Bridging the Gap: Studio-like Avatar Creation from a Monocular Phone Capture

ShahRukh Athar[1,3*] Shunsuke Saito[2] Zhengyu Yang[2] Stanislav Pidhorskyi[2]
Chen Cao[2]

[1]Captions Research, New York [2]Meta Reality Labs, Pittsburgh.

[3]Stony Brook University, New York

shahrukh@nocapinc.com shunsuke.saito16@gmail.com
stpidhorskyi@meta.com zhengyu-yang@outlook.com
zju.caochen@gmail.com

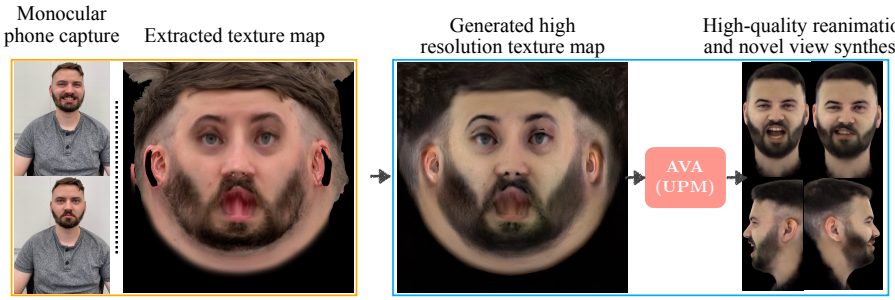

Fig. 1: From a low-resolution texture map obtained through monocular phone capture, our method produces a high-resolution texture map with precise facial details, studio-like illumination, and inpainted missing regions. This generated texture map can subsequently be utilized to create a high-quality, photorealistic avatar using the pretrained Universal Prior Model (UPM) from AVA [6].

Abstract. Creating photorealistic avatars for individuals traditionally involves extensive capture sessions with complex and expensive studio devices like the LightStage system. While recent strides in neural representations have enabled the generation of photorealistic and animatable 3D avatars from quick phone scans, they have the capture-time lighting baked-in, lack facial details and have missing regions in areas such as the back of the ears. Thus, they lag in quality compared to studio-captured avatars. In this paper, we propose a method that bridges this gap by generating studio-like illuminated texture maps from short, monocular phone captures. We do this by parameterizing the phone texture maps using the W^+ space of a StyleGAN2, enabling near-perfect reconstruction. Then, we finetune a StyleGAN2 by sampling in the W^+ parameterized space using a very small set of studio-captured textures as an adversarial training signal. To further enhance the realism and accuracy of facial details, we super-resolve

*Work done while interning at Meta Reality Labs

the output of the StyleGAN2 using carefully designed diffusion model that is guided by image gradients of the phone-captured texture map. Once trained, our method excels at producing studio-like facial texture maps from casual monocular smartphone videos. Demonstrating its capabilities, we showcase the generation of photorealistic, uniformly lit, complete avatars from monocular phone captures.

1 Introduction

Photorealistic and animatable avatars are paramount for lifelike human-to-human interactions in AR/VR applications. Creating high-fidelity avatars often requires sophisticated devices like the LightStage capture system that capture human heads with a range of facial expressions [27, 28]. Typically, these captures occur in a studio-like environment, with uniform illumination and densely sampled views to reconstruct complete avatars with consistent lighting. While these methods excel in generating hyper-realistic avatars, they cannot be scaled to millions of people as capturing so many people with a LightStage-like capture system is impractical and non-trivial.

Recently, there has been extensive work in generating photorealistic avatars through a monocular capture [3,6,14]. These methods involve scanning various facial expressions and head poses of the user to reconstruct a 3D avatar that closely aligns with the captured data. However, a notable limitation of these methods is that the captured lighting is embedded in the avatars, causing their appearance to be heavily dependent on the capture devices and surrounding environments. Furthermore, due to the constraints of the capture setup, certain areas, such as the back of the head or ears, are never visible, leaving visible holes and artifacts when viewed during animation.

An innovative strategy to tackle this challenge is to use image-to-image translation to transform the phone data into studio-captured data. This transformation can be learnt through supervised training using paired data [18], or through unsupervised training using unpaired datasets [42]. Due to reasons mentioned earlier, creating a large-scale paired dataset of studio and phone captures is impractical, which rules out the possibility of using supervised image-to-image translation methods. On the flip side, current unsupervised image-to-image translation methods fail to preserve fine detail in the transformation process [38, 42] which is paramount to creating a high-fidelity avatar.

In this paper, we introduce a method capable of generating studio-like, high-quality avatars from monocular phone captures. We do this by parameterizing a large-scale dataset of phone-captured face texture maps using the W^+ space of a StyleGAN2. Then, we finetune this StyleGAN2 by sampling from this parametrized W^+ space using a small set of unpaired studio-captured texture maps, to create a Studio-StyleGAN2 model. Our key insight is that sampling from the W^+ space instead of Z space leads to a more generalizable model as samples from the W^+ are, by construction, as diverse as the training set while samples from the Z space often suffer from mode-collapse. During inference, the given phone-captured texture map, is first inverted to the W^+ space of the StyleGAN2. This inverted W^+ vector is then given as input to the Studio-StyleGAN2 to generate a low-resolution studio-like texture map. Finally, a novel facial detail conditioned diffusion model is used to enhance facial details of the low-resolution

studio-like texture map obtained in the previous step. Once trained, our method excels at producing high-quality studio-like face texture maps from monocular phone captures, which are then used as inputs to the universal face prior model (UPM) from Authentic Volumetric Avatars (AVA) [6] to generate photorealistic avatars. In summary, our key contributions are as follows:

- Introducing a groundbreaking method for creating studio-like, photorealistic avatars from monocular phone captures.
- Innovative finetuning of a pre-trained generative model using a minimal dataset from another domain by sampling in the inverted W^+ space, which facilitates the development of a generative model for a new domain while preserving the integrity of the W^+ latent space.
- A novel diffusion model conditioned on the phone texture gradient, that is designed to upsample studio textures, effectively enhancing facial details and contributing to the overall realism of the generated avatars.

2 Related Work

Our paper aims to bridge the gap between a studio capture and a phone capture by creating a studio-like, photorealistic avatar through a monocular phone capture. In delineating the related work, we offer a comprehensive overview of key research domains, including studio-captured avatars, phone-captured avatars, and image-to-image translation.

Studio-captured avatars. The reconstruction of high-fidelity static and dynamic models of the human head based on photometric measurements has a long-standing history in computer graphics and vision. Achieving a photorealistic human avatar often requires specialized hardware in high-end production, such as the LightStage system. To model the complex skin appearance, various physically-based models have been explored. Notably, subsurface scattering [5], linear polarization patterns [12] and fine-scale skin details [1, 2] have been investigated. For dynamic expression details, Jimenez et al. [20] compute dynamic skin appearances by blending hemoglobin distributions captured with different expressions. In their subsequent work [19], expression-dependent normal maps are interpolated to add realistic wrinkles to an animated face. Nagano et al. [32] synthesize skin microstructures based on local geometric features derived from high-precision microgeometry, acquired with an LED sphere and a skin deformer. While these methods have been instrumental for offline movie production, their substantial compute requirements make them less suitable for real-time applications. Despite recent efforts to enable real-time rendering of physically-based avatars [36], heavy compute remains challenge. In response to the challenges posed by complex physical computations, researchers have proposed a deep appearance model [27]. This model utilizes a coarse 3D triangle mesh in conjunction with view-dependent texture mapping. The texture is regressed by a neural network conditioned on viewpoint and expression latent codes. This conditioning accounts for view- and expression-dependent variations while compensating for the imperfect proxy geometry. Subsequent work extends the mesh-texture representation to a volumetric representation using a Mixture of Volumetric Primitives (MVP) [28], further enhancing the model's quality. Pixel Codec Avatars (PiCA), as demonstrated by [30], showcase the efficiency of rendering such models, even on mobile hardware platforms,

by leveraging efficient per-pixel processing. Moreover, Bi et al. [4], based on relighting captured data, can also relight avatars with any novel point light or environment maps. While these methods can achieve hyper-realistic avatars, their studio requirements make them challenging to generalize to ordinary users.

Phone-captured avatars. There are several methods aimed at creating avatars in lightweight ways, even from a phone or a single image. Some of these approaches focus on generating stylized avatars based on different input requirements, ranging from capturing multi-view images using a phone [17] to utilizing a single monocular image [26, 29, 35, 37]. While these methods excel in producing animatable avatars, it is important to note that their appearance tends to be cartoonish and lacks realism.

Another category of methods is dedicated to creating realistic human avatars based on graphics pipelines [7, 8, 11, 15, 24, 25, 41]. While relying on traditional graphics methods, these approaches often result in avatars that appear uncanny. Building upon deep learning-based representations, researchers have proposed methods to generate avatars with increased realism, including notable works by Gafni et al. [10] and Grassal et al. [14]. Among these, the work of Cao et al. [6], Authentic Volumetric Avatars (AVA), stands out for its focus on creating photorealistic avatars from phone scans. The process involves training a Universal Prior Model (UPM) using studio-captured data, followed by personalizing this UPM using data from a phone scan of an unseen subject. While the method successfully produces avatars with realistic appearance and animation, it is worth noting that the lighting is baked into the personalized avatar, and certain details, such as the back of the ears, may be missing.

Image-to-image translation We use image-to-image translation to map images from the source domain (phone data) to the target domain (studio data). Isola et al. introduced Pix2Pix [18], a method that utilizes adversarial training strategies [13] to achieve this mapping. Additionally, Wang et al. [40] focused on increasing the resolution of generated results from semantic label maps. While these methods successfully map images between domains, it is essential to note that they require paired training data for effective implementation. In many real-world scenarios, obtaining paired training data can be challenging and expensive.

To address this issue, unsupervised image-to-image translation has been introduced. Zhu et al. [42] proposed a novel cycle-consistency loss to ensure that translating an image from one domain to another and back again should result in the original image. This helps the model maintain consistency and produce more realistic translations. Subsequent methods have further improved unsupervised image-to-image translation from different perspectives, including a multimodal model [16], few-shot input for video-to-video translation [39], translation of images with human control [34], and translation of real images into different styles [38]. However, none of these methods are designed with preservation of facial identity and the generation of realistic facial details in mind, especially with such little training data. Later in the paper we show how this prior work compares to ours for transforming low-resolution phone-captured texture maps to high-resolution studio-captured texture maps.

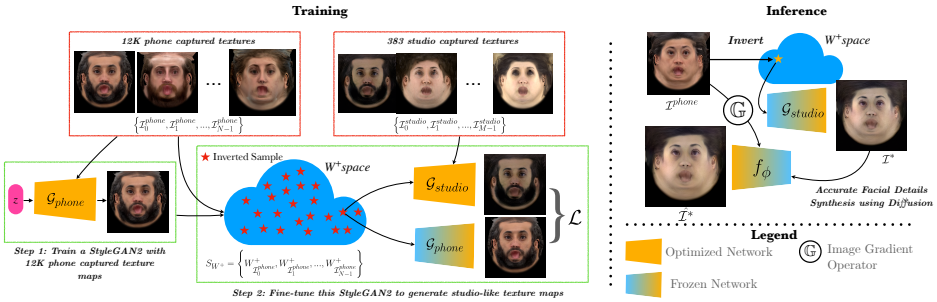

Fig. 2: Method Overview. Our method employs a two-step process to train $\mathcal{G}_{Studio}$. Initially, we train a StyleGAN2 on 12k neutral texture maps captured by phones, yielding $\mathcal{G}_{phone}$. Subsequently, we initialize $\mathcal{G}_{Studio}$ with the weights of $\mathcal{G}_{phone}$ and fine-tune it by sampling from $S_{W^+} = \{W^+_{\mathcal{I}_0^{phone}}, \ldots, W^+_{\mathcal{I}_{N-1}^{phone}}\}$;, where $N = 12k$, and $W^+_{\mathcal{I}_i^{phone}}$ represents the vector obtained by inverting the i'th phone-captured texture map, $\mathcal{I}_0^{phone}$, in the W^+ space of $\mathcal{G}_{phone}$. During inference, the given phone-captured texture map is inverted to W^+, then passed to $\mathcal{G}_{Studio}$ to generate a low-resolution studio-lit texture map, $\mathcal{I}^*$. Accurate facial details are subsequently added using diffusion model f_ϕ, conditioned on the gradient of the phone texture. This process results in the final high-resolution, studio-lit, and completed texture map, $\mathcal{I}^*$.

3 Method

In this section, we present our method for generating studio-like avatars from the phone captures. Our approach consists of two key components: a StyleGAN2 for texture translation and a diffusion model for facial detail generation. In Sect 3.1, we describe the generation of a low-resolution texture map with studio-like lighting and missing regions inpainted. First, we pretrain a StyleGAN2 on 12k phone textures and then finetune it using a small set of studio-captured texture maps. In order to improve the generalization of the finetuned StyleGAN2, we optimize it by sampling in the W^+ space, instead of the W-space or Z-space, using a set of 12k W^+ vectors obtained by inverting the phone captured texture maps. In Sect 3.2, we introduce a diffusion model that generates facial details. The diffusion model takes the output from the aforementioned StyleGAN2 and generates a high-resolution neutral texture with realistic facial details. After obtaining a high-resolution studio-like neutral texture from our method, we use it to learn a color transform to transfer phone-captured expression textures to studio-lit expression textures. These expressions are subsequently utilized for animating a high-quality avatar (Sect 3.3). Notably, we intentionally avoid applying inpainting or super-resolution techniques to the expression textures. Fig. 2 provides an overview of our method.

3.1 Illumination manipulation and inpainting

In our initial step, we track the geometry from a monocular phone capture of the user's neutral face and extract the neutral texture $\mathcal{I}^{phone}$ from the captured image, employing the method outlined in [6]. Subsequently, we translate this phone-captured texture with in-the-wild lighting and missing regions into a texture map with studio lighting and

missing regions in-painted. This translation is accomplished by parametrizing $\mathcal{I}^{phone}$ using the W^+ space of a StyleGAN2, as follows:

$$W^+_{\mathcal{I}^{phone}} = \underset{W^+}{\operatorname{argmin}} ||\mathcal{G}_{phone}(W^+) - \mathcal{I}^{phone}||_2^2 + LPIPS(\mathcal{G}_{phone}(W^+), \mathcal{I}^{phone}), \tag{1}$$

where $\mathcal{G}_{phone}$ is a StyleGAN2 trained on phone-captured textures, and $W^+_{\mathcal{I}^{phone}}$ is the optimized vector in the W^+ space of $\mathcal{G}_{phone}$. Now, we generate the low-resolution studio-like texture as follows:

$$\mathcal{I}^* = \mathcal{G}_{Studio}(W^+_{\mathcal{I}^{phone}}), \tag{2}$$

where $\mathcal{G}_{Studio}$ is a StyleGAN2 responsible for generating low-resolution studio-like textures, while $\mathcal{I}^*$ refers to the studio-lit version of $\mathcal{I}^{phone}$ with the inpainted missing regions. Ideally, $\mathcal{I}^*$ should retain the identity and semantics of $\mathcal{I}^{phone}$, with the only distinctions being the illumination and the inpainting of missing regions. Below we describe the training procedure for $\mathcal{G}_{Studio}$.

We initialize the synthesis network of $\mathcal{G}_{Studio}$ using weights from $\mathcal{G}_{phone}$, which we then finetune to generate texture maps with studio lighting and inpaint the missing regions. We choose to finetune $\mathcal{G}_{phone}$ for the following reasons: 1) Due to the immense costs of a studio capture, studio quality training data is typically very limited. In our case, we only have 383 studio-captured texture maps, making training from scratch hard [22]; 2) Since we want $\mathcal{I}^*$ and $\mathcal{I}^{phone}$ to have the same facial identity and semantics meaning, it is beneficial to learn a latent space that is shared between $\mathcal{G}_{Studio}$ and $\mathcal{G}_{phone}$. For finetuning, we first invert the entire dataset of N phone-captured texture maps using Eq. (1), giving us a set of N vectors in the W^+ space: $S_{W^+} = \{W^+_{\mathcal{I}^{phone}_0}, \ldots, W^+_{\mathcal{I}^{phone}_{N-1}}\}$. Next, we randomly sample vectors from the inverted set, S_{W^+}, and finetune the synthesis network of $\mathcal{G}_{phone}$ to give $\mathcal{G}_{Studio}$. We choose to finetune the generator with samples from inverted W^+ instead of the traditional W space, since they are, by construction, from the real data distribution and are consequently more diverse, leading to better generalization of $\mathcal{G}_{Studio}$. The losses we use during finetuning are described as below.

Studio-Discriminator loss. This loss uses a discriminator to ensure the texture maps generated by $\mathcal{G}_{Studio}$ are from the distribution of studio-captured texture maps:

$$\begin{aligned}\mathcal{L}_{Adv} = \mathbb{E}_{\mathcal{I}^{Studio} \sim P(\mathcal{I}^{Studio})}\left[min(0, -1 + \mathcal{D}_{Studio})\right] \\ + \mathbb{E}_{W^+_{\mathcal{I}^{phone}_i} \sim S_{W^+}}\left[min(0, -1 - \mathcal{D}_{Studio}(\mathcal{G}_{Studio}(W^+_{\mathcal{I}^{phone}_i}))\right],\end{aligned} \tag{3}$$

where $\mathcal{D}_{Studio}$ is a discriminator that initialized from the pretrained phone-captured texture StyleGAN2 Discriminator and finetuned using ground-truth studio-captured textures and textures generated by $\mathcal{G}_{Studio}$.

Face-Identity Loss This loss compares the face identity embeddings of the renders of texture maps generated from $\mathcal{G}_{Studio}$ and $\mathcal{G}_{phone}$ in order to ensure the identity is preserved:

$$\mathcal{L}_{FaceID} = \left\| |\mathcal{F}\left(\mathcal{G}_{Studio}(W^+_{\mathcal{I}^{phone}_i})\right) - \mathcal{F}\left(\mathcal{G}_{phone}(W^+_{\mathcal{I}^{phone}_i})\right) \right\| |_2^2, \tag{4}$$

where $\mathcal{F}$ is a pretrained face recognition network[1].

[1]We use the network from here

Perceptual Loss This loss encourages the preservation of semantic features between the texture maps generated $\mathcal{G}_{Studio}$ and $\mathcal{G}_{phone}$ by minimizing their distance in the VGG feature space

$$\mathcal{L}_{Percp} = \text{LPIPS}\Big(\mathcal{G}_{Studio}(W^{+}_{\mathcal{I}_i^{phone}}), \mathcal{G}_{phone}(W^{+}_{\mathcal{I}_i^{phone}})\Big). \tag{5}$$

Perceptual Reconstruction Loss This loss ensures that skintones are preserved using a small amount of paired data (i.e subjects for whom we have both phone and studio textures maps i.e both $\mathcal{I}^{phone}$ and $\mathcal{I}^{Studio}$)

$$\mathcal{L}_{Percp-Recons} = \frac{1}{k}\sum_{i=1}^{K} \text{LPIPS}\Big(\mathcal{G}_{Studio}(W^{+}_{\mathcal{I}_i^{phone}}), \mathcal{I}_i^{Studio}\Big), \tag{6}$$

where $\mathcal{I}_i^{Studio}$ and $\mathcal{I}_i^{phone}$ are the ground-truth studio and ground-truth phone texture maps of the same person. We only have a very small amount of this data for training (K=83). *R1-Regularization*: We regularize $\mathcal{D}_{Studio}$ using the R1 regularization [31] as follows:

$$\mathcal{L}_{R1} = -\frac{\gamma}{2}\mathbb{E}_{\mathcal{I}^{Studio}\sim P(\mathcal{I}^{Studio})}\left[||\nabla \mathcal{D}_{Studio}(\mathcal{I}^{Studio})||_2^2\right], \tag{7}$$

where $\gamma = 10$.

In order to further encourage identity preservation, we leverage the multi-resolution architecture of StyleGAN2 [23] and only optimize the generator parameters after the 8×8 resolution during the finetuning process. The intuition being that identity-specific information is mostly stored in the low-resolution maps, we ablate this choice in the Supplementary section of the paper. The final optimization for the joint finetuning of $\mathcal{G}_{Studio}$ and $\mathcal{D}_{Studio}$ is the following:

$$\min_{\mathcal{G}_{Studio}^{\theta(8+)}} \max_{\mathcal{D}_{Studio}} \mathcal{L}_{Adv} + \mathcal{L}_{R1} + \mathcal{L}_{Percp-Recons} + \lambda_1 \mathcal{L}_{Percp} + \lambda_2 \mathcal{L}_{FaceID}, \tag{8}$$

where $\lambda_1 = 0.5$ and $\lambda_2 = 10$. We ablate $\mathcal{L}_{Percp-Recons}, \mathcal{L}_{Percp}, \mathcal{L}_{FaceID}$ in the supplementary.

3.2 Synthesis of accurate facial details

Generating high-quality facial details. $\mathcal{G}_{Studio}$ can produce an inpainted texture map, $\mathcal{I}^*$, with illumination consistent with that of a studio capture. However, due to the limited amount of finetuning data, $\mathcal{G}_{Studio}$ often struggles to reproduce facial details that are visibly present in $\mathcal{I}^{phone}$ itself. This limitation results in oversmoothed and inaccurate avatars. Achieving realistic facial details necessitates an accurate modeling of the facial texture distribution, a task that the StyleGAN2 fails to do.

Motivated by the recent success of latent diffusion models in accurately modeling data distributions, we propose the task of generating facial details as the reverse process of a Markov chain that transforms a low-resolution face texture map to a high-resolution one. We adopt the formulation from [43], where the residual between the low-res and high-res studio-captured texture maps, denoted as $e_0 = \mathcal{I}_{LR}^{Studio} - \mathcal{I}^{Studio}$, is used to define the Markov process. More specifically, the forward process is defined as follows:

$$\begin{gathered} q(\mathcal{I}_t^{Studio}|\mathcal{I}_{t-1}^{Studio}, \mathcal{I}_{LR}^{Studio}) = \mathcal{N}\Big(\mathcal{I}_t^{Studio}; \mathcal{I}_{t-1}^{Studio} + \alpha_t e_0, \kappa^2 \alpha_t I\Big), \\ t = 1, 2, \ldots, T, \end{gathered} \tag{9}$$

where α_t controls the schedule with which the residual is added, κ is a hyperparameter controlling the noise schedule and I is the identity matrix. The reverse process is defined as

$$q(\mathcal{I}_{t-1}^{Studio}|\mathcal{I}_{\{t,LR\}}^{Studio}, \mathcal{I}^{Studio}) = \mathcal{N}\left(\mathcal{I}_{t-1}^{Studio} \middle| \frac{\eta_{t-1}}{\eta_t}\mathcal{I}_t^{Studio} + \frac{\alpha_t}{\eta_t}\mathcal{I}^{Studio}, \kappa^2\frac{\eta_{t-1}}{\eta_t}\alpha_t I\right), \tag{10}$$

where $\eta_t = \alpha_t + \eta_{t-1}$. We refer the reader to [43] for details regarding the formulation of both the forward and reverse process. Finally, the diffusion model, f_ϕ, is trained to minimize the following objective

$$\min_{\phi} \textstyle\sum_t ||f_\phi(\mathcal{I}_t^{Studio}, \mathcal{I}_{LR}^{Studio}, t) - \mathcal{I}^{Studio}||_2^2 \tag{11}$$

During inference, we use f_ϕ to add realistic facial details to the low-resolution output of $\mathcal{G}_{Studio}$ as follows:

$$\hat{\mathcal{I}^*} = \text{ReverseProcess}\left(f_\phi, \mathcal{I}^*\right). \tag{12}$$

Recovering accurate facial details. While f_ϕ enhances $\mathcal{I}^*$ by adding realistic facial details, it struggles to recover details already present in the phone-captured texture map $\mathcal{I}^{phone}$ due to the inherent lack of such details in $\mathcal{I}^*$. The loss of details occurs during the illumination manipulation and inpainting process, as described in Eq. (1) and Eq. (2). To address this issue, we incorporate the image gradient from the phone-captured texture map into the low-resolution texture map during the training of the diffusion model, as follows:

$$\mathcal{I}_{LR}^{Studio*} = \mathcal{I}_{LR}^{Studio} + \mathbb{G}(\mathcal{I}^{phone}) \tag{13}$$

where $\mathcal{I}^{phone}$ represents the phone-captured texture map, and $\mathcal{I}^{Studio}$ is the studio-captured texture map of the same person. $\mathbb{G}$ denotes the operator used to calculate the image gradient. During training, $\mathcal{I}_{LR}^{Studio*}$ replaces $\mathcal{I}_{LR}^{Studio}$ in Eq. (9), Eq. (10), and Eq. (11). In the inference stage, we augment $\mathcal{I}^*$ by adding the gradient of the phone-captured texture map as follows:

$$\hat{\mathcal{I}^*} = \text{ReverseProcess}\left(f_\phi, \mathcal{I}^* + \mathbb{G}(\mathcal{I}^{phone})\right) \tag{14}$$

Implementation details We utilize 83 paired texture maps, representing subjects for whom we have both $\mathcal{I}^{Studio}$ and $\mathcal{I}^{phone}$, to train the diffusion model. Following the approach in [43], we employ a latent diffusion model. To prevent overfitting, training is conducted on random 512×512 crops of the 1024×1024 resolution texture map. During the inference stage, we employ the full-resolution texture map.

3.3 Driving a high-quality avatar

With the studio-lit version $\mathcal{I}^*$ now available for a given neutral phone-captured texture map $\mathcal{I}^{phone}$, we proceed to estimate a color transform mapping from the phone-captured texture map to the studio-lit texture map as follows:

$$\{G, B\} \leftarrow \underset{\{G,B\}}{\text{argmin}} ||\mathcal{I}^* - \left(\texttt{Rsz}(G) \times \mathcal{I}^{phone} + \texttt{Rsz}(B)\right)||. \tag{15}$$

Here, G and B represent gain and bias maps of resolution $k \times k$ and $\texttt{Rsz}$ is the resizing operator with bilinear interpolation. Utilizing G and B, we perform a transformation

on phone-captured expression texture maps to achieve studio-like lighting, as outlined below:

$$\mathcal{I}^*{}_{exp} = \texttt{Rsz}(G) \times \mathcal{I}^{phone}_{exp} + \texttt{Rsz}(B). \tag{16}$$

In our experiment, we select the value of $k = 32$ to efficiently transform the lighting while preserving the details.

Now, the studio-like high-resolution neutral texture $\hat{I^*}$ serves as conditioning data for the universal avatar prior from [6]. By combining the expression code generated by inputting $\mathcal{I}^*{}_{exp}$ into the expression encoder from [6], we can render a high-quality avatar from any desired view v as follows:

$$\mathbf{I} = \text{AVA}(\hat{\mathcal{I}^*}, \mathcal{I}^*{}_{exp} - \mathcal{I}^*, v, \mathbf{F}), \tag{17}$$

where $\mathbf{I}$ represents the render of the avatar, and $\mathbf{F}$ is the geometry generated during 3D face tracking for a monocular phone capture. AVA corresponds to the inference process of the universal prior model. For more details about the universal prior model, please refer to [6].

4 Results

In this section, we introduce the dataset used in this paper, along with the baselines of our method. Subsequently, we present both quantitative and qualitative results of our method, comparing it to prior work. All phone-captured texture maps are generated using the mesh fitting procedure outlined in Authentic Volumetric Avatars (AVA) [6]. Furthermore, we utilize AVA to render avatars based on the texture maps generated by all the methods, facilitating the calculation of image space metrics, including face identity similarity.

Table 1: Quantitative results on paired phone-cum-studio captured texture maps. Our method outperforms prior work across all metrics. Best and Second Best scores are highlighted.

Models	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
Ours	22.76	0.726	0.364	0.163
AgileGAN [38]	18.05	0.657	0.406	0.180
CUT [33]	21.440	0.642	0.402	0.169
CycleGAN [42]	21.087	0.643	0.400	0.175

Table 2: FaceID results on unpaired phone captured texture maps. Our method better preserves identity than prior work without sacrificing the quality of the generated texture maps. Best and Second Best scores are highlighted.

Models	Ours	AgileGAN	CUT	CycleGAN
FaceID	$4.31e-4$	$1.36e-3$	$6.89e-4$	$5.19e-4$

Training and Evaluation Data. We utilize a dataset comprising 12,543 neutral phone-captured texture maps to train $\mathcal{G}_{phone}$, and 383 studio-captured texture maps for fine-tuning $\mathcal{G}_{phone}$ to obtain $\mathcal{G}_{Studio}$. Among the 383 maps, 83 are paired neutral texture maps $\{\mathcal{I}^{phone}, \mathcal{I}^{Studio}\}$ used for calculating the perceptual reconstruction loss, as described in Eq. (6). This paired dataset also serves for training the detail-preserving diffusion model. For quantitative and qualitative evaluation, we employ 10 paired phone-cum-studio captured texture maps and 31 phone-captured texture maps.

Baselines. We compare our method to the following prior works on unpaired image-to-image translation. 1) AgileGAN [38], which utilizes an aligned StyleGAN latent space

to stylize images, even with very few examples; 2) Contrastive Unpaired Translation (CUT) [33], which employs a patch-based contrastive loss in a learned feature space, enabling domain adaptation, even for single images; 3) CycleGAN [42], which trains two generators based on a cycle-consistency loss to translate images between two domains. Since AgileGAN [38] has no code available, we implement it ourselves. We use publically available code for CUT [33] and [42] for all experiments.

Fig. 3: Comparisons with baselines on paired phone-cum-studio texture data reveal that our method produces results closest to the ground truth. It achieves uniform studio-like lighting, well-reconstructed facial details visible in the phone capture, and effective inpainting of missing regions. In contrast, AgileGAN [38] fails to preserve identity, while CUT [33] and CycleGAN [42] introduce significant artifacts.

Quantitative Results. In Table 1, we present the results of a quantitative evaluation comparing our method to the baselines using 10 paired texture maps captured with both phones and in a studio setting. The evaluation metrics include mean PSNR, SSIM, LPIPS [21], and DISTS [9]. DISTS and LPIPS are perceptual metrics aligned with human judgment, while PSNR and SSIM are pixel-based metrics. As shown in Table 1, our method consistently outperforms the baseline across all metrics. The improvement is not limited to perceptual metrics but also extends to PSNR and SSIM, highlighting the efficacy of our proposed approach. It is worth noting that the relatively low PSNR and SSIM scores can be attributed to their sensitivity to small pixel shifts between the studio and phone-captured textures.

In Table 2, we display the face embedding distances, measured using Eq. (4), between avatars rendered from the phone-captured texture and the textures generated by various methods on the 31 unpaired phone captures. Our method exhibits the best preservation of facial identity.

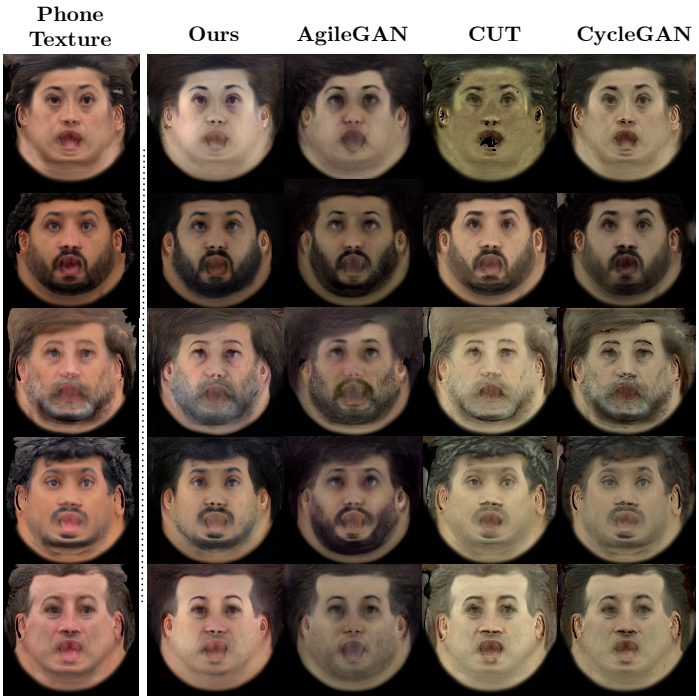

Fig. 4: Comparisons on Unpaired Phone-Captured Data: In comparison to prior work, our method excels in generating texture maps with superior preservation of identity, enhanced photorealism in facial details, more uniform illumination, and improved inpainting of missing regions.

Qualitative Results. In Fig 3, we provide some qualitative results on the paired test data. It is evident that our method generates a more plausible and photorealistic texture map compared to prior work, showcasing improvements in illumination transfer, facial details, and inpainting of missing regions.

In Fig 4, we present qualitative results comparing our method with prior work. While AgileGAN [38] successfully changes the illumination to be studio-like and inpaints missing regions, it introduces a significant identity shift (quantitatively measured in Table 2) and lacks facial details. We attribute this to the use of the Z^+ space [38], which may not be flexible enough for high-fidelity inversion, and the absence of identity-preserving constraints, both in architecture and optimization, during training. Due to its contrastive training paradigm, CUT [33] preserves identity better than AgileGAN but introduces significant artifacts. It is also unable to inpaint the missing regions around the ears and corners of the head. Similarly, like CUT [33], CycleGAN [42] also preserves identity better than AgileGAN [38] but struggles with inpainting missing regions. The textures also contain numerous uncanny artifacts that are uncharacteristic of human skin. In contrast, our method generates a high-quality studio-illuminated texture map with accurate facial details and inpainted missing regions.

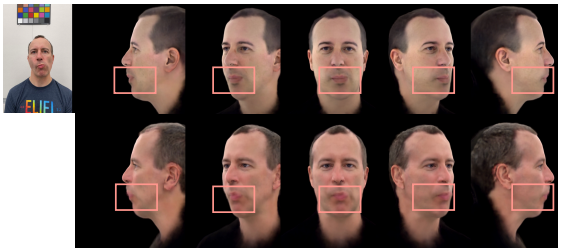

Fig. 5: Avatar Reanimation. The top row showcases a multiview render of an avatar generated using studio-like neutral and expression texture maps, created through our method as described in Sect 3.3. In contrast, the bottom row utilizes phone-captured texture maps. Notably, the lip region, highlighted with pink rectangles, appears more realistic and less blurry when using studio-like texture maps generated by our method.

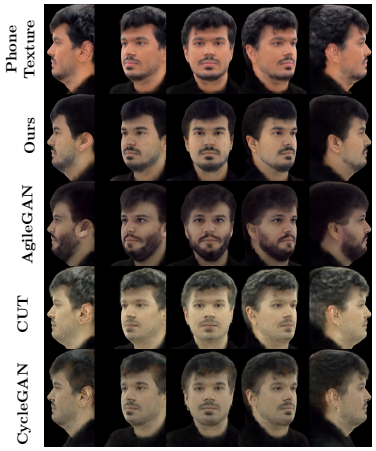

Fig. 6: Comparison of Avatars Generated Using [6]: In this comparison, we evaluate the quality of avatars generated by the Universal Prior Model (UPM) from AVA [6], using texture maps from various methods as input. Evidently, the avatars generated using texture maps from our method far exceed those generated by prior work.

Reanimation Results. As explained in Sect 3.3, we can transform the phone expression texture maps $\mathcal{I}_{exp}^{phone}$ to studio-like illumination using Eq. (15) and Eq. (16). We observe that utilizing studio-like illuminated expression texture maps ${\mathcal{I}^*}_{exp}$ results in a modest improvement in the quality of reanimated avatars using the universal prior model from AVA [6]. In Fig 5, we present an example where the top row shows the render of an avatar using studio-like neutral and expression textures, while the bottom row displays a render using phone-captured neutral and expression textures. It is evident that the lip region appears more realistic and less blurry when reanimated with the studio-like neutral and expression textures compared to the phone-captured textures. We recommend that readers refer to our supplementary video for a more comprehensive comparison.

4.1 Ablations

In this section, we conduct ablation studies to evaluate the various components of our method.

Details Conditioned Diffusion.

We ablate the necessity of using facial details extracted from the phone capture texture map, in the form of an image gradient, to synthesize accurate facial details using a diffusion model. We explore three scenarios:

1) A Diffusion model that does not use any conditioning on phone-captured texture gradient (Vanilla Diffusion); 2) A Diffusion model that uses the phone-captured texture gradient only during inference but **not** during training; 3) Our model, which incorporates the phone-captured texture gradient during both training and inference.

Table 3: Detail conditioning ablation. We calculate quantitative metrics on the 10 paired phone-cum-studio captured texture maps. It is evident that utilizing detail conditioning during both training and inference yields the best performance. Best and Second Best scores are highlighted.

Models	LPIPS ↓	DISTS ↓
Vanilla Diffusion	0.383	0.179
Inference-only Details conditioning	0.376	0.183
Training + Inference Details conditioning (Ours)	0.364	0.163

In Fig 7, we present the results of each of the three scenarios. While the Vanilla Diffusion generates a realistic-looking facial texture, it fails to preserve facial details present in the phone-captured texture map. Prominent moles, marked by the blue boxes in the phone-captured texture map, are not generated by the Vanilla Diffusion model. When conditioning the reverse process using the phone-texture gradient only during inference, we observe that the model misses some details and transfers shading from the phone-captured texture map to the studio-lit texture map (marked by red boxes in Fig 7), which is undesirable. Ideally, we want facial details to be preserved while eliminating illumination-dependent effects, such as strong shading and shadows, from the studio-lit texture map. Our method conditions the diffusion model on the phone-texture gradient during both inference and training. As shown in Fig 7, this allows the model to learn to retain facial details while ignoring illumination-induced effects, such as strong shading, shadows, and specularities when synthesizing facial details. In Table 3, we quantitatively validate each of the aforementioned diffusion model designs on paired data using LPIPS [21] and DISTS [9]. As can be seen, conditioning the diffusion model on phone-texture gradients both during training and inference gives the best results.

Table 4: W^+ vs. Z Space ablation. We utilize the 10 paired phone-cum-studio captured texture maps to compare the outcomes of training in the W^+ and Z spaces. The results clearly demonstrate that training in the W^+ space yields significantly better outcomes.

Models	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
Z Space	21.368	0.718	0.401	0.172
W^+ Space (Ours)	22.76	0.726	0.364	0.163

W^+ space vs. Z space for finetuning. We now evaluate the effectiveness of the W^+ space over the standard Z space for fine-tuning $\mathcal{G}_{Studio}$. For a quantitative comparison, we calculate the FaceID loss over the 31 unpaired evaluation texture maps and compute PSNR, SSIM, LPIPS [21], and DISTS [9] over the 10 paired studio-cum-phone texture maps. Qualitative results, along with the average FaceID distance over unpaired data, are shown in Fig 8, and the metrics on paired data are presented in Table 4. As evident, sampling in the W^+ space generalizes much better to unseen identities, exhibiting

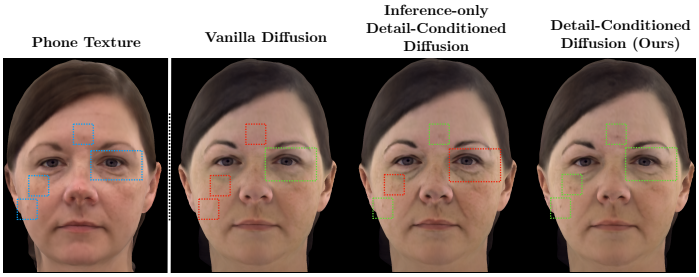

Fig. 7: Ablation on Detail-conditioned Diffusion. When no detail conditioning is applied, facial details visible in the phone capture are not reproduced. When detail conditioning is employed only during inference, undesirable shading effects are transferred to the generated texture map. Optimal results, with the most accurate and plausible reproduction of facial details, are achieved when detail conditioning is applied during both training and inference. Please refer to the text for further details.

Fig. 8: Qualitative W^+ space ablation. We observe that when $\mathcal{G}_{Studio}$ is finetuned using the Z space, it fails to generalize to novel subjects and exhibits significant artifacts. In contrast, training $\mathcal{G}_{Studio}$ in the W^+ lends it far better generalization.

significantly fewer artifacts and yielding superior results in terms of FaceID, PSNR, SSIM, LPIPS, and DISTS. We posit that this is because samples in the W^+ space (i.e., samples from S_{W^+} space) are near-perfect inversions of the training data, making them more diverse than those generated from the Gaussian-distributed Z space. Consequently, as seen in Fig 8, this leads to better generalization to unseen subjects. The model trained using the Z space exhibits uncanny artifacts in its results.

5 Conclusion and Limitations

In this paper, we present a method for generating studio-like, high-quality avatars from monocular phone captures, using a StyleGAN2-based image-to-image translation and diffusion-based image upsampling. Experiments show the effectiveness of our approach in manipulating lighting, inpainting missing parts and generating facial details thus enabling the creation of complete, studio-lit textures for rendering high-quality avatars. However, our method does have limitations. It struggles with input textures exhibiting extreme non-uniform lighting due to the constrained lighting conditions in our training data (we include examples in the supplementary). We also do not fine-tune the universal prior model to personalize the avatar based on phone capture data, as suggested in [6], thus the avatars lack personalized details for different facial expressions. Finally, our avatars are incomplete, featuring only the head. Future work involves extending the model to include the neck, shoulders, hands, and the entire body.

References

1. Alexander, O., Rogers, M., Lambeth, W., Chiang, J., Ma, W., Wang, C., Debevec, P.: The digital emily project: Achieving a photoreal digital actor. IEEE Computer Graphics and Applications **30** (2009) 3
2. Alexander, O., Fyffe, G., Busch, J., Yu, X., Ichikari, R., Jones, A., Debevec, P., Jimenez, J., Danvoye, E., Antionazzi, B., et al.: Digital ira: Creating a real-time photoreal digital actor. In: ACM SIGGRAPH 2013 Posters, pp. 1–1 (2013) 3
3. Athar, S., Xu, Z., Sunkavalli, K., Shechtman, E., Shu, Z.: Rignerf: Fully controllable neural 3d portraits. In: CVPR (June 2022) 2
4. Bi, S., Lombardi, S., Saito, S., Simon, T., Wei, S.E., Mcphail, K., Ramamoorthi, R., Sheikh, Y., Saragih, J.: Deep relightable appearance models for animatable faces. ACM Transactions on Graphics (TOG) **40**(4), 1–15 (2021) 4
5. Borshukov, G., Lewis, J.P.: Realistic human face rendering for" the matrix reloaded". In: ACM Siggraph 2005 Courses, pp. 13–es (2005) 3
6. Cao, C., Simon, T., Kim, J.K., Schwartz, G., Zollhoefer, M., Saito, S.S., Lombardi, S., Wei, S.E., Belko, D., Yu, S.I., Sheikh, Y., Saragih, J.: Authentic volumetric avatars from a phone scan. ACM Trans. Graph. (2022) 1, 2, 3, 4, 5, 9, 12, 14
7. Cao, C., Wu, H., Weng, Y., Shao, T., Zhou, K.: Real-time facial animation with image-based dynamic avatars. ACM Transactions on Graphics **35**(4) (2016) 4
8. Casas, D., Alexander, O., Feng, A.W., Fyffe, G., Ichikari, R., Debevec, P., Wang, R., Suma, E., Shapiro, A.: Rapid photorealistic blendshapes from commodity rgb-d sensors. In: Proceedings of the 19th Symposium on Interactive 3D Graphics and Games. pp. 134–134 (2015) 4
9. Ding, K., Ma, K., Wang, S., Simoncelli, E.P.: Image quality assessment: Unifying structure and texture similarity **44**(5), 2567–2581 (2022). https://doi.org/10.1109/TPAMI.2020.3045810 10, 13
10. Gafni, G., Thies, J., Zollhofer, M., Nießner, M.: Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8649–8658 (2021) 4
11. Garrido, P., Zollhöfer, M., Casas, D., Valgaerts, L., Varanasi, K., Pérez, P., Theobalt, C.: Reconstruction of personalized 3d face rigs from monocular video. ACM Transactions on Graphics (TOG) **35**(3), 1–15 (2016) 4
12. Ghosh, A., Fyffe, G., Tunwattanapong, B., Busch, J., Yu, X., Debevec, P.: Multiview face capture using polarized spherical gradient illumination. ACM Trans. Graph. (2011) 3
13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020) 4
14. Grassal, P.W., Prinzler, M., Leistner, T., Rother, C., Nießner, M., Thies, J.: Neural head avatars from monocular rgb videos. In: CVPR. pp. 18653–18664 (June 2022) 2, 4
15. Hu, L., Saito, S., Wei, L., Nagano, K., Seo, J., Fursund, J., Sadeghi, I., Sun, C., Chen, Y.C., Li, H.: Avatar digitization from a single image for real-time rendering. ACM Transactions on Graphics (ToG) **36**(6), 1–14 (2017) 4
16. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: Proceedings of the European conference on computer vision (ECCV). pp. 172–189 (2018) 4
17. Ichim, A.E., Bouaziz, S., Pauly, M.: Dynamic 3d avatar creation from hand-held video input. ACM Transactions on Graphics (ToG) **34**(4), 1–14 (2015) 4
18. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR (2017) 2, 4

19. Jimenez, J., Echevarria, J.I., Oat, C., Gutierrez, D.: Practical and realistic facial wrinkles animation. In: GPU Pro 360 Guide to Geometry Manipulation, pp. 95–107. AK Peters/CRC Press (2018) 3
20. Jimenez, J., Scully, T., Barbosa, N., Donner, C., Alvarez, X., Vieira, T., Matts, P., Orvalho, V., Gutierrez, D., Weyrich, T.: A practical appearance model for dynamic facial color. In: ACM SIGGRAPH Asia 2010 papers, pp. 1–10 (2010) 3
21. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV (2016) 10, 13
22. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. In: Proc. NeurIPS (2020) 6
23. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR (2019) 7
24. Lattas, A., Moschoglou, S., Gecer, B., Ploumpis, S., Triantafyllou, V., Ghosh, A., Zafeiriou, S.: Avatarme: Realistically renderable 3d facial reconstruction" in-the-wild". In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 760–769 (2020) 4
25. Lattas, A., Moschoglou, S., Ploumpis, S., Gecer, B., Ghosh, A., Zafeiriou, S.: Avatarme++: Facial shape and brdf inference with photorealistic rendering-aware gans. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(12), 9269–9284 (2021) 4
26. Lin, J., Yuan, Y., Zou, Z.: Meingame: Create a game character face from a single portrait. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 311–319 (2021) 4
27. Lombardi, S., Saragih, J., Simon, T., Sheikh, Y.: Deep appearance models for face rendering. ACM Transactions on Graphics (ToG) **37**(4), 1–13 (2018) 2, 3
28. Lombardi, S., Simon, T., Schwartz, G., Zollhoefer, M., Sheikh, Y., Saragih, J.: Mixture of volumetric primitives for efficient neural rendering. ACM Transactions on Graphics (ToG) **40**(4), 1–13 (2021) 2, 3
29. Luo, H., Nagano, K., Kung, H., Xu, Q., Wang, Z., Wei, L., Hu, L., Li, H.: Normalized avatar synthesis using stylegan and perceptual refinement. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 4
30. Ma, S., Simon, T., Saragih, J., Wang, D., Li, Y., De La Torre, F., Sheikh, Y.: Pixel codec avatars. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 64–73 (2021) 3
31. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for gans do actually converge? (2018) 7
32. Nagano, K., Fyffe, G., Alexander, O., Barbic, J., Li, H., Ghosh, A., Debevec, P.E.: Skin microstructure deformation with displacement map convolution. ACM Trans. Graph. **34**(4), 109–1 (2015) 3
33. Park, T., Efros, A.A., Zhang, R., Zhu, J.Y.: Contrastive learning for unpaired image-to-image translation. In: European Conference on Computer Vision (2020) 9, 10, 11
34. Pinkney, J.N., Adler, D.: Resolution dependent gan interpolation for controllable image synthesis between domains. arXiv preprint arXiv:2010.05334 (2020) 4
35. Sang, S., Zhi, T., Song, G., Liu, M., Lai, C., Liu, J., Wen, X., Davis, J., Luo, L.: Agileavatar: Stylized 3d avatar creation via cascaded domain bridging. In: SIGGRAPH Asia 2022 Conference Papers. pp. 1–8 (2022) 4
36. Seymour, M., Evans, C., Libreri, K.: Meet mike: epic avatars. In: ACM SIGGRAPH 2017 VR Village, pp. 1–2 (2017) 3
37. Shi, T., Yuan, Y., Fan, C., Zou, Z., Shi, Z., Liu, Y.: Face-to-parameter translation for game character auto-creation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 161–170 (2019) 4

38. Song, G., Luo, L., Liu, J., Ma, W.C., Lai, C., Zheng, C., Cham, T.J.: Agilegan: stylizing portraits by inversion-consistent transfer learning. ACM Transactions on Graphics (TOG) **40**(4), 1–13 (2021) 2, 4, 9, 10, 11
39. Wang, T.C., Liu, M.Y., Tao, A., Liu, G., Kautz, J., Catanzaro, B.: Few-shot video-to-video synthesis (2019) 4
40. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8798–8807 (2018) 4
41. Yamaguchi, S., Saito, S., Nagano, K., Zhao, Y., Chen, W., Olszewski, K., Morishima, S., Li, H.: High-fidelity facial reflectance and geometry inference from an unconstrained image. ACM Transactions on Graphics (TOG) **37**(4), 1–14 (2018) 4
42. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networkss. In: Computer Vision (ICCV), 2017 IEEE International Conference on (2017) 2, 4, 9, 10, 11
43. Zongsheng Yue, J.W., Loy, C.C.: Resshift: Efficient diffusion model for image super-resolution by residual shifting. In: Advances in Neural Information Processing Systems (NeurIPS) (2023) 7, 8