Supplementary material for "Unsupervised Creation of Parameterized Avatars"

1 Summary of Notations

Tab. 1 itemizes the symbols used in the submission. Fig. 2,3,4 of the main text illustrate many of these symbols.

2 DANN results

Fig. 1 shows side by side samples of the original image and the emoji generated by the method of [1]. As can be seen, these results do not preserve the identity very well, despite considerable effort invested in finding suitable architectures.

3 Multiple Images Per Person

Following [4], we evaluate the visual quality that is obtained per person and not just per image, by testing TOS on the Facescrub dataset [3]. For each person p, we considered the set of their images X_p, and selected the emoji that was most similar to their source image, i.e., the one for which:

$$\underset{x \in X_p}{\operatorname{argmin}} ||f(x) - f(e(c(G(x))))||. \tag{1}$$

Fig. 2 depicts the results obtained by this selection method on sample images form the Facescrub dataset (it is an extension of Fig. 7 of the main text). The figure also shows, for comparison, the DTN [4] result for the same image.

4 Detailed Architecture of the Various Networks

In this section we describe the architectures of the networks used in for the emoji and avatar experiments.

4.1 TOS

Network g maps DeepFace's 256-dimensional representation [5] into 64×64 RGB emoji images. Following [4], this is done through a network with 9 blocks, each consisting of a convolution, batch-normalization and ReLU, except the last layer which employs Tanh activation. The odd blocks 1,3,5,7,9 perform upscaling convolutions with 512-256-128-64-3 filters respectively of spatial size 4×4. The even ones perform 1×1 convolutions [2]. The odd blocks use a stride of 2 and padding of 1, excluding the first one which does not use stride or padding.

Network e maps emoji parameterization into the matching 64×64 RGB emoji. The parameterization is given as binary vectors in $\mathbb{R}^{813}$ for emojis; Avatar parameterization is in $\mathbb{R}^{354}$. While there are dependencies among the various dimensions (an emoji cannot have two hairstyles at once), the binary representation is chosen for its simplicity and generality. e is trained in a fully supervised way, using pairs of matching parameterization vectors and images in a supervised manner.

The architecture of e employs five upscaling convolutions with 512-256-128-64-3 filters respectively, each of spatial size 4×4. All layers except the last one are batch normalized followed by a ReLU activation. The last layer is followed by Tanh activation, generating an RGB image with values in range $[-1, 1]$. All the layers use a stride of 2 and padding of 1, excluding the first one which does not use stride or padding.

Symbol	Meaning	Line
$\mathcal{X}$	Input space	214
$\mathcal{Y}$	Output space	214
$\mathcal{Y}_1, \mathcal{Y}_2$	Tied output spaces	345
D_S	Source distribution	228
D_T	Target distribution	230
D_1,D_2	Input/output or other pairs of distributions	246 or 335
ℓ	A loss function, typically $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$	230
$\text{disc}_{\mathcal{C}}(D_1, D_2)$	Discrepancy between two distributions D_1 and D_2, using functions from the hypothesis class $\mathcal{C}$	161
$R_D[c_1, c_2]$	The risk between two functions c_1 and c_2, i.e.,$\mathbb{E}_{x\sim D}\left[\ell(c_1(x), c_2(x))\right]$	233
y	The function from input to output we learn	248
y_S, y_T	In domain adaptation, the source and target functions from input to output	228
f	A pre-trained feature map	252
e	A mapping from configurations ($\mathcal{Y}_2$) to parameterized outputs ($\mathcal{Y}_1$)	237
$c \in \mathcal{H}_3$	A learned mapping from the space of parameterized outputs ($\mathcal{Y}_1$) to configurations ($\mathcal{Y}_2$)	243 and 461
$g \in \mathcal{H}_2$	A generator function from the feature space (image of f) to the output space	251
d	The discriminator of the GAN	444
$h \in \mathcal{H}$	A mapping from input to output (different to each problem)	231,251, or 342
G	A generator from the input space to the space $\mathcal{Y}_1$	422
L_{GAN}	The GAN loss term. Used together with Eq. 7	444
L_c	The loss that arises from the mismatch between G and c. The specific form used is given in Eq. 5	457
ℓ_e	For a given x, the mismatch between $G(x)$ and $e \circ c$ applied to it	464
$\mathbf{s}$	The training set in $\mathcal{X}$	472
$\mathbf{t}$	The training set in $\mathcal{Y}_1$	473
L_{CONST}	The term that enforces f-constancy	481
L_{TID}	The term that enforces idempotency for the learned mapping	524
L_{TV}	Total Variation loss, which encourages smoothness of the output	534
α,β,γ,δ	Tradeoff parameters (weights in the loss term the network minimizes)	530
p	The feature map of the DANN algorithm [1]	399
l	The label predictor network of the DANN algorithm [1]	725

Table 1: The mathematical notations used in the paper.

Network d takes 152×152 RGB images (either natural or scaled-up emoji) and outputs log-probabilities predicting if the image is fake or real. It consists of 6 blocks, each containing a convolution with stride 2, batch normalization, and a leaky ReLU with leakiness coefficient of 0.2. Each block contains 64-128-256-512-512-3 filters respectively. As before, the last layer does not employ batch normalization and ReLU.

Network c maps a 64×64 emoji to parameterization vector. It contains five convolutional layers, each followed by batch normalization and a leaky ReLU with a leakiness coefficient of 0.2. Each layer contains 64-128-256-512-813 filters respectively. The last layer is followed by Tanh activation, generating a parameterization vector with values in range $[-1, 1]$.

The networks used for the synthetic polygon experiment are somewhat simpler: g has the same structure of as in the emoji experiment excluding the even convolutions i.e., it does not contain the 1×1 convolutions. The architecture of d is unchanged. Finally, the architectures of e and c are updated to match the synthetic experiment parameterization. e is changed to map a parameterization vectors in $\mathbb{R}^3$ to RGB images, and c is trained to predict such a vector.

4.2 DANN

In the domain adaptation experiments, network p extracts 2048-dimensional feature vectors from 64×64 RGB images. It resembles the structure of network c - with 4 convolution layers. Each convolution is with 64-128-256-512 filters respectively. The last convolutional layer employs a stride of 1 instead of 2 and does not use batch-normalized or leaky ReLU. Finally, the network output is flattened to 1-dimensional feature vector.

The label prediction network l accepts as input feature vectors generated by p and outputs emoji param-

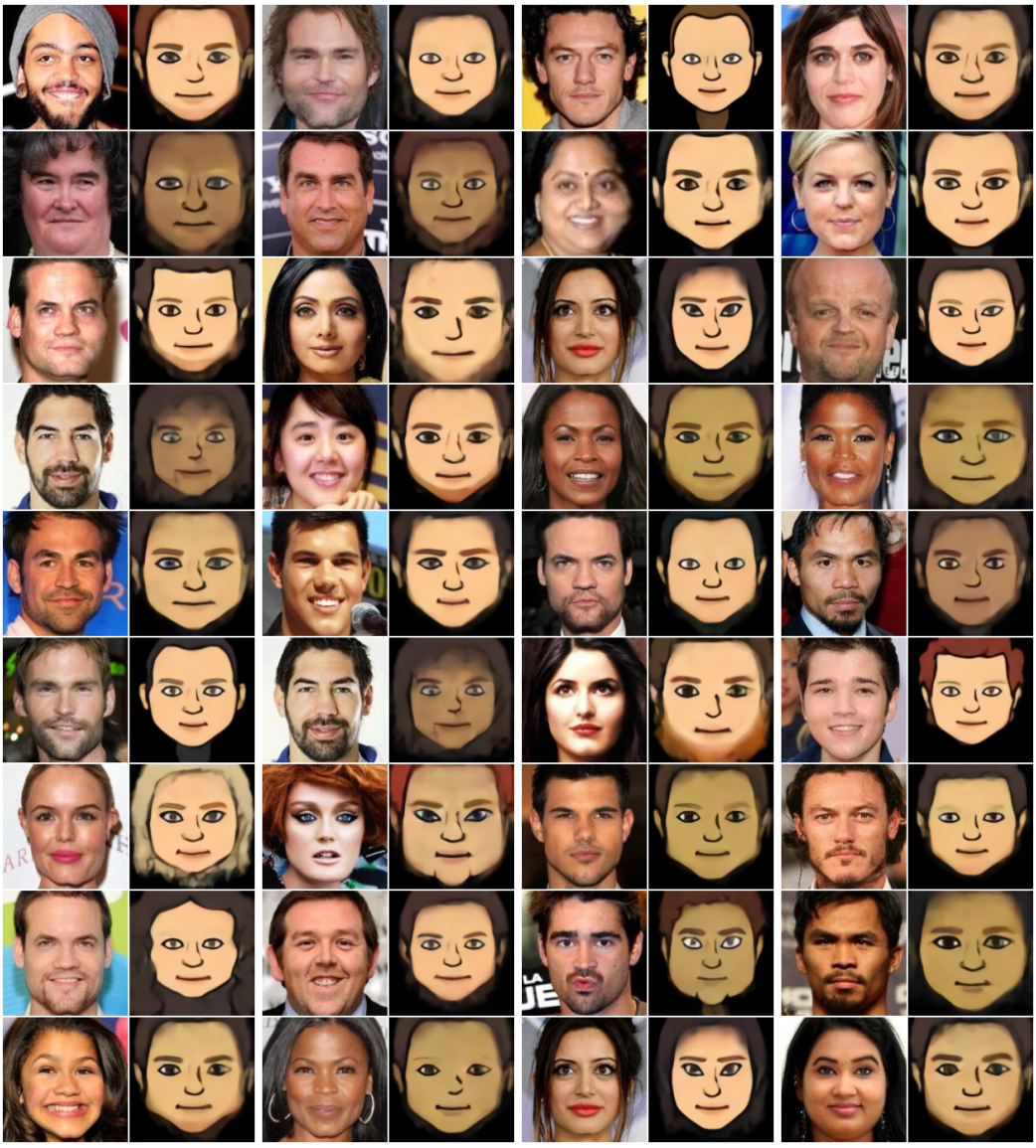

Figure 1: Shown, side by side are sample images from the CelebA dataset and the results obtained by the DANN domain adaptation method [1]. These results are not competitive.

eterization vectors matching the input image. It consists of 3 fully connected layers. Each hidden layer is followed by batch-normalization and leaky ReLU activation. The last layer is followed by Tanh activation. The hidden layers contain 1024 and 512 units respectively.

The discriminator d predicts the input image domain given its feature vector. It consists of two fully connected layers with 512 hidden units. The hidden layer is followed by batch normalization and leaky ReLU activations. It is preceded by a gradient reversal layer to ensure that the feature distributions of both domains are similar. The last layer is followed by Sigmoid activation, predicting the input image domain.

References

[1] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, 2016.

[2] M. Lin, Q. Chen, and S. Yan. Network In Network. *ArXiv e-prints: 1312.4400*, 2013.

[3] H. Ng and S. Winkler. A data-driven approach to cleaning large face datasets. In *ICIP*, 2014.

[4] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[5] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

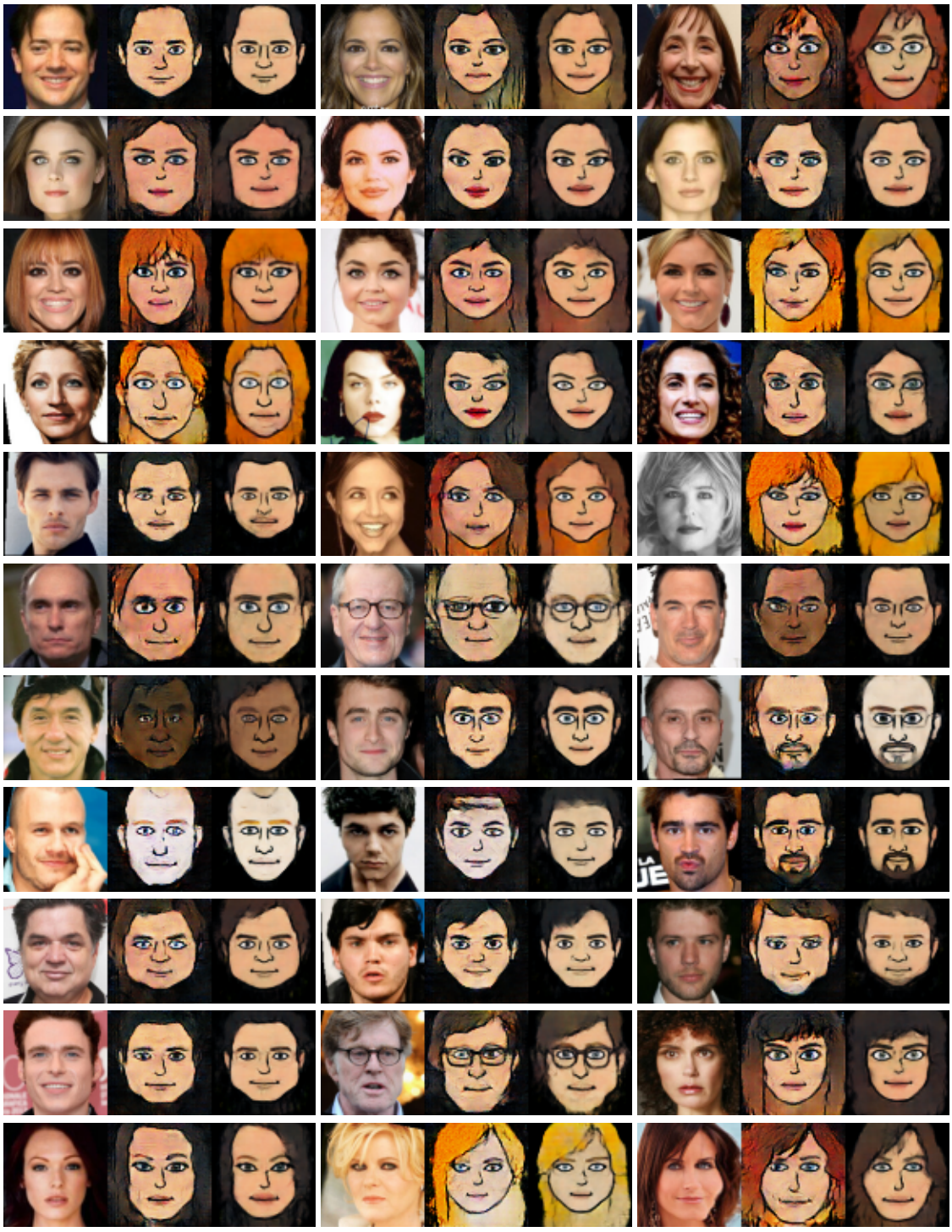

Figure 2: The results obtained by the TOS method for a sample of individuals from the Facescrub dataset. Shown, side by side, are the image used to create the TOS and the DTN emoji, the DTN emoji, and the TOS emoji, obtained by $e \circ c \circ g \circ f$. The image that represents a person maximizes, out of all images for this person, f-constancy for the TOS method.