AgileAvatar: Stylized 3D Avatar Creation via Cascaded Domain Bridging

Shen Sang  
ByteDance  
Mountain View, USA  
shen.sang@bytedance.com

Tiancheng Zhi  
ByteDance  
Mountain View, USA  
tiancheng.zhi@bytedance.com

Guoxian Song  
ByteDance  
Mountain View, USA  
guoxiansong@bytedance.com

Minghao Liu  
UC Santa Cruz  
Santa Cruz, USA  
mliu40@ucsc.edu

Chunpon Lai  
ByteDance  
Mountain View, USA  
chunpon.lai@bytedance.com

Jing Liu  
ByteDance  
Mountain View, USA  
jing.liu@bytedance.com

Xiang Wen  
ByteDance  
Hangzhou, China  
sidao@bytedance.com

James Davis  
UC Santa Cruz  
Santa Cruz, USA  
davis@cs.ucsc.edu

Linjie Luo  
ByteDance  
Mountain View, USA  
linjie.luo@bytedance.com

Figure 1: (a) Given a front-facing user image as input, (b) our method progressively bridges the domain gap between real faces and 3D avatars through three stages: (b.1) The stylization stage performs an image space translation to generate a stylized portrait while normalizing expressions. (b.2) The parameterization stage uses a learned model to find avatar parameters which match the results of stylization. (b.3) The conversion stage searches for a valid avatar vector matching the parameterization that can be rendered by the graphics engine. (c) The output is a user editable 3D model which can be animated and applied to various applications, for example personalized emoji.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 ACM. ISBN 978-1-4503-9470-3/22/12. $15.00  
https://doi.org/10.1145/3550469.3555402

ABSTRACT

Stylized 3D avatars have become increasingly prominent in our modern life. Creating these avatars manually usually involves laborious selection and adjustment of continuous and discrete parameters and is time-consuming for average users. Self-supervised approaches to automatically create 3D avatars from user selfies promise high quality with little annotation cost but fall short in application to stylized avatars due to a large style domain gap. We propose a novel self-supervised learning framework to create high-quality stylized 3D avatars with a mix of continuous and discrete...
parameters. Our cascaded domain bridging framework first lever-
age a modified portrait stylization approach to translate input
selfies into stylized avatar renderings as the targets for desired 3D
avatars. Next, we find the best parameters of the avatars to match
the stylized avatar renderings through a differentiable imitator we
train to mimic the avatar graphics engine. To ensure we can ef-
effectively optimize the discrete parameters, we adopt a cascaded
relaxation-and-search pipeline. We use a human preference study
to evaluate how well our method preserves user identity compared
to previous work as well as manual creation. Our results achieve
much higher preference scores than previous work and close to
to those of manual creation. We also provide an ablation study to
justify the design choices in our pipeline.

CSCS CONCEPTS
• Computing methodologies → Non-photorealistic rendering.

KEYWORDS
Avatar Creation, Human Stylization

ACM Reference Format:
Shen Sang, Tiancheng Zhi, Guoxian Song, Minghao Liu, Chunpong Lai, Jing Liu, Xiang Wen, James Davis, and Linjie Luo. 2022. AgileAvatar: Stylized
3D Avatar Creation via Cascaded Domain Bridging. In SIGGRAPH Asia 2022
Conference Papers (SA ’22 Conference Papers), December 6–9, 2022, Daegu,
Republic of Korea. ACM, New York, NY, USA, 8 pages. https://doi.org/10.
1145/3550469.3555402

1 INTRODUCTION
An attractive and animatable 3D avatar is an important entry point
to the digital world that has become increasingly prominent in
modern life for socialization, shopping and gaming etc. A good
avatar should be both personalized (reflecting the person’s unique
appearance) and good-looking. Many popular avatar systems adopt
cartoonized and stylized designs for their playfulness and appeal-
ingness to the users such as Zepeto1 and ReadyPlayer2. However,
creating an avatar manually usually involves laborious selections and
adjustments from a swarm of art assets which is both time-
consuming and difficult for average users with no prior experience.

In this paper, we study automatic creation of stylized 3D avatars
from a single front-facing selfie image. To be specific, given a selfie
image, our algorithm predicts a strict avatar vector in which discrete parameters
are one-hot vectors.

A naive solution is to annotate a set of selfie images and train a
model to predict the avatar vector via supervised learning. How-
ever, large scale annotations are needed to handle a large range of
assets (usually in the hundreds). To alleviate the annotation cost,
self-supervised methods [Shi et al. 2019, 2020] are proposed to train
a differentiable imitator that mimics the renderings of the graphics
engine to automatically match the rendered avatar image with the
selfie image using various losses of identity and semantic segmenta-
tion. While these methods proved effective to create semi-realistic

1https://zepeto.me/
2https://readyplayer.me/

avatars close to user’s identity, they fall short in application to
stylized avatars since the style domain gap between selfie images
and stylized avatars are too large (see Fig. 7).

Our main technical challenges are two folds: (1) the large domain
gap between user selfie images and stylized avatars and (2) the com-
plicated optimization of a mix of continuous and discrete parameters
in the avatar vector. To address these challenges, we formulate a
cascaded framework which progressively bridge the domain gap
while ensuring optimization convergence on both continuous and
discrete parameters. Our novel framework consists of three stages:
Portrait Stylization, Self-supervised Avatar Parameterization, and
Avatar Vector Conversion. Fig. 1 shows the domain gap gradually
bridged across the three stages, while the identity information (hair
style, skin tone, glasses, etc.) is maintained throughout the pipeline.

First, the Portrait Stylization stage focuses on 2D real-to-stylized
visual appearance domain crossing. This stage translates input
selfie image to a stylized avatar rendering and remains in image
space. Naively applying existing stylization methods [Pinkney and
Adler 2020; Song et al. 2021] for translation will retain factors
such as expression, which would unnecessarily complicate later
stages of our pipeline. Thus, we create a modified variant from
AgileGAN [Song et al. 2021] to ensure uniformity in expression
while preserving user identity.

Next, the Self-Supervised Avatar Parameterization stage focuses
on crossing from image pixel domain to avatar vector domain. We
observed that strictly enforcing parameter discreteness causes opti-
mization to fail to converge. To address this, we use a relaxed for-
mulation called a relaxed avatar vector in which discrete parameters
are encoded as continuous one-hot vectors. To enable differentia-
bility in training, we trained an imitator in similar spirit to F2P [Shi
et al. 2019] to mimic the behavior of the non-differentiable engine.

Finally, the Avatar Vector Conversion stage focuses on domain
crossing from the relaxed avatar vector space to the strict avatar
vector space where all the discrete parameters are one-hot vectors.
The strict avatar vector can then be used by the graphics engine to
create final avatars and for rendering. We employ a novel search
process that leads to better results than direct quantization.

To evaluate our results, we use a human preference study to
evaluate how well our method preserves personal identity relative
to baseline methods including F2P [2019] as well as manual creation.
Our results achieve much higher scores than baseline methods and
close to those of manual creation. We also provide an ablation study to
justify the design choices in our pipeline.

In summary, our technical contributions are:

• A novel self-supervised learning framework to create high-
quality stylized 3D avatars with a mix of continuous and
discrete parameters;
• A novel approach to cross the large style domain gap in
stylized 3D avatar creation using portrait stylization;
• A cascaded relaxation and search pipeline that solves the
convergence issue in discrete avatar parameter optimization.

2 RELATED WORK
3D Face Reconstruction: Photorealistic 3D face reconstruction
from images has been studied extensively for many years. Ex-

SA ’22 Conference Papers, December 6–9, 2022, Daegu, Republic of Korea
Sang, S. et al
multiple cameras followed by a stereo or photogrammetry reconstruction [Beeler et al. 2010; Yang et al. 2020]. When only a single image is available, researchers leverage a parameterized 3D morphable model to reconstruct realistic 3D faces [Blanz and Vetter 1999; Chen and Kim 2021; Deng et al. 2019b; Peng et al. 2017; Xu et al. 2020]. Excellent surveys [Egger et al. 2020; Zollhöfer et al. 2018] exist providing great insights in this direction. These methods focus on an accurate reconstruction of the real human, and the model parameters often lack physical meaning. In contrast our work focuses on cross domain creation of a stylized avatar which has parameters with direct meaning to casual users.

3D Caricature: Non-photorealistic 3D face reconstruction has also received interest recently, a popular style being caricature. Qiu et al. [2021] created a dataset of 3D caricature models for reconstructing meshes from caricature images. Some works generate caricature meshes by exaggerating or deforming real face meshes, with [Cai et al. 2021; Wu et al. 2018] or without [Lewiner et al. 2011; Vieira et al. 2013] caricature image input. Sketches can be used to guide the creation [Han et al. 2017, 2018]. Recent works [Li et al. 2021; Ye et al. 2021] use GANs to generate 3D caricatures given real images. However, these methods are designed for reconstructing caricature meshes and/or textures while we focus on cartoonish avatars constrained by parameters with semantic meaning.

Game Avatars: Commercial products such as Zepeto and Ready-Player use a graphics engine to render cartoon avatars from user selfies. While no detailed description of their methods exists, we suspect these commercial methods are supervised with a large amount of manual annotations, something this paper seeks to avoid.

Creating semi-realistic 3D avatars has also been explored [Cao et al. 2016; Hu et al. 2017; Ichim et al. 2015; Luo et al. 2021]. Most relevant to our framework, Shi et al. [2019] proposed an algorithm to search for the optimal avatar parameters by comparing the input image directly to the rendered avatar. Follow-up work improves efficiency [Shi et al. 2020], and seeks to use the photograph’s texture to make the avatar match more closely [Lin et al. 2021]. These efforts seek to create a similar looking avatar, while this paper seeks to create a highly stylized avatar with a large domain gap.

Portrait Stylistization: Many methods for non-photorealistic stylization of 2D images exist. Gatys et al. [2016] proposed neural style transfer, matching features at different levels of CNNs. Image-to-image models focus on the translation of images from a source to target domain, either with paired data supervision [Isola et al. 2017] or without [Park et al. 2020; Zhu et al. 2017]. Recent development in GAN inversion [Richardson et al. 2021; Tov et al. 2021] and interpolation [Pinkney and Adler 2020] methods make it possible to achieve high quality cross-domain stylization [Cao et al. 2018; Song et al. 2021; Zhu et al. 2021]. The end result of these methods are in 2D pixels space and directly inspire the first stage of our pipeline.

3 PROPOSED APPROACH

Our cascaded avatar creation framework consists of three stages: Portrait Stylistization (Sec. 3.1), Self-supervised Avatar Parameterization (Sec. 3.2), and Avatar Vector Conversion (Sec. 3.3). A diagram of their relationship is shown in Fig. 2. Portrait Stylistization transforms a real user image into a stylized avatar image, keeping as much personal identity (glasses, hairs, colors, etc.) as possible, while simultaneously normalizing the face to look closer to an avatar rendering. Next, the Self-supervised Avatar Parameterization module regresses a relaxed avatar vector from the stylization latent code via a MLP based Mapper. Finally, the Avatar Vector Conversion module discretizes part of the relaxed avatar vector to meet the requirement of the graphics engine using an appearance-based search.

3.1 Portrait Stylistization

Portrait Stylistization transforms user images into stylized images close to our target domain. This stage of our pipeline occurs entirely within the 2D image domain. We adopt an encoder-decoder framework for the stylization task. A novel transfer learning approach is applied to a StyleGAN model [Karras et al. 2020], including W+ space transfer learning, using a normalized style exemplar set, and a loss function that supports these modifications.

W+ space transfer learning: We perform transfer learning directly from the W+ space, unlike previous methods [Gal et al. 2021; Song et al. 2021] where stylization transfer learning is done in the more entangled Z/Z+ space. The W+ space is more disentangled and can preserve more personal identity features. However, this design change introduces a challenge. We need to model a distribution prior W of the W+ space, as it is a highly irregular space [Wulff and Torralba 2020], and cannot be directly sampled like the Z/Z+ space (standard Gaussian distribution). We achieve this by inverting a large dataset of real face images into a W+ embeddings via a pre-trained image encoder [Tov et al. 2021], and then sample the latent codes from that prior. Fig. 3 provides one example of better preserved personalization. Notice that our method preserves glasses which are lost in the comparison method.

Normalized Style Exemplar Set: Our stylization method seeks to ignore pose and expression and produce a normalized image. In contrast, existing methods are optimized to preserve source to target similarities literally, transferring specific facial expressions, head poses, and lighting conditions directly from user photos into target stylized images. This is not desirable for our later avatar parameterization stage as we are trying to extract the core personal identity features only. In order to produce normalized stylizations
we limit the rendered exemplars provided during transfer learning to contain only neutral poses, expressions and illumination to ensure a good normalization. Fig. 3 provides an example of a smiling face. The comparison method preserves the smile, while our method successfully provides only the normalized core identity.

**Loss:** Our loss contains non-standard terms to support the needs of our pipeline. The target output stylization is not exactly aligned with the input due to pose normalization. Therefore, commonly used perceptual loss [Zhang et al. 2018] cannot be applied directly in decoder training. We instead use a novel segmented color loss.

The full objective comprises three loss terms to fine-tune the generator $G_{\phi}$. Let $G_{\phi_{w}}$ and $G_{\phi_{y}}$ be the model before and after fine-tuning. We introduce a color matching loss at a semantic level. Specifically, we leverage two face segmentation models from BiSeNet [Yu et al. 2018] pre-trained on real and stylized data separately to match the color of semantic regions. Let $\mathbb{S} = \{\text{hair, skin}\}$ be the classes taken into consideration, and $\mathcal{B}^k_\phi(i)$ ($k \in \mathbb{S}$) be the mean color of pixels belonging to class $k$ in image $i$. $\mathcal{B}^k_{\text{real}}$ and $\mathcal{B}^k_{\text{style}}$ represent real and stylized models separately. The semantic color matching loss is:

$$L_{\text{sem}} = \mathbb{E}_{w \sim \mathcal{W}} \left[ \sum_{k \in \mathbb{S}} \left( \left\| \mathcal{B}^k_{\text{real}}(G_{\phi_{w}}(w)) - \mathcal{B}^k_{\text{style}}(G_{\phi_{y}}(w)) \right\|_2^2 \right) \right]$$

An adversarial loss is used to match the distribution of the translated images to the target stylized set distribution $Y$, where $D$ is the StyleGAN2 discriminator [Karras et al. 2020].

$$L_{\text{ado}} = \mathbb{E}_{y \sim Y} \left[ \min(0, -1 + D(y)) \right] + \mathbb{E}_{w \sim \mathcal{W}} \left[ \min(0, -1 - D(G_{\phi_{y}}(w))) \right]$$

Also, to improve training stability and prevent artifacts, we use R1 regularization [Mescheder et al. 2018] for the discriminator:

$$L_{R1} = \frac{1}{2} \mathbb{E}_{y \sim Y} \left[ \| \nabla D(y) \|_2^2 \right]$$

Finally, the generator and discriminators are jointly trained to optimize the combined objective $\min_{\theta} \max_D L_{\text{st y l i z e}}$, where

$$L_{\text{st y l i z e}} = \lambda_{\text{ado}} L_{\text{ado}} + \lambda_{\text{sem}} L_{\text{sem}} + \lambda_{R1} L_{R1}$$

where $\lambda_{\text{ado}} = 1, \lambda_{\text{sem}} = 12, \lambda_{R1} = 5$ are constant weights set empirically. Please see supplementary for more details.

**Mapper Training:** The Mapper takes the results of portrait stylization as input and outputs an avatar vector which defines a similar looking avatar. Rather than using the stylized image itself as input, we use the latent code $w$ derived from the stylization encoder, since it is a more compact representation and contains facial semantic styles from coarse to fine [Karras et al. 2019].

The Mapper is built as an MLP, and trained using a Mapper Loss which measures the similarity between the stylized image, $I_{\text{style}}$, and the imitator output, $I_{\text{imitate}}$. This loss function contains several terms to measure the global and local similarity.

To preserve global appearance, we incorporate identity loss $L_{\text{id}}$ measuring the cosine similarity between two faces built upon a pretrained ArcFace [Deng et al. 2019a] face recognition network $R$: $L_{\text{id}} = 1 - \cos(R(I_{\text{style}}), R(I_{\text{imitate}})).$ For a more fine-grained similarity measurement, LPIPS loss [Zhang et al. 2018] is adopted:

$$L_{\text{LPIPS}} = \| F(I_{\text{style}}) - F(I_{\text{imitate}}) \|_2$$

where $F$ denotes the perceptual feature extractor. Additionally, we use a color matching loss to obtain more faithful colors for the skin and hair region:

$$L_{\text{color}} = \sum_{k \in \mathbb{S}} \left( \left\| \mathcal{B}^k_{\text{style}}(I_{\text{style}}) - \mathcal{B}^k_{\text{style}}(I_{\text{imitate}}) \right\|_2^2 \right)$$

The final loss function is:

$$L_{\text{mapper}} = \lambda_{\text{id}} L_{\text{id}} + \lambda_{\text{LPIPS}} L_{\text{LPIPS}} + \lambda_{\text{color}} L_{\text{color}}$$

where $\lambda_{\text{id}} = 0.4, \lambda_{\text{LPIPS}} = 0.8, \lambda_{\text{color}} = 0.8$ are set empirically.

We empirically choose the best loss terms to provide good results. An ablation study of these terms is provided in the results section.
while still discouraging mixtures of too many asset types. We found that quantization after optimization, which relaxes the constraint during training and re-apply it as postprocessing, is more effective for our task. Below we describe the relaxed optimization and in Sec. 3.3 we present the quantization method.

Our solution to training discrete parameters in the mapper makes use of the imitator’s interpolation property. When mixing two avatar vectors, the imitator still produces a valid rendering. That is, given the one-hot encoding \( v_1 \) and \( v_2 \) of two hair or beard types, their linear interpolation \( v_{\text{mix}} = (1 - \alpha) \cdot v_1 + \alpha \cdot v_2 (\alpha \in [0, 1]) \) produces a valid result. Please see supplementary for details.

Thus, when training the mapper we do not strictly enforce discrete parameters, and instead apply a softmax function to the final activation of the mapper to allow a continuous optimization space while still discouraging mixtures of too many asset types.

We compare our relaxed training with a strict training method performing quantization during optimization. In the forward pass, it quantizes the softmax result by picking the entry with maximum probability. In the backward pass, it back-propagates unaltered gradients in a straight-through way [Bengio et al. 2013]. In Fig. 4, our method produces a much closer match to the stylization results.

### 3.3 Avatar Vector Conversion

The graphics engine requires discrete inputs for attributes such as hair and glasses. However, the mapper module in Avatar Parameterization produces continuous values. One straightforward approach for discretization is to pick the type with the highest probability given the softmax result. However, we observe that this approach does not achieve good results, especially when dealing with multi-class attributes (e.g. 45 hair types). The challenge is that the solution space is under-constrained. Medium length hair can be achieved by selecting the medium length hair type, or by mixing between short and long hair types. In the latter case, simply selecting the highest probability of short or long hair is clearly not optimal.

We discretize the relaxed avatar vector via searching over all available discrete types. Notice in this example that the skin tone and hair type are much closer using our method.

**Differentiable Imitator:** The imitator is a neural renderer trained to replicate the output of the graphics engine as closely as possible given an input avatar vector. The imitator has the important property of differentiability, making it suitable for inclusion in an optimization framework. We leverage an existing neural model [Karras et al. 2019] as the backbone generator, which is capable of generating high quality avatar renderings. We train it with synthetic avatar data supervisedly. See supplementary for details.

**Discrete Parameters:** Solving for discrete parameters is challenging because of unstable convergence. Some methods handle this via quantization during optimization [Bengio et al. 2013; Cheng et al. 2018; Jang et al. 2016; Van Den Oord et al. 2017]. However, we found that quantization after optimization, which relaxes the discrete constraint during training and re-apply it as postprocessing, is more effective for our task. Below we describe the relaxed optimization and in Sec. 3.3 we present the quantization method.

Our solution to training discrete parameters in the mapper makes use of the imitator’s interpolation property. When mixing two avatar vectors, the imitator still produces a valid rendering. That is, given the one-hot encoding \( v_1 \) and \( v_2 \) of two hair or beard types, their linear interpolation \( v_{\text{mix}} = (1 - \alpha) \cdot v_1 + \alpha \cdot v_2 (\alpha \in [0, 1]) \) produces a valid result. Please see supplementary for details.

Thus, when training the mapper we do not strictly enforce discrete parameters, and instead apply a softmax function to the final activation of the mapper to allow a continuous optimization space while still discouraging mixtures of too many asset types.

We compare our relaxed training with a strict training method performing quantization during optimization. In the forward pass, it quantizes the softmax result by picking the entry with maximum probability. In the backward pass, it back-propagates unaltered gradients in a straight-through way [Bengio et al. 2013]. In Fig. 4, our method produces a much closer match to the stylization results.

**Figure 5:** Avatar Vector Conversion is necessary to convert the relaxed result produced during parameterization into discrete types suitable for the graphics engine. Direct classification often fails to select the best type. Our conversion selects the best match to the relaxed type by searching through all available discrete types. Notice in this example that the skin tone and hair type are much closer using our method.

**Figure 6:** Progressive domain crossing. (b) At the portrait stylization stage, the images may still contain characteristics outside the domain of a graphics avatar, such as hair shape and non-frontal pose. (c) At the parameterization stage, the images are within the target domain, but may contain mixtures of components. (d) Finally, after vector conversion the output is a strict avatar vector which can be rendered by the graphics engine. Using FID as a measure of image distribution similarity, notice that each step brings us closer to the final target avatar domain. ©Marcin Wichary and TechCrunch.

**4 EXPERIMENTAL ANALYSIS**

**Cascaded Domain Bridging:** To illustrate the effect of each stage in the proposed three-stage pipeline, the intermediate results are visualized in Fig. 6. Notice how the three stages progressively bridge the domain gap between real images and stylized avatars. To measure how close the intermediate results are in comparison to the target avatar domain, we use the perceptual metric FID [Kilgour et al. 2019]. Notice that the FID becomes lower after each stage, demonstrating the gradual reduction of domain gap.

**Visual Comparison with Baseline Methods:** We compare the proposed method against a number of baselines, shown in Fig. 7. CNN is a naive supervised method using rendered avatar images to train a CNN [Sandler et al. 2018] to fit ground truth parameters. The
CNN is then applied on the segmented head region of the input image. The domain gap causes the CNN to make poor predictions. Our stylization + CNN narrows the domain gap by applying the CNN to our stylized results. This noticeably improves predictions, however errors in hair and face coloration remain. Since the CNN is only trained on synthetic data, it cannot regress the parameters properly due to the domain gap between training and test data even for stylized images. F2P [2019] is a self-supervised optimization-based method designed for semi-realistic avatars. This method fails to do well, likely because it naively aligns the segmentation of real faces and the avatar faces, without considering the domain gap. Manual results were created by expert-trained users. Given a real face, the users were asked to build an avatar that preserves personal identity while demonstrating high attractiveness based on their own judgement. Visually, our method shows a quality similar to manual creation, demonstrating the utility of our method.

Numerical Comparison with Baseline Methods: To evaluate results numerically we rely on judgements made by human observers recruited through Amazon Mechanical Turk. We conduct two user studies for quantitative evaluation: Attribute Evaluation and Matching. We perform attribute evaluation to evaluate whether users believe that specific identity attributes such as hair color and style match the source photograph using a yes/no selection. 330 opinions were collected for each of 6 attributes. Table 1 shows results, indicating that our method retains photograph identity better than the baseline. In the matching task, we evaluate whether an avatar retains personal identity overall. Four random and diverse images were used to create avatars, and the subject must choose which is the correct match to a specific photograph. A total of 990 judgements were collected. Avatars created with our method were identified correctly significantly more often than baseline methods, approaching the level of manually created avatars.

Figure 7: Results comparison. (a) Given an input image, (b) our method produces an avatar in the target cartoon style that looks similar to the user. (c) A CNN trained on synthetic data produces incorrect beard, hair style, and glasses on real image inputs due to the significant domain gap. (d) Applying the CNN instead to the results of stylization reduces the domain gap and thus improves results, however significant errors remain. (e) F2P, a baseline method intended to produce semi-realistic avatars does not consider the domain gap and thus produces poor results when used with stylized avatars [Shi et al. 2019]. (f) Manual results were created by expert-trained users. Our results approximate the quality obtainable through manual creation. ©Sebastiaan ter Burg, NIGP, YayA Lee and S Pakhrin.

Numerical Comparison with Baseline Methods: To evaluate results numerically we rely on judgements made by human observers recruited through Amazon Mechanical Turk. We conduct two user studies for quantitative evaluation: Attribute Evaluation and Matching. We perform attribute evaluation to evaluate whether users believe that specific identity attributes such as hair color and style match the source photograph using a yes/no selection. 330 opinions were collected for each of 6 attributes. Table 1 shows results, indicating that our method retains photograph identity better than the baseline. In the matching task, we evaluate whether an avatar retains personal identity overall. Four random and diverse images were used to create avatars, and the subject must choose which is the correct match to a specific photograph. A total of 990 judgements were collected. Avatars created with our method were identified correctly significantly more often than baseline methods, approaching the level of manually created avatars.

3https://www.mturk.com/
Table 1: Numerical results from two user studies. Our method is judged to produce better avatars than the baseline methods, approaching the quality of manual work. Attribute evaluation: judge whether a specific attribute of the created avatar matches the human image. Matching: choose the correct one out of four avatars which matches the human image.

| Attribute Evaluation | Match Task |
|----------------------|------------|
| beard shape          | 0.36 0.46 0.22 0.21 |
| face type            | 0.12 0.12 0.12 0.12 |
| brow type            | 0.36 0.36 0.36 0.36 |
| hair type            | 0.67 0.67 0.67 0.67 |
| hair color           | 0.57 0.57 0.57 0.57 |
| hair style           | 0.43 0.43 0.43 0.43 |
| skin tone            | 0.66 0.66 0.66 0.66 |
|                      | 0.82 0.82 0.82 0.82 |
|                      | 0.92 0.92 0.92 0.92 |

Table 2: Ablation study for mapper training losses. Users picked the best matching avatar from the six candidates produced by loss combinations. The scores show the fraction of each combination picked. $L_{\text{ID}}$ is the most significant component, while $L_{\text{ID}}$ and $L_{\text{Color}}$ also improve the results.

| ID          | LPISP | ID+LPISP | ID+Color | LPISP+Color | ID+LPISP+Color |
|-------------|-------|----------|----------|-------------|----------------|
| Manual      | 0.94  | 0.97     | 0.85     | 0.90        | 0.86           |
| F2P [2019]  | 0.36  | 0.46     | 0.22     | 0.21        | 0.12           |
| CNN         | 0.17  | 0.54     | 0.22     | 0.46        | 0.30           |
| Stylization+CNN | 0.45 | 0.69     | 0.38     | 0.57        | 0.43           |
| Ours        | 0.82  | 0.94     | 0.88     | 0.82        | 0.72           |

5 LIMITATIONS

We observe two main limitations to our method. First, our method occasionally produces wrong predictions on assets covering a small area, because their contribution to the loss is small and gets ignored. The eye color in Fig. 9 (a) is an example of this difficulty. Redesigning the loss function might resolve this problem. Second, lighting is not fully normalized in the stylization stage, leading to incorrect skin tone estimates when there are strong shadows, shown in Fig. 9 (b). This problem could potentially be addressed by incorporating intrinsic decomposition into the pipeline. In addition to the limitations of our method, we experience a loss of ethnicity in the final results, which is mainly introduced by the graphics engine, as also evidenced by the manually-created results. This issue could be addressed by improving the diversity of the avatar system.

6 CONCLUSION

In summary, we present a self-supervised stylized avatar auto-creation method with cascaded domain crossing. Our method demonstrates that the gap between the real images domain and the target avatar domain can be progressively bridged with a three-stage pipeline: portrait stylization, self-supervised avatar parameterization, and avatar vector conversion. Each stage is carefully designed and cannot be simply removed. Experimental results show that our approach produces high quality attractive 3D avatars with personal identities preserved. In the future, we will extend the proposed pipeline to other domains, such as cubism and caricature avatars.
