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

Chunpeng Lai  
ByteDance  
Mountain View, USA  
chunpeng.lai@bytedance.com

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

Xiang Wen  
ByteDance  
Hangzhzhou, 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.

© H JACQUOT and Montclair Film.

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 leverages 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 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 those of manual creation. We also provide an ablation study to justify the design choices in our pipeline.

CCS CONCEPTS
• Computing methodologies → Non-photorealistic rendering

KEYWORDS
Avatar Creation, Human Stylization

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 appealingness to the users such as Zepe1 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 an avatar vector as the complete configuration for a graphics engine to generate a 3D avatar and render avatar images from predefined 3D assets. The avatar vector consists of parameters specific to the predefined assets which can be either continuous (e.g., head length) or discrete (e.g., hair types).

A naive solution is to annotate a set of selfie images and train a model to predict the avatar vector via supervised learning. However, 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 segmentation. While these methods proved effective to create semi-realistic 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 complex 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 optimization to fail to converge. To address this, we use a relaxed formulation called a relaxed avatar vector in which discrete parameters are encoded as continuous one-hot vectors. To enable differentiability 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. Extremely high quality models can be obtained using gantries with multiple cameras followed by a stereo or photogrammetry reconstruction [Beeler et al. 2018; 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. 2019; 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,
Figure 2: Pipeline. Our framework consists of three modules: Portrait Stylization for image-space real-to-stylized domain crossing, Self-supervised Avatar Parameterization for recovering relaxed avatar vector from the stylization latent code, and Avatar Vector Conversion for discretizing the predicted relaxed avatar vector into a strict avatar vector that can be taken by the graphics engine directly. ©NGAO STUDIO.

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 Stylization: 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 Stylization (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 Stylization 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 Stylization
Our stylization method seeks to achieve a similar looking avatar, while this paper seeks to create a highly stylized avatar with a large domain gap. 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 $\mathcal{W}$ 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.

Figure 3: Portrait stylization results. Compared with a state-of-the-art stylization method, AgileGAN [Song et al. 2021], our stylization does a better job at preserving the user’s personal identity (e.g. glasses are preserved), and simultaneously normalizing the expressions (e.g. mouth is closed) for easier fitting in the downstream pipeline. ©Sebustiaan ter Burg.
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 $G$. Let $G_{\phi}$ and $G_{\hat{\phi}}$ 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 $S = \{\text{hair, skin}\}$ be the classes taken into consideration, and $B^k(i) \ (k \in S)$ be the mean color of pixels belonging to class $k$ in image $i$. $B^k_{\text{real}}$ and $B^k_{\text{style}}$ represent real and stylized models separately. The semantic color matching loss is:

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

(1)

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]:

$$L_{\text{adv}} = \mathbb{E}_{y \sim Y}[\min(0,-1+D(y))] + \mathbb{E}_{w \sim W}[\min(0,-1-D(G_{\phi}(w)))]$$

(2)

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

$$L_{R1} = \frac{\lambda_{\text{R1}}}{2} \mathbb{E}_{y \sim Y}[\| \nabla D(y) \|^2], \text{where we set } \lambda_{\text{R1}} = 10 \text{ empirically.}$$

Finally, the generator and discriminators are jointly trained to optimize the combined objective $\min_{\phi} \max_{\theta} L_{\text{style}}$, where

$$L_{\text{style}} = \lambda_{\text{adv}} L_{\text{adv}} + \lambda_{\text{sem}} L_{\text{sem}} + \lambda_{\text{R1}} L_{\text{R1}}$$

(3)

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

3.2 Self-supervised Avatar Parameterization

Avatar Parameterization finds a set of parameters for the rendering engine which produces an avatar matching the stylized portrait as closely as possible. We call the module which finds parameters the mapper. To facilitate training the mapper, we use a differentiable neural rendering engine we call the imitator.

A particular avatar is defined by an avatar vector with both continuous and discrete parameters. Continuous parameters are used to control primarily placement and size, for example eye size, eye rotation, mouth position, and head width. Discrete parameters are used to set individual assets and textures such as hair types, beard types, and skin tone textures. All parameters are concatenated into a vector with discrete parameters represented as one-hot vectors.

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

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

(4)

The final loss function is:

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

(5)

where $\lambda_{\text{id}} = 0.4$, $\lambda_{\text{pips}} = 0.8$, $\lambda_{\text{color}} = 0.8$ 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.

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 \(v_{mix} = (1 - \alpha) \cdot v_1 + \alpha \cdot v_2\ (\alpha \in [0, 1])\) 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 candidates from the asset list for each attribute, while fixing all other parameters. Using the image result from the imitator \(\hat{I}_{imitate}\) as target, we use the loss function from Eq. 5 as an objective to measure the similarity between \(\hat{I}_{imitate}\) and the candidate result \(\hat{I}_{cond}\). By minimizing the objective, we can find the best solution for each attribute. The selections for each attribute are combined to create the avatar vector used for graphics rendering and animation.

Fig. 5 provides a comparison of direct classification and our method. Note that direct classification makes incorrect choices for hair type and skin color while ours closely matches the reference image.

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.
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.

Numerical Comparison with Baseline Methods: To evaluate results numerically we rely on judgements made by human observers recruited through Amazon Mechanical Turk 3. 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.

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Attribute Evaluation</th>
<th>Match Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>0.82 0.94 0.88 0.72</td>
<td>0.82 0.92</td>
</tr>
<tr>
<td>CNN</td>
<td>0.45 0.69 0.38 0.57</td>
<td>0.66 0.82</td>
</tr>
<tr>
<td>Stylization+CNN</td>
<td>0.45 0.69 0.38 0.57</td>
<td>0.66 0.82</td>
</tr>
<tr>
<td>F2P [2019]</td>
<td>0.36 0.46 0.22 0.12</td>
<td>0.36 0.67</td>
</tr>
<tr>
<td>Manual</td>
<td>0.90 0.90 0.86 0.94</td>
<td>0.90 0.96</td>
</tr>
</tbody>
</table>

3https://www.mturk.com/
We observe two main limitations to our method. First, our method cannot be simply removed. Experimental results show that our

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. $\mathcal{L}_{\text{ID}}$ is the most significant component, while $\mathcal{L}_{\text{ID+LPIPS+Color}}$ also improve the results.

<table>
<thead>
<tr>
<th></th>
<th>ID</th>
<th>LPIPS</th>
<th>ID+LPIPS</th>
<th>ID+Color</th>
<th>LPIPS+Color</th>
<th>ID+LPIPS+Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.3%</td>
<td>17.8%</td>
<td>18.7%</td>
<td>14.8%</td>
<td>19.1%</td>
<td>20.3%</td>
<td></td>
</tr>
</tbody>
</table>

Portrait Stylization Ablation: To study the impact of Portrait Stylization on the complete avatar creation pipeline we compare three options, shown in Fig. 8. No stylization removes this stage entirely and uses the real image as input to parameterization loss calculation. Without stylization, the parameterization module tries to match the real image with the target stylized avatar, leading to poor visual quality. AgileGAN [Song et al. 2021] is a state-of-the-art stylization method. It provides stylization and thus improves the final avatar attractiveness compared to no stylization. However, it cannot remove the impact of expressions and does not handle glasses well. In Row 1 (b), the smile expression is explained as a big mouth in the fitting stage, and personal information like glasses is not preserved in Row 2 (b). Our method addresses these issues and achieves better results in both visual quality and personal identity.

Mapper Losses Ablation: To study the importance of including all losses while training the Mapper, we generate results using different permutations of loss terms (identity, LPIPS, color). We then collected 990 user judgements from Amazon Mechanical Turk, to select the best matching results to the input image among six permutation results. Table 2 shows the fraction of each option selected. The full set of losses achieves the best score by a small margin, matching our observations that the overall method is robust to the precise selection of loss, but that the additional terms help in some cases.

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

Figure 8: We ablate by removing the stylization stage, as well as replacing our stylization with a state-of-the-art method. In each case the final renderings from the graphics engine are shown. (a) Fitting directly on a user image results in an avatar that lacks attractiveness. (b) Replacing our stylization with AgileGAN [2021] suffers from missing personal information such as glasses and artifacts where smiles are misinterpreted as heavy lips or mustache. (c) Our stylization retains personal features like glasses, and generate visually appealing results in spite of expressions. ©Chang-Ching Su and Luca Boldrini.

Figure 9: Limitations: (a) failure on a parameter (eye color) affecting a small number of pixels. (b) incorrect skin tone prediction caused by shadows. ©Daniel Åberg and Peter Bright.

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.

REFERENCES


