiavatars animated in 3D. Access the gallery by opening the `index.html` file in your web browser. The gallery includes the following highlights:

1. Dual-stylized avatars with dynamic facial animations displayed from various novel viewpoints.
2. A demonstration of the avatars' capabilities in facial puppeting for augmented reality applications.
3. Screen captures of the Snapmojiuser interfaces, showcasing the ease of creating dual-stylized avatars and posing them using blendshapes.

B.2. More Applications

3D Avatar Animation. Dual-stylization offers the ability to swiftly visualize avatars in various scenarios, unlocking numerous applications. As illustrated in Fig. 1, Snapmoji avatars can be employed to create personalized comics and stickers, offering users a unique way to express themselves. Another promising application lies in augmented reality (AR), where avatars can be controlled and animated with real-time tracked facial expressions. Examples of this application are shown in Fig. 11 and within the HTML gallery. By utilizing *Mediapipe*'s real-time blendshape tracker, we animate the 3D avatars and seamlessly integrate them with video content, enabling them to be rendered in an AR environment through alpha compositing.

Real-time Web Rendering. Our choice to represent avatars using Gaussian Splats enables efficient real-time rendering on mobile devices. As demonstrated in Fig. 12 and in the HTML gallery, the avatars achieve a rendering rate of 90-100 FPS on a laptop, and 30-40 FPS on a phone. When paired with a face tracker, these avatars can be used to generate engaging filters and augmented reality effects. The demonstration showcases an avatar rendered in Google Chrome on a MacBook, entirely on the client side.

GDA Generalization. GDA demonstrates that the features learned from few-shot 3D reconstruction models are transferrable to new tasks. Shown in Fig. 13, GDA can be applied for more domains such as cats. We hope that GDA can inspire future work on using Gaussian features for other tasks.

B.3. Ablation Studies

3DMM Tracking. As shown in Fig. 14, our ablation study highlights the complementary strengths of 3DMM tracking and FACS-based blendshape features in avatar animation. 3DMM is adept at capturing realistic facial expressions, making it ideal for animating real faces, but it struggles with the exaggerated features typical of cartoon avatars.

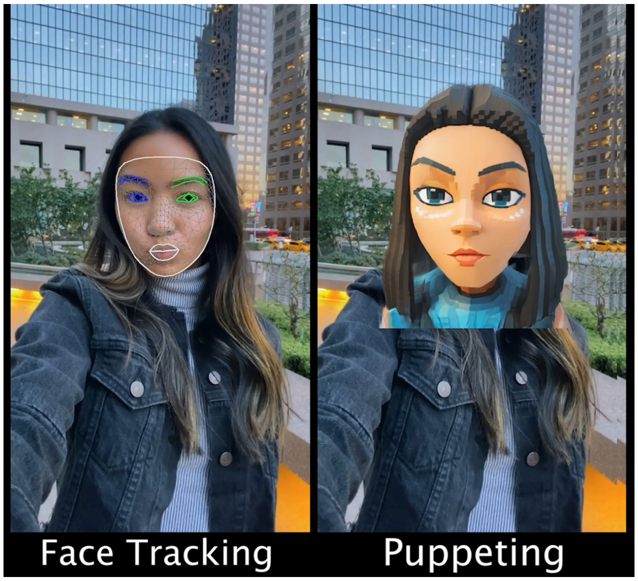

Figure 11. **Augmented Reality Puppeting.** This example demonstrates the use of *Mediapipe*'s real-time face detection to animate avatars based on estimated blendshape weights. By alpha-compositing the avatars with the original input, we enable dynamic puppeting in augmented reality. For live demonstrations, please refer to the HTML gallery.

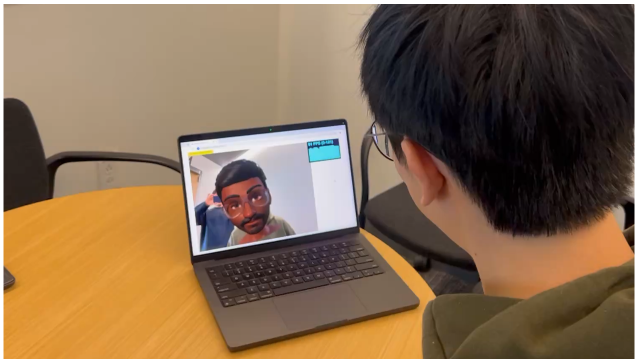

Figure 12. **Real-time Web Rendering.** Leveraging Gaussian Splats, our avatars efficiently render at 90–100 FPS on laptops and 30–40 FPS on mobile devices while occupying only 3 MB of disk space. In conjunction with a mobile face tracker, these avatars facilitate the creation of engaging filters and AR effects. The example shown features an avatar rendered in Google Chrome on an MacBook, developed in JavaScript.

Conversely, FACS blendshapes excel in stylizing facial elements, such as eye and mouth shapes, crucial for cartoon animation. By integrating the precision of 3DMM with the expressive capability of FACS blendshapes, we enhance the overall animation quality, enabling our avatars to faithfully portray both realistic and stylized expressions, thus delivering a more versatile and convincing animation experience.

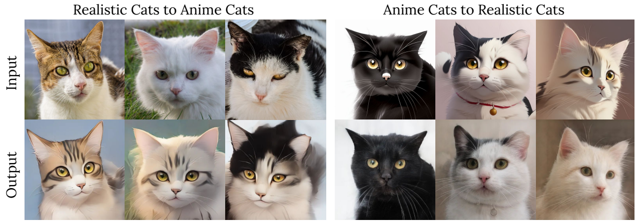

Figure 13. **GDA Generalization Across Domains.** This illustration showcases the versatility of Gaussian Domain Adaptation (GDA) as an image-to-image translation method. Demonstrated above is GDA's capability to transform realistic cat images into anime-style representations and vice versa, highlighting its potential for a wide range of applications beyond avatar creation.

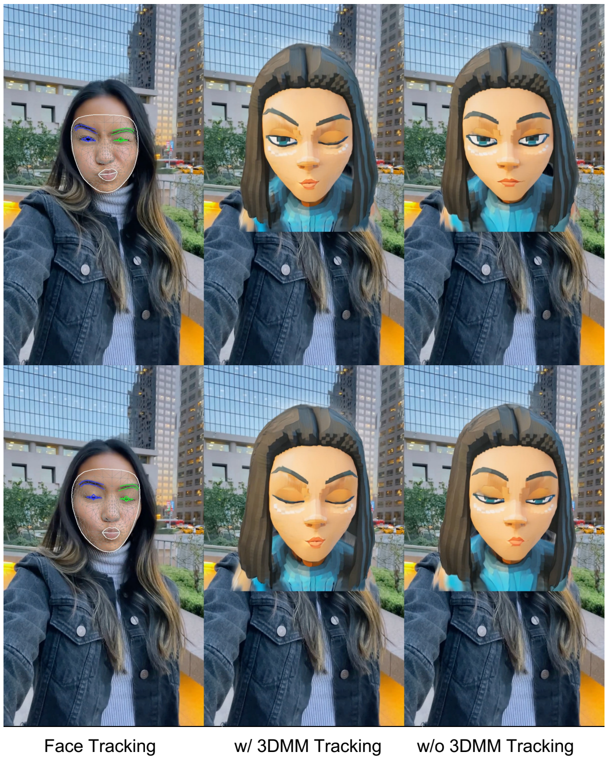

Figure 14. **Ablation Study on 3DMM Tracking.** This figure demonstrates the effects of using 3DMM features in conjunction with FACS blendshape weights. The combination enhances the expressiveness and fidelity of avatar animation, accommodating both realistic and stylized facial expressions.

C. Ethical Discussion

The use of photorealistic avatars has raised significant privacy and ethical concerns, particularly in relation to their potential misuse in creating deep fakes and spreading misinformation. In contrast, stylized cartoon avatars offer a safer alternative as they are not easily exploited for direct impersonation. In our work, we have prioritized user privacy by ensuring that no real person's images are used to train our models. Instead, the realistic images employed for training the Gaussian Domain Adaptation (GDA) system are generated by a GAN. We recognize, however, that GAN-generated data can reflect the biases present in the original datasets used for training. Consequently, we remain vigilant about these limitations and are committed to continuous evaluation and improvement to mitigate any unintended biases.