Self-supervised Deformation Modeling for Facial Expression Editing

ShahRukh Athar, Zhixin Shu, and Dimitris Samaras
Stony Brook University
New York, 11794, U.S.A.
{sathar,zhshu,samaras}@cs.stonybrook.edu

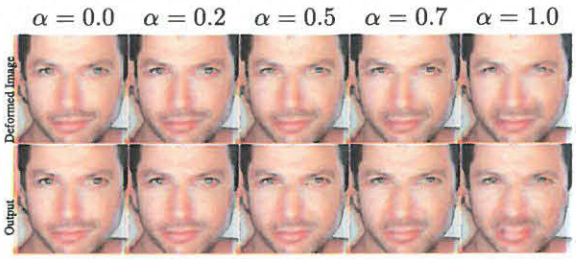
Abstract

Recent advances in deep generative models have demonstrated impressive results in photo-realistic facial image synthesis and editing. Facial expressions are inherently the result of muscle movement. However, existing neural network-based approaches usually only rely on texture generation to edit expressions and largely neglect the motion information. In this work, we propose a novel end-to-end network that disentangles the task of facial editing into two steps: a "motion-editing" step and a "texture-editing" step. In the "motion-editing" step, we explicitly model facial movement through image deformation, warping the image into the desired expression. In the "texture-editing" step, we generate necessary textures, such as teeth and shading effects, for a photo-realistic result. Our physically-based task-disentanglement system design allows each step to learn a focused task, removing the need of generating texture to hallucinate motion. Our system is trained in a self-supervised manner, requiring no ground truth deformation annotation. Using Action Units [8] as the representation for facial expression, our method improves the state-of-the-art facial expression editing performance in both qualitative and quantitative evaluations.

1. Introduction

Editing facial expressions of faces found 'in-the-wild' is a problem of great interest within the Computer Vision and the Computer Graphics communities with a wide variety of applications in industries ranging from cinema to photography to e-commerce. An ideal facial expression editing system would allow its user to seamlessly change the expression of a given input face without affecting invariant attributes such as facial identity, age, etc. Recent advances in computer vision, driven by improvements in the adversarial learning framework [11] have now made it possible to successfully perform such expression edits in many cases [25, 6].

It is well known that facial expressions result from complex and constrained movements of facial muscles in three dimensions thus making the task of modelling and encoding them in two dimensional pixel space rather challenging. In 1978, Eckman and Friesen [8] developed the Facial Action Coding System (FACS) that 'can be used to describe any facial movement (observed in photographs, motion picture film or video tape) in terms of anatomically based action units'. Each Action Unit (AU) corresponds to a change in some specific region of the face and any anatomically possi-



(a) Expression Editing using DefGAN

(b) Expression Editing using GANimation [25]

Figure 1: Expression Editing using DefGAN. The image on the top shows an input face image being edited to the target expression 'disgust', where α controls the activation of the target AUs [8]. DefGAN edits the image in two phases. First, in the 'Motion editing' phase, we deform the input image to conform to the target expression as can be seen in the top row of Fig. 1a. Next, in the 'Texture editing' phase we hallucinate the necessary textures to give us the final image (bottom row Fig. 1a). As can be seen, the deformation models facial movements and performs most of the editing work. In contrast, as shown in Fig. 1b, GANimation [25] edits the image entirely using texture hallucinations leading to artifacts in the final image (bottom right image of Fig. 1b)

ble expression can be represented as a vector of AU intensities and/or detections. For example, a smile (corresponding to happiness) can be encoded by the intensity of AU6 and AU12 and sadness can be encoded by the intensity of AU1, AU4 and AU15. Due to their interpretability and universality AUs are an ideal representation for editing expressions. Ground truth AU annotation of images is generally done by trained experts and is rather time consuming. Over the years however, a number of learning-based methods have been developed that are able to predict the AU activations of a face in any given image with low error rates, thus making the AU annotation of large scale datasets feasible. Recent work, such as GANimation [25] relies on AUs as a weak supervision signal, to learn a facial expression model that allows one to seamlessly transition between expressions by interpolating in the AU space. Despite its use of AUs as a supervision signal, GANimation [25] does not explicitly model facial movement and instead relies on texture synthesis to mimic facial movement effects. A downside of this is that there can be significant artifacts in the editing results such as the disappearing eyebrows and changes in the beard that can be seen in Fig. 1 .

In this work, we propose DefGAN, a method that separates the task of editing expressions into two sequential phases - a 'Motion Editing' phase which models facial movements as an image deformation followed by a 'Texture Editing' phase to hallucinate final details that arise from the appearance or disappearance of texture (such as teeth). In the 'Motion Editing' phase, a Convolutional Neural Network (CNN) models facial movement explicitly by predicting a deformation field that deforms the input face to better conform to the target expression (for example curving the region around the mouth into a crescent to generate a smile). We leverage recent work from [26] and model facial movements using an offset based deformation field. In the 'Texture Editing' phase, we hallucinate the necessary textures (such as teeth, shadows) using another CNN on top of the deformed image which completes the editing process and gives us the final edited image.

We weakly supervise our networks using available, easily obtainable, AU annotations on the EmotionNet dataset [9] and use an adversarial learning framework similar to [25, 6] for training. The generator is tasked with editing a given input image towards a desired target expression while ensuring the edited image can be mapped back to the original image using a cyclic transformation [25, 6, 34], while the discriminator ensures that the edited image looks real and also conforms to the target expression.

To summarize, we develop a method that learns to edit expressions by learning anatomically consistent expression-conditioned deformation fields (without ground-truth deformation annotations) through the explicit disentanglement of the editing process into 'Motion editing' and 'Texture editing' phases. The user studies we have conducted show that facial expression editing methods such as GANimation [25] and ours produce realistic expressions (with an average plausible score of 3.5 out of a maximum score of 4.0) with the scores having a bimodal distribution. Encouraged by these results, we carried out a user study to directly compare the quality of DefGAN's editing results with GANimation [25] and found that, on average, users prefer the editing results of DefGAN over GANimation [25]. In addition to the user study, we also carry extensive expressions edits on a large variety of faces in-the-wild and show that editing expressions by explicit disentanglement of facial movement and texture synthesis leads to more realistic results and while better preserving expression-invariant features of the face.

2. Related Work

Over the past few years there has been a significant amount of work on editing expression and more broadly in transferring images from one domain to another. In this section we discuss work that is most relevant to ours.

Face Manipulation. Extensive work has been done on face manipulation and expression editing within the field computer vision, computer graphics and machine learning. The earliest face expression models relied on mass-and-skin models to model facial movement [10], such models however could not model finer skin movements that are often involved in facial expressions. Another line of research used registered 3D scans of faces to linear low-dimensional embedding of the face [5] by explicitly taking into account variations due to expressions. Though such models often produce realistic expressions edits, they require expensive 3D scans of the same person with different expressions and cannot be easily scaled to learn from larger datasets. Other work using detailed 3D Scans of the human face to edit expressions include [32, 30, 29].

Recent developments in generative adversarial networks have allowed the training of Convolutional Neural Networks to not only generate face images with great photorealism [19] but have also led to the development of a number of unsupervised and weakly supervised facial manipulation methods such as [22, 27]. More specifically, in the case of expression editing, adversarial learning has made possible the use of landmarks [28], discreet expression labels [6] and continuous Action Units [25] as a source of weak supervision. In this work, we choose to use Action Units [8] as a source of weak supervision due their interpretability (each AU corresponds to a change in some region of the face) and wide applicability (AUs can encode any anatomically possible facial expression [8]).

Generative Adversarial Networks. Generative Adversarial Networks (GANs) [11] are a powerful class of generative models that have essentially become standard within the computer vision community for unconditional image generation. GANs working by pitting neural networks against each other; the generator network is tasked with producing samples that are indistinguishable from the real data distribution while the discriminator is tasked with distinguishing between the samples generated by the generator and the real data. Follow up work [3, 12], have significantly improved the training stability of GANs by minimizing the Wasserstien Distance instead of the Jenson-Shanon divergence between distributions. Ever since their inception, GANs have used for a wide variety of computer vision tasks ranging from image inpainting [14, 33] to super-resolution [24].

Conditonal GANs. Conditional GANs are a subset of GANs that use some model conditonal distributions instead on unconditonal distributions. Conditional GANs have been incredibly successfull in a variety of computer vision tasks such as image in-painting [14], super-resolution [24], and domain transfer [34, 6, 20].

Deformation modelling. Over the past few years there has been a growing interest in learning deformations from large scale datasets. Work done in [26] shows that it is possible to model face images as deformations of texture templates in a completely unsupervised manner. Early work on using deformations to edit expressions includes Expression flow [31] which edits expressions by warping the input image using a 2D flow field. This flow field is estimated using another image of the same person with the desired expression. This work builds on prior work on incorporating deformations within convolutional networks such as [16, 7].

Unpaired Image-to-Image Translation. We cast the problem of editing expressions in the framework of unpaired image to image translation where the target image is the image of the person in the input with the target expression. The advent of GANs [11] have made it possible to produce incredibly photorealistic results from when translating from one domain to another. For example, pix2pix [15] is able to generate high quality images object from mere sketches or segmentation maps as inputs. More relevant to this paper is work like [34, 6, 20, 25] which can be used to transfer between various attributes of the face including including expressions.

3. Method

Consider an input image $\mathcal{I}_{\mathbf{x}} \in \mathbb{R}^{H \times W \times 3}$ with some expression $\mathbf{x}$. We'd like to change the expression of the person in $\mathcal{I}_{\mathbf{x}}$ to some target expression $\mathbf{y}$ to give us $\mathcal{I}_{\mathbf{y}}$. Here, the expressions are encoded by AU [8] intensities, $\mathbf{x} = (x_1, \ldots, x_n)$, where each x_i is the intensity of the i^{th} AU scaled between 0 and 1. To carry out this expression edit we transform the image in two stages. First, in the motion editing phase, we deform the input image - modeling facial movement in the pixel space - to conform to the target expression. More specifically, the first stage can be written as follows:

$$\mathcal{I}^*_{\mathbf{y}} = G^W_{\text{Def}}(\mathcal{I}_{\mathbf{x}}, \mathbf{x}, \mathbf{y}) \tag{1}$$

where, G^W_{Def} is the deformation generator that takes as input $\mathcal{I}_{\mathbf{x}}$, the input expression $\mathbf{x}$ and target expression $\mathbf{y}$ and produces the deformed image $\mathcal{I}^*_{\mathbf{y}}$. Although this image could have been appropriately deformed to achieve the target expression y, it might still lack the necessary texture modifications to look realistic. For example, if we had to edit a face with a neutral expression to a face with a grin, the best a deformed image could give us is a very wide (and possibly unrealistic) smile, we'd still need to hallucinate the texture of the teeth to get the correct target expression. We hallucinate the necessary texture, in the texture editing phase, using another convolutional network as follows:

$$\mathcal{I}_{\mathbf{y}} = G_{\text{Texture}}(\mathcal{I}^*_{\mathbf{y}}, \mathbf{y}) \tag{2}$$

where, G_{Texture} is the texture hallucination network. In the interest of brevity, we denote the entire transformation from input image to the final image i.e $\mathcal{I}_{\mathbf{x}} \rightarrow \mathcal{I}_{\mathbf{y}}$, as follows:

$$\mathcal{I}_{\mathbf{y}} = G_{\text{Comp}}(\mathcal{I}_{\mathbf{x}}, \mathbf{x}, \mathbf{y}) \tag{3}$$

where, $G_{\text{Comp}} := G_{\text{Texture}} \odot G^W_{\text{Def}}$ denotes the composition of (1) and (2).

3.1. Architecture

DefGAN consists of three convolutional networks. The deformation generator G^W_{Def}, the texture hallucination network G_{Texture}, and a discriminator D with a critic output, D_c, and an AU regression output, D_{exp}.

3.1.1 Motion Editing Phase

The first stage of DefGAN's expression editing involves deforming the input image, $\mathcal{I}_{\mathbf{x}}$ to $\mathcal{I}^*_{\mathbf{y}}$ such that $\mathcal{I}^*_{\mathbf{y}}$ closely approximates the target expression. We carry out this transformation using a deformation generator G^W_{Def} that first predicts a deformation grid and then warps the input image using it. More specifically, the transformation can be written as follows

$$\begin{aligned} \mathcal{G} &= G_{\text{Def}}(\mathcal{I}_{\mathbf{x}}, \mathbf{x}, \mathbf{y}) \\ \mathcal{I}^*_{\mathbf{y}} &= \text{Warp}(\mathcal{G}, \mathcal{I}_{\mathbf{x}}) \end{aligned} \tag{4}$$

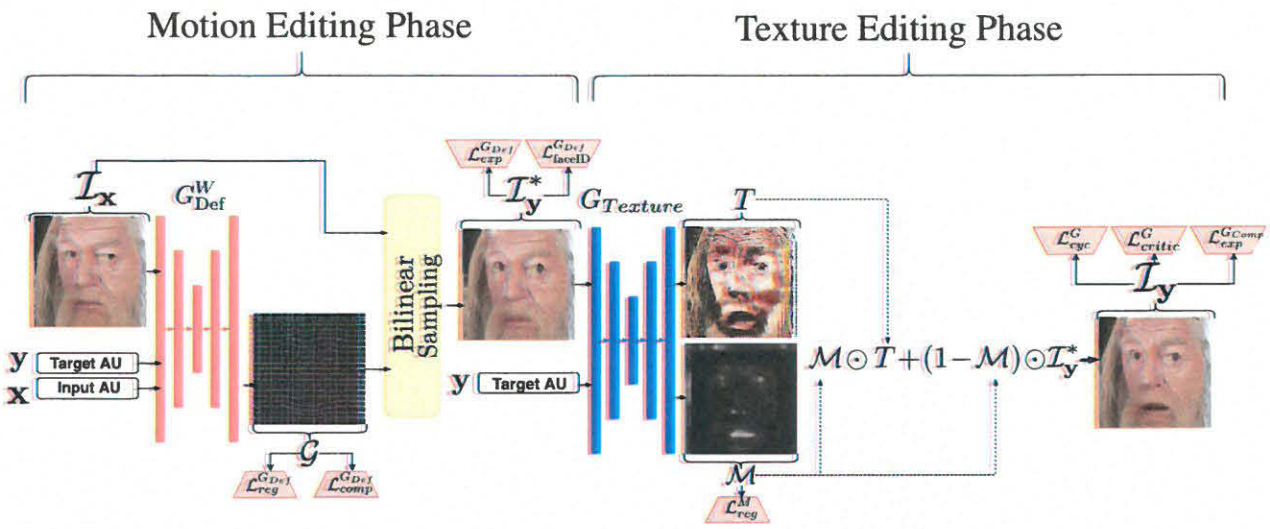


Figure 2: The Architecture of DefGAN and its associated losses. DefGAN edits expressions in two phases. First, in the 'Motion editing' phase, DefGAN models facial movement by predicting a deformation field and deforms the input image using it (note the widening of the eyes in $\mathcal{I}^*_{\mathbf{y}}$). Next, in the 'Texture Editing' phase we hallucinate the necessary textures to complete the editing process (note the opening of the mouth in $\mathcal{I}_{\mathbf{y}}$).

where, $\mathcal{G}$ is the predicted deformation grid. The deformation generator, G^W_{Def}, is just the composition of the above two operations

$$\begin{aligned} G^W_{\text{Def}}(\mathcal{I}_{\mathbf{x}}, \mathbf{x}, \mathbf{y}) &:= \text{Warp}(G_{\text{Def}}(\mathcal{I}_{\mathbf{x}}, \mathbf{x}, \mathbf{y}), \mathcal{I}_{\mathbf{x}}) \\ \mathcal{I}^*_{\mathbf{y}} &= G^W_{\text{Def}}(\mathcal{I}_{\mathbf{x}}, \mathbf{x}, \mathbf{y}) \end{aligned} \tag{5}$$

We use an offset based deformation grid as proposed in [26] with a maximum offset of 5 pixels.

3.1.2 Texture Editing Phase

The second stage of DefGAN's expression edit, the texture editing phase, involves hallucinating the necessary features that cannot be modelled by a deformation to improve the realism of the final image and its fidelity to the target expression. Like prior work [25], we found that a masking mechanism helps texture generator create better quality edits

$$\begin{aligned} \mathcal{T}, \mathcal{M} &= G^*_{\text{Texture}}(\mathcal{I}^*_{\mathbf{y}}, \mathbf{y}) \\ \mathcal{I}_{\mathbf{y}} &= \mathcal{M} \odot T + (1 - \mathcal{M}) \odot \mathcal{I}^*_{\mathbf{y}} \end{aligned} \tag{6}$$

where, $\mathcal{T}$ is the hallucinated texture map and $\mathcal{M}$ is the attention mask. The texture network, G_{Texture} is just the composition of the above two operations.

3.2. Training

Similar to prior work [25, 6] we rely on a GAN based framework [11] to train the deforming network, G^W_{Def} and the texture network, G_{Texture} jointly. In addition to an adversarial loss, we train DefGAN to also minimize an expression loss, a cycle consistency loss, a facial identity loss and a regularization loss on the deformation grid.

Adversarial Loss. In order to ensure DefGAN's image edit looks natural we train the generators to minimize an adversarial loss [11]. Instead of using the standard GAN loss, which corresponds to minimizing the Jenson-Shannon divergence, we use the WGAN-GP loss [12] which minimizes the Earth Mover Distance between the generated and the real distribution. Specifically, let $\mathcal{I}_{\mathbf{x}}$ be the input image with expression $\mathbf{x}$, let $\mathbf{y}$ be the target expression and let $\mathcal{P}_r$ be the real distribution of images. The critic loss for the discriminator, D is given as follows

$$\begin{aligned} \mathcal{L}^D_{critic} =& \mathbb{E}_{\mathcal{I}_{\mathbf{x}} \sim \mathcal{P}_r} \left[D_c \left(G_{\text{Comp}}(\mathcal{I}_{\mathbf{x}}, \mathbf{x}, \mathbf{y}) \right) \right] \\ & - \mathbb{E}_{\mathcal{I}_{\mathbf{x}} \sim \mathcal{P}_r} \left[D_c \left(\mathcal{I}_{\mathbf{x}} \right) \right] \\ & + \lambda_{gp} \mathbb{E}_{\hat{\mathcal{I}} \sim \hat{\mathcal{P}}} \left[\left(\| \nabla_{\hat{\mathcal{I}}} D_c(\hat{\mathcal{I}}) \|_2 - 1 \right)^2 \right] \end{aligned} \tag{7}$$

Where, D_c is the critic output of the discriminator D, λ_{gp} is the gradient penalty coefficient and $\hat{\mathcal{P}}$ is the interpolated distribution. The generators, G^W_{Def} and G_{Texture} are trained to 'please' the critic by maximizing the critic score. We express this loss as:

$$\mathcal{L}^G_{critic} = -\mathbb{E}_{\mathcal{I}_{\mathbf{x}} \sim \mathcal{P}_r} \left[D_c \left(G_{\text{Comp}}(\mathcal{I}_{\mathbf{x}}, \mathbf{x}, \mathbf{y}) \right) \right] \tag{8}$$

Expression Loss. To ensure that the generators, while producing a realistic image, are also generating the correct

target expression we add a loss that penalizes deviations from the target expression. This loss is defined using the AU output of the discriminator, D_{exp}, that is trained to predict the AU intensities for any given input image $\mathcal{I}_{\mathbf{x}}$. The loss is defined as follows:

$$\begin{aligned}\mathcal{L}_{exp}^{G_{Def}} &= \lambda_{exp}^{G_{Def}} \mathop{\mathbb{E}}_{\mathcal{I}_{\mathbf{x}}\sim\mathcal{P}_r} \left[\|D_{exp}(G_{\text{Def}}(\mathcal{I}_{\mathbf{x}}, \mathbf{x}, \mathbf{y})) - \mathbf{y}\|_2^2\right] \\ \mathcal{L}_{exp}^{G_{Comp}} &= \lambda_{exp}^{G_{Comp}} \mathop{\mathbb{E}}_{\mathcal{I}_{\mathbf{x}}\sim\mathcal{P}_r} \left[\|D_{exp}(G_{\text{Comp}}(\mathcal{I}_{\mathbf{x}}, \mathbf{x}, \mathbf{y})) - \mathbf{y}\|_2^2\right] \\ \mathcal{L}_{exp}^{G} &= \mathcal{L}_{exp}^{G_{Def}} + \mathcal{L}_{exp}^{G_{Comp}}\end{aligned} \tag{9}$$

Here, $\lambda_{exp}^{G_{Comp}}$ and $\lambda_{exp}^{G_{Def}}$ are the coefficients of each term. We apply the expression loss both on the final image output $\mathcal{I}_{\mathbf{y}} = G_{\text{Comp}}(\mathcal{I}_{\mathbf{x}}, \mathbf{x}, \mathbf{y})$ and the intermediate image output $\mathcal{I}_{\mathbf{y}}^* = G_{\text{Def}}^W(\mathcal{I}_{\mathbf{x}}, \mathbf{x}, \mathbf{y})$. The AU output, D_{exp}, is trained to minimize the AU prediction error on real images

$$\mathcal{L}_{exp}^{D} = \mathbb{E}_{\mathcal{I}_{\mathbf{x}}\sim\mathcal{P}_r} \left[\|D_{exp}(\mathcal{I}_{\mathbf{x}}) - \mathbf{x}\|_2^2\right] \tag{10}$$

Cycle Loss. In order to preserve subject identity as Def-GAN edits the image, we enforce a cycle consistency loss on the generator networks as follows:

$$\mathcal{L}_{cyc}^{G} = \lambda_{cyc}\mathbb{E}_{\mathcal{I}_{\mathbf{x}}\sim\mathcal{P}_r} \left[\|G_{\text{Comp}}(G_{\text{Comp}}(\mathcal{I}_{\mathbf{x}}, \mathbf{x}, \mathbf{y}), \mathbf{y}, \mathbf{x}) - \mathcal{I}_{\mathbf{x}}\|_1\right] \tag{11}$$

In the absence of ground truth images for each person with different annotated expressions, we found that this cycle loss ensures the identity of the person does not change as the expression changes.

Face Identity Loss. We regularize the deformation generator, G_{Def}^W, by ensuring that it preserves the identity of the person as it deforms the input image. More specifically, we maximize the cosine similarity between the OpenFace [2] embedding of the deformed input image, $\mathcal{I}_{\mathbf{y}}^*$, and the input image $\mathcal{I}_{\mathbf{x}}$. This loss can be expressed as

$$\mathcal{L}_{\text{faceID}}^{G_{Def}} = \mathbb{E}_{\mathcal{I}_{\mathbf{x}}\sim\mathcal{P}_r} \left[(1 - \cos(\text{OpenFace}(\mathcal{I}_{\mathbf{y}}^*), \text{OpenFace}(\mathcal{I}_{\mathbf{x}}))\right] \tag{12}$$

Composition Loss. We found that imposing a composition loss on the generated deformation grid $\mathcal{G}$ was useful in producing realistic expression edits. The grid composition loss is defined as follows:

$$\mathcal{L}_{comp}^{G_{Def}} = \lambda_{comp}\mathbb{E}_{\mathcal{I}_{\mathbf{x}}\sim\mathcal{P}_r} \left[(\text{Warp}(\mathcal{G}_{cyc}, \mathcal{G}) - \mathcal{I}_{Def})^2\right] \tag{13}$$

where, $\mathcal{G}_{cyc}$ is the deformation grid produced during the cycle transformation and $\mathcal{I}_{Def}$ is the identity deformation grid.

Regularization. To ensure smoothness of the generated deformation fields we add a TV-regularization term on the deformation grid, $\mathcal{G}$ as defined in (4), and also penalize the difference between $\mathcal{G}$ and the identity deformation. The regularization terms can be written as follows

$$\begin{aligned}\mathcal{L}_{reg}^{G_{Def}} &= \lambda_{eye}^{\mathcal{G}}\mathbb{E}_{\mathcal{I}_{\mathbf{x}}\sim\mathcal{P}_r} \left[(\mathcal{G} - \mathcal{I}_{Def})^2\right] \\ &+ \lambda_{TV}^{G}\mathbb{E}_{\mathcal{I}_{\mathbf{x}}\sim\mathcal{P}_r} \left[\sum_{i,j}^{H,W} (\mathcal{G}_{i+1,j} - \mathcal{G}_{i+1,j})^2 + (\mathcal{G}_{i,j+1} - \mathcal{G}_{i,j})^2\right]\end{aligned} \tag{14}$$

where, $\mathcal{I}_{Def}$ is the identity deformation grid. We also add a similar regularization to the mask, $\mathcal{M}$ that is generated during the texture editing phase. This regularization term is as follows

$$\begin{aligned}\mathcal{L}_{reg}^{M} &= \lambda_{eye}^{M}\mathbb{E}_{\mathcal{I}_{\mathbf{x}}\sim\mathcal{P}_r} \left[\|\mathcal{M}\|_2^2\right] \\ &+ \lambda_{TV}^{\mathcal{M}}\mathbb{E}_{\mathcal{I}_{\mathbf{x}}\sim\mathcal{P}_r} \left[\sum_{i,j}^{H,W} (\mathcal{M}_{i+1,j} - \mathcal{M}_{i+1,j})^2 + (\mathcal{M}_{i,j+1} - \mathcal{M}_{i,j})^2\right]\end{aligned} \tag{15}$$

Final Loss. The total loss on the generators is

$$\mathcal{L}_{Total}^{G} = \mathcal{L}_{critic}^{G} + \mathcal{L}_{exp}^{G} + \mathcal{L}_{cyc}^{G} + \mathcal{L}_{\text{faceID}}^{G_{Def}} + \mathcal{L}_{comp}^{G_{Def}} + \mathcal{L}_{reg}^{G_{Def}} + \mathcal{L}_{reg}^{M} \tag{16}$$

We minimize this loss over the parameters of G_{Def} and G_{Texture} to convergence

$$G_{\text{Def}}^{W*}, G_{\text{Texture}}^{*} = \mathop{\text{argmin}}_{G_{\text{Def}}^W, G_{\text{Texture}}} \mathcal{L}_{Total}^{G} \tag{17}$$

The total loss on the discriminator is

$$\mathcal{L}_{Total}^{D} = \mathcal{L}_{critic}^{D} + \mathcal{L}_{exp}^{D} \tag{18}$$

The discriminator is trained to minimize this loss

$$D^* = \mathop{\text{argmin}}_{D} \mathcal{L}_{Total}^{D} \tag{19}$$

4. Experiments and Results

We evaluated our model on a variety of facial identities and on a range of expression editing tasks to test its quality and robustness. Around 170 different facial identities from the CelebA-HQ dataset [18] were used for evaluation. In addition to CelebA-HQ [18] we scraped 40 more images from the internet with variations in pose, illumination and facial attributes to test the robustness of our model on more challenging input images. First, we show the results of editing expressions on a number of in-the-wild faces and measure the change in facial identity after the expression edit by comparing the OpenFace [2] embeddings of the edited

image and the input image. Next, we show the results of manipulating single Action Units [8] using our model and we finally discuss the results of a user study conducted to determine which model among DefGAN and GANimation [25] produced better expression edits on in-the-wild images as judged by humans.

4.1. Training Details

DefGAN. DefGAN was trained on a subset of the EmotionNet dataset [9] containing 190k images. We use the Adam optimizer [21] with an initial learning rate of $1e-4$, $\beta_1 = 0.5$, $\beta_2 = 0.999$ and a batch size of 25. The model was trained for 40 epochs with the learning rate decaying to 0 over the last 20 epochs. The deformation generator and the texture generator were optimized jointly. The critic was trained for 10 steps for every step of the generators.

GANimation. GANimation [25] was trained on the same subset of the EmotionNet dataset [9] containing 190k images. We used the hyperparameters used by the authors in [25] with the only difference that we train for 40 epochs.

4.2. Expression Synthesis on Wild Faces

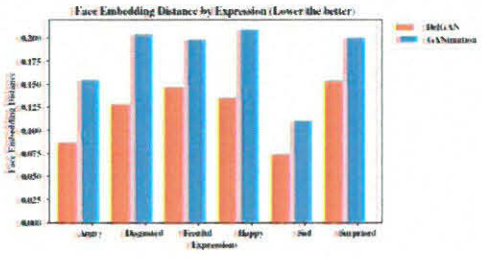


Figure 3: **Face Embedding Distance.** Here we show the distance between the CMU-OpenFace [2] embeddings of the input image and the images edited by DefGAN and GANimation. As one can see, DefGAN's edits consistently preserve facial identity better than GANimations's.

Method Name	FID Score
GANimation [21]	4.75
DefGAN	**3.82**

Table 1: FID Scores of GANimation [25] and DefGAN.

We first tested our method by performing expression edits on in-the-wild images from CelebA-HQ dataset [18] and 40 more images scraped from the internet. The AU representation for each expression was computed by running the OpenFace AU detector [4] on all the peak expression images of a randomly selected person from the MUG dataset [1]; which consists of seven labelled expressions (anger, disgust, fear, neutral, happy, sad and surprise) for 84 persons. Fig.4 shows the results of expression edits performed on a few images from the internet and from CelebA-HQ [18]. As can be seen, our model consistently performs better edits across all expressions and faces. In particular, we noticed that GANimation [25] tends to distort the face by either producing artifacts (for example, the result of 'Disgust' in row 3 of Fig.4) or by 'over-editing' (for example, the results of 'Happy' in row 1, 2 and 4 of Fig.4) which we posit is due to its complete reliance on the hallucination mechanism. In contrast, DefGAN, due its use of a deformation to warp the face to conform to the target expression, only hallucinates the necessary details and does not produce such artifacts. The Fréchet Inception Distance (FID) Score [13] has become a standard measure to evaluate the realism of generative models. Lower the FID score of a model the more realistic are its images. Table 1 shows the FID [13] score of the images edited by DefGAN and GANimation, and as one can see this further suggests that the editing results of DefGAN are more realistic than those of GANimation.

Fig.5 shows the absolute pixel-wise difference between the input image and the edited image across a few target expressions. Ideally, we'd only see differences in regions that correspond to the expression change. As can be seen in Fig.5 , the edits made by DefGAN are significantly more concentrated to the regions relevant to the final expression than the edits made by GANimation which tend to be more spread out. Fig. 3 shows the distance between the edited image and the input image in the OpenFace [2] embedding space on fifty different randomly chosen representations of each expression. DefGAN retains the input facial identity much better than GANimation across all the six target expressions. The retention of facial identity can also be seen visually in Fig. 4 , where attributes such as the beard and eyebrows of the edited images (results of 'Disgust' in all rows, the results of 'Happy' in row 2) have greater fidelity to the input with DefGAN's edits than GANimation's edits which tend to either erase or thicken them. Fig. 1 shows how deformations can be especially helpful in certain expression edits, such as going from a close to neutral expression to a 'disgust' expression. In this expression transformation we can see that the deformation in DefGAN does most of the work of converting the input face to the 'disgust' expression while the hallucination only adds minor details to the final image.

4.3. Learnt facial movements conditioned on Action Units

In this section, we analyze the effect of changing individual AUs (AU14, AU5, AU14, and AU26) on an input face. We show the effect both on the deformed image. As can

Figure 4: Editing Facial Expressions. Here we show images edited by DefGAN and GANimation [25] to various target expressions. GANimation [25] tends to produce artifacts (results of 'Anger' and 'Disgust' in row 3) or ends up hallucinating inaccurate textures (results of 'Happy' in row 2 and results of 'Anger' in row 4). In contrast, the editing results of DefGAN are more consistent with fewer artifacts and more accurate textures.

be seen from Fig. 6 , DefGAN successfully learns to faithfully model facial movement through its deformation mechanism. For example, when increasing the intensity of AU5 (Upper Lid Raiser) we clearly see the eyebrows raising up while other regions of the face remain unchanged. Coherent facial movements resulting from deformations can also be

Figure 5: Difference Images. This figure shows the pixel-wise absolute difference between images edited by GANimation and the input image. As one can see, DefGAN only changes the parts of the input that are required to attain the target expression while GANimation [25] changes larger portions of the input image regardless of target expression.

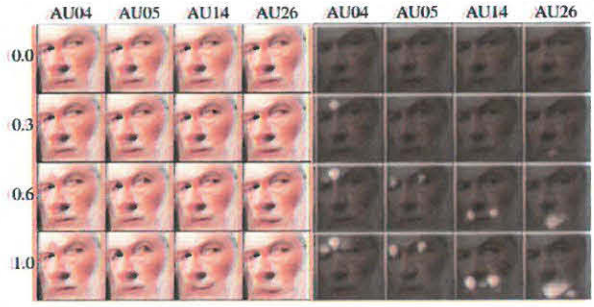


Figure 6: Learnt Action Unit conditioned facial movements. **Left:** Here we show the effects of single AU activations on the deformed image $\mathcal{I}_{\mathbf{y}}^{*}$. **Right:** Here we show the regions of the face affected by the change in intensity of the corresponding AU. As can be seen, changing the intensity of any particular AU causes smooth changes in the corresponding facial regions akin to the results of true facial muscle movement as encoded by that AU.

seen as we change the intesities of AU4 (Brow Lowerer), AU14 (Dimpler) and AU26 (Jaw Drop). Examples of more AU activations along with results of the final edited images can be found in the appendix.

4.4. User Study

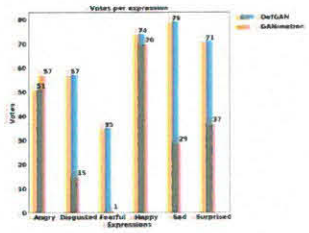


Figure 7: Results of the User Study. Here we show the results of the second stage of the user study. About 63% of the total votes went to DefGAN as compared to 37% to GANimation [25]. Across expressions, the editing results of DefGAN were preferred over the editing results of GANimaton [25] with the only exception being 'Anger' and 'Happy' where the results were close.

We evaluate the quality of expression edits done by DefGAN and GANimation [25] by conducting a user study. We carry out the user study in two stages, in the first stage (55 users) we evaluate how realistic are the editing results of each method, without directly comparing them. We randomly sample 10 images edited by each method and show them to the users, asking them to give rate the plausibility of the image from 1 (Definitely implausible) to 4 (Definitely plausible). Each user was shown the same set of 20 images (10 from each method, in random order). The results of the user study showed that around 60% of all the images shown were rated as plausible with the average plausible image having a score of 3.5 and the average implausible image having a score 1.67. In the next stage of the user study we directly compared the results of DefGAN and GANimation [25]. Sixteen random in-the-wild images were chosen for this stage. These images were selected to have a close to neutral input expression and to not have extreme poses. Each image was assigned a random target expression from the following expressions: happy, disgust, sad, fear, angry, and surprise. The results of our method and GANimation [25] were placed side-by-side and users were asked to judge the quality of each edited image with respect to its fidelity to the facial identity of the input image, the closeness to the target expression and the overall plausibility of the image. The order in which the options were shown was randomly chosen for each question to avoid user bias. The results of the user study are shown in Fig. 7. Users mostly preferred the results of DefGAN over GANimation [25] on the whole and across most expressions. The results were quite close when the target expression was 'angry' and

'happy' but were overwhelmingly in our favor for all other expressions. Further details of the user study are given in the appendix. The results of the user study provide further evidence that expression editing results of DefGAN are not only more realistic but also preserve facial identity and attain the target expression better than GANimation.

5. Conclusion

We presented a novel method for facial expression editing that can produce high quality expression edits on in-the-wild images. We leverage the most recent advances in deformation modelling [26] and expression editing [25] to create a method that is able to learn facial movements as deformations without using ground-truth deformation annotations. The explicit use of a deformation in the "motion-editing" phase allows DefGAN to perform targeted edits on the input face which extensive evaluations show are not only able to produce very high quality edited images but also better retain expression invariant facial attributes. In future work, we hope to improve expression modelling by taking into account also the temporal nature of expression and possibly extending expression editing to in-the-wild video sequences.

Acknowledgements. We would like to thank Francesc Moreno-Noguer for his valuable comments. This work was supported by a gift from Adobe, NSF grants CNS-1718014 and DMS 1737876, the Partner University Fund, and the SUNY2020 Infrastructure Transportation Security Center.

References

[1] Niki Aifanti, Christos Papachristou, and Anastasios Delopoulos. The mug facial expression database. 11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10, pages 1–4, 2010. 6

[2] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016. 5 ,6 , 10 ,11

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Proc. ICML, pages 214–223, 2017. 3

[4] Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. Openface 2.0: Facial behavior analysis toolkit. In 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), pages 59–66. IEEE, 2018. 6 ,11

[5] Volker Blanz, Thomas Vetter, et al. A morphable model for the synthesis of 3d faces. 1999. 2

[6] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018. 1 ,2 ,3 ,4 ,17 , 18

[7] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In Proceedings of the IEEE international conference on computer vision, pages 764–773, 2017. 3

[8] Friesen W Ekman, P. Facial action coding system: A technique for the measurement of facial movement. In Consulting Psychologists Press, 1978. 1 ,2 ,3 ,6

[9] C Fabian Benitez-Quiroz, Ramprakash Srinivasan, and Aleix M Martinez. Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5562–5570, 2016. 2 ,6

[10] Martin A Fischler and Robert A Elschlager. The representation and matching of pictorial structures. IEEE Transactions on computers, (1):67–92, 1973. 2

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014. 1 ,3 ,4

[12] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In Advances in Neural Information Processing Systems, pages 5767–5777, 2017. 3 ,4

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 6626–6637. Curran Associates, Inc., 2017. 6

[14] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and Locally Consistent Image Completion. Proc. SIGGRAPH, 36(4):107:1–107:14, 2017. 3

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In Proc. CVPR, 2017. 3 ,17

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In Advances in neural information processing systems, pages 2017–2025, 2015. 3

[17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In European conference on computer vision, pages 694–711. Springer, 2016. 17

[18] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In Proc. ICLR, 2018. 5 ,6

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. CoRR, abs/1812.04948, 2018. 2

[20] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pages 1857–1865. JMLR. org, 2017. 3

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Proc. ICLR, 2015. 6

[22] Jean Kossaifi, Linh Tran, Yannis Panagakis, and Maja Pantic. GAGAN: geometry-aware generative adversarial networks. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pages 878–887, 2018. 2

[23] Oliver Langner, Ron Dotsch, Gijsbert Bijlstra, Daniel HJ Wigboldus, Skyler T Hawk, and AD Van Knippenberg. Presentation and validation of the radboud faces database. Cognition and emotion, 24(8):1377–1388, 2010. 11 ,17 ,18

[24] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. ArXiv e-prints, Sept. 2016. 3

[25] A. Pumarola, A. Agudo, A.M. Martinez, A. Sanfeliu, and F. Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In Proceedings of the European Conference on Computer Vision (ECCV), 2018. 1 ,2 ,3 ,4 ,6 ,7 ,8 ,9 ,10 ,11 ,12 ,17 ,18

[26] Zhixin Shu, Mihir Sahasrabudhe, Riza Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In Proceedings of the European Conference on Computer Vision (ECCV), pages 650–665, 2018. 2 ,3 ,4 ,9

[27] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras. Neural face editing with intrinsic image disentangling. In Computer Vision and Pattern Recognition, 2017. CVPR 2017. IEEE Conference on, pages –. IEEE, 2017. 2

[28] Lingxiao Song, Zhihe Lu, Ran He, Zhenan Sun, and Tieniu Tan. Geometry guided adversarial facial expression synthesis. In Proceedings of the 26th ACM International Conference on Multimedia, MM '18, pages 627–635, New York, NY, USA, 2018. ACM. 2

[29] M. Song, Z. Dong, C. Theobalt, H. Wang, Z. Liu, and H. Seidel. A generic framework for efficient 2-d and 3-d facial expression analogy. IEEE Transactions on Multimedia, 9(7):1384–1395, Nov 2007. 2

[30] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. Face transfer with multilinear models. ACM Transactions on Graphics, 24(3):426–433, 2005. 2

[31] Fei Yang, Jue Wang, Eli Shechtman, Lubomir Bourdev, and Dimitri Metaxas. Expression flow for 3d-aware face component transfer. In ACM SIGGRAPH 2011 Papers, SIGGRAPH '11, pages 60:1–60:10, New York, NY, USA, 2011. ACM. 3

[32] Chan-Su Lee Song Zhang Zhiguo Li Dimitris Samaras Dimitris Metaxas Ahmed Elgammal Peisen Huang Yang Wang, Xiaolei Huang. High resolution acquisition, learning and transfer of dyanmic 3-d facial expressions. In Computer Graphics Forum, pages III: 677–686, 2004. 2

[33] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with deep generative models. In Proc. CVPR, pages 6882–6890, 2017. 3

[34] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Computer Vision (ICCV), 2017 IEEE International Conference on, 2017. 2 , 3 ,17

Appendix

A. Additional Results on Expression Transformations

In this section we provide additional results of expression transformations as well as comparisons with previous state-of-the-art [25]. In Fig. 9 , we show expression transformations using DefGAN on face images with variations in pose, lighting and ethnicity. Our results faithfully capture the details of each facial expression in each example while successfully maintaining characteristic details of the person in the input. Thanks to deformation disentanglement, the deformation generator successfully captures the facial movement of each expression, generating vivid details on facial regions such as the eyebrows, mouth etc.

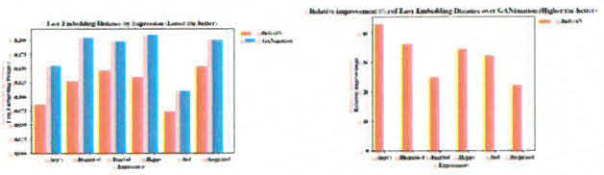


Figure 8: **Face Embedding Distance. Left:** Here we show the distance between the CMU-OpenFace [2] embeddings of the input image and the images edited by DefGAN and GANimation. As one can see, DefGAN's edits consistently preserve facial identity better than GANimations's. **Right:** Here we show the average relative improvement of the Face Embedding Distance between DefGAN's and GANimation's edits. We see that "Angry" has the highest relative improvement.

In Fig. 10 , we highlight the utility of explicit deformation modeling for facial expressions by providing some detailed comparisons with the prior state-of-the-art, GANimation [25]. Due to the explicit deformation modeling for AUs, DefGAN disentangles the editing task into an image deformation followed by a texture hallucination. This allows each generator within DefGAN, i.e the deformation generator and the texture generator, to perform a more focused task. As a result, our expression editing results better retain characteristic details of the input face, such as facial hair, lip color and eyebrows. In contrast, GANimation [25], sometimes produces seemingly arbitrary and unwanted facial details, such as missing/inaccurate facial hair or missing eyebrows.

Figure 9: **Expression Transformation with DefGAN.** We provide additional results on expression transformation using our approach. Our results faithfully capture the details of each facial expression for each example, while successfully maintaining characteristic details of the person in the input image.

A consequence of DefGAN better preserving characteristic facial details of the input is that it also preserves facial identity better. In Fig. 8 , we provide further numerical evidence of this by comparing the CMU-OpenFace[2] embeddings of the edited image and the input image using GANimation [25] as a baseline. CMU-OpenFace [2] uses a facial recognition network to map an input face image to a 128-dimensional embedding. The distance between two facial identities in this embedded space is given by $1-cos(I_x, I_y)$ where I_x and I_y are the embeddings and cos is the cosine similarity. We calculate this facial embedding distance for fifty different instances of AUs for each expression. More specifically, we randomly select fifty different people from the RaFD Dataset [23] and calculate the AUs using Cambridge-Openface [4] on their frontal expression-annotated images. We see that across all expressions and their instances DefGAN preserves facial identity better than GANimation [25].

B. Additional Results on AU Activations

In this section, we complement the results discussed in Section 4.3 of the paper and show the results of manipulating individual AUs on both the deformed image and the output.

In Fig. 11 , we observe the effects of changing the AU intensity on the deformed image $\mathcal{I}_{\mathbf{y}}^{*}$. We clearly see that DefGAN is able to faithfully model facial movements through

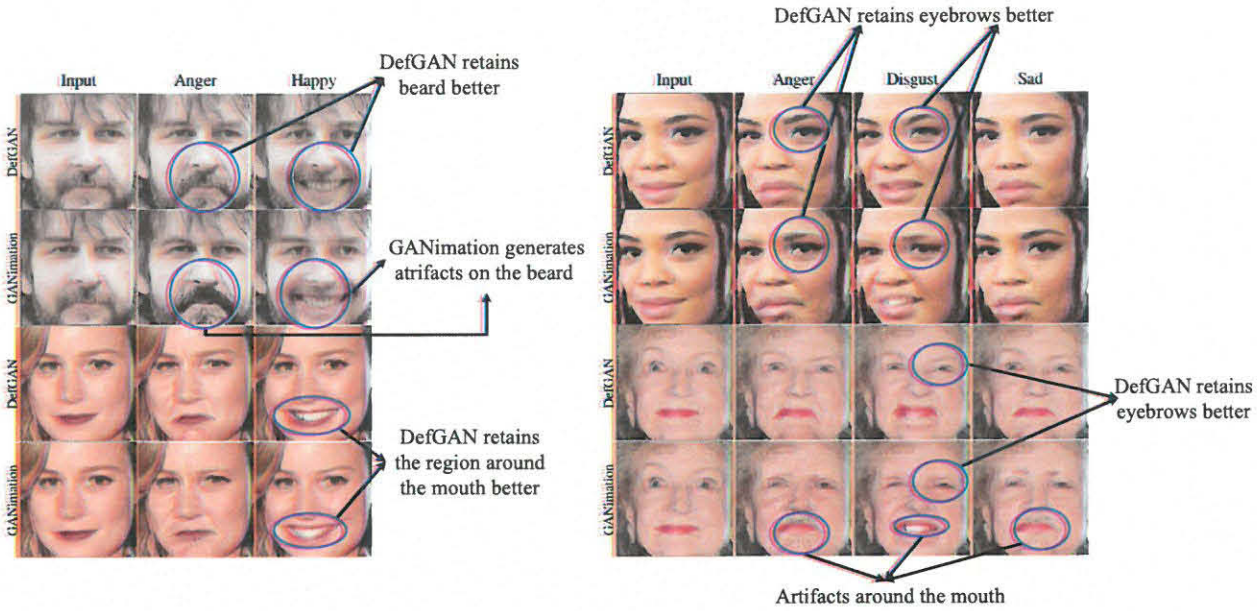


Figure 10: **Result comparison of DefGAN and GANimation [25] in details.** We hereby demonstrate the clear benefit of DefGAN comparing to GANimation by highlighting details of several expression edits. Due to the explicit deformation modeling for AUs, DefGAN disentangles the editing task into an image deformation followed by a texture hallucination. This allows each generator within DefGAN, i.e the deformation generator and the texture generator, to perform a more focused task. As a result, our expression editing results better retain characteristic details of the input face, such as facial hair, lip color and eyebrows. On the contrary, GANimation [25] sometimes produces seemingly arbitrary details, including missing/inaccurate facial hair, missing eyebrows etc.

its deformation mechanism. For example, with the increasing intensity of AU02 (Outer Brow Raiser) and AU07 (Lid Tightener) we see that the eyebrows moving up and we see the eyes becoming smaller respectively. Similarly, with increasing intensity of AU26 (Jaw Drop) and AU14 (Dimpler) we see changes around the mouth.

AU26, which represents a "Jaw Drop" is a good example of an AU who's intensity change entails both facial movement and the generation of new textures (the opening of the mouth), and we see that DefGAN handles both really well. In the "AU26" column of Fig. 11 we see DefGAN changing the region around the mouth using deformations (mimicking facial muscle movements) and in Fig. 12 ("AU26" column) we see DefGAN hallucinating the texture of an open mouth thus completing the editing process.

In cases like that of AU45 (Blink), which requires significant generation of new textures (the opening and closing of eyelids), we see that DefGAN producing little change in the deformed image ("AU45" column of Fig. 11) and generating the texture of the eyelids in the final output ("AU45" column of Fig. 12).

C. Additional Results for Concentration of Edits

In this section, we provide further evidence (in addition to that given in Section 4.2 of the paper) of DefGAN's edits being more targeted and concentrated than GANimation's, using the difference image. Fig. 13 and Fig. 14 show the absolute pixel-wise difference between the images edited by DefGAN and GANimation and the input image.

In Fig. 13 , the biggest differences can be seen when the target expressions are "Disgust", "Happy" and "Sad". Specifically, in the case of the person at the bottom of Fig. 13 (the old man with a beard) we see than GANimation makes significant changes to the beard and the eyebrows when the target expressions are "Disgust" and "Sad".

Similarly, in Fig. 14 we see the biggest changes in "Anger" and "Fear". We also observe the same effects in Fig. 10 where GANimation produces more change than desired. This tendency of GANimation to "overedit" also manifests itself in Fig. 8 where we clearly see that it introduces more change in facial identity of the edited image than DefGAN does.

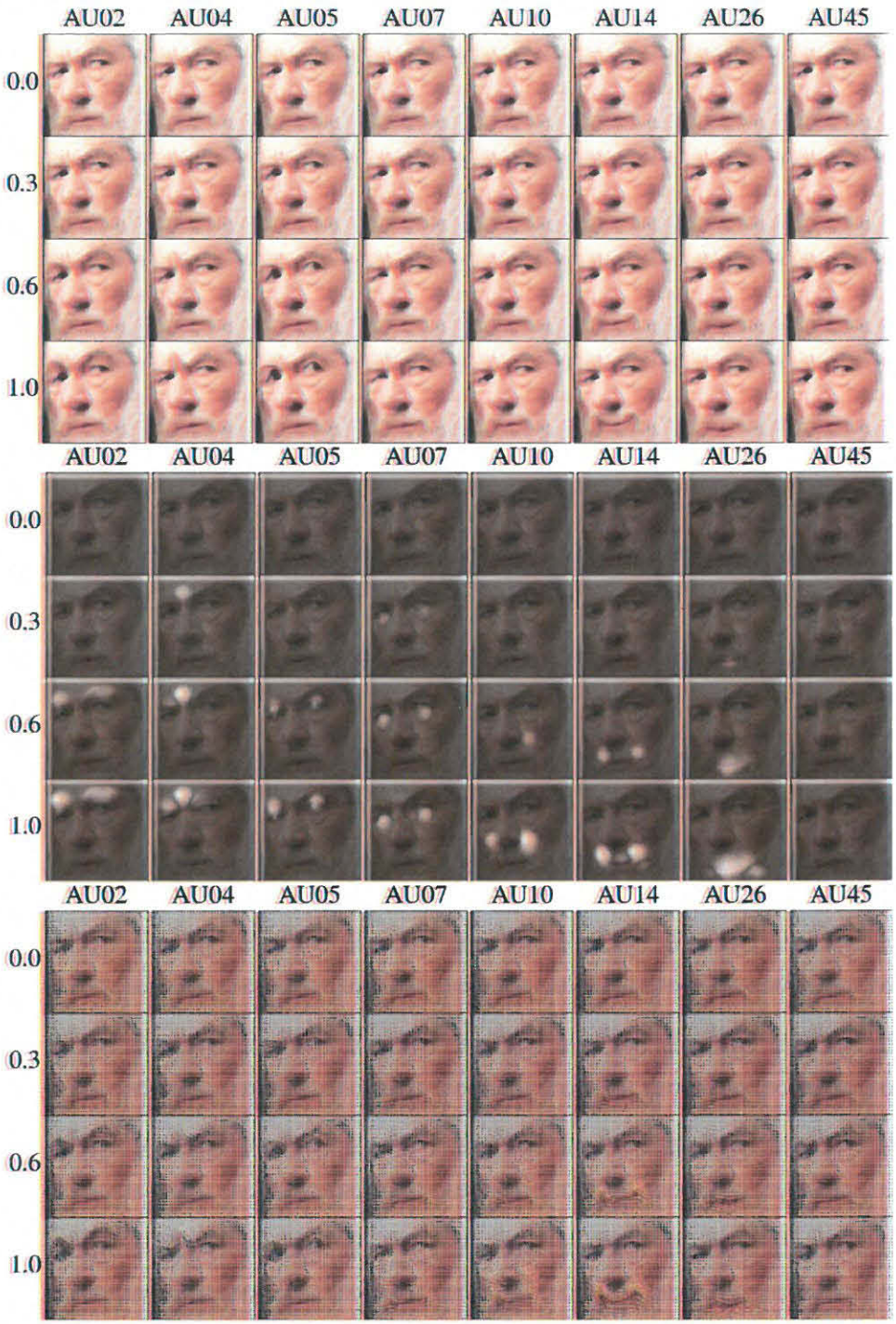


Figure 11: **Effects of Single AU editing on the Deformed image $\mathcal{I}_{\mathbf{y}}^{*}$.** In this figure we show the effects of changing single AU activations on the deformed input image. **Top:** As can be seen, changing the intensity of AUs causes smooth changes in facial regions. **Middle:** This figure shows the regions of the face that are deformed as we change the intensity of each AU. We see that the movement is only restricted to regions of the face relevant to the corresponding AU being changed, akin to the results of true facial movement. **Bottom:** Similar to the figure in the middle, looking at the deformation grid, we see that the movement is only restricted to regions of the face relevant to the corresponding AU being changed.

Figure 12: **Effects of Single AU editing on the Output Image $\mathcal{I}_y$.** In this figure we show the effects of single AU activations on the final edited image. As can be seen, changing the intensity of AUs causes smooth changes facial expression without any artifacts or inaccurate texture hallucinations.

Figure 13: **Difference Images.** The rows marked as "GANimation DiffImg" and "DefGAN DiffImg" show the absolute pixel-wise difference between the edit made by the respective methods and the input image. As can be seen, DefGAN's edits are more concentrated to the region relevant to the expression transformation. For example, the transformation to 'Happy' and 'Sad' of both people shown above. DefGAN almost exclusively changes the area around the mouth while GANimation's edits are spread out all across the input face.

Figure 14: **Difference Images.** The rows marked as "GANimation DiffImg" and "DefGAN DiffImg" show the absolute pixelwise difference between the edit made by the respective methods and the input image. As can be seen, DefGAN's edits are more concentrated to the region relevant to the expression transformation. For example, the transformation to 'Anger' and 'Fear' of both people shown above. DefGAN almost exclusively changes the area around the eyebrows and the mouth GANimation's edits are spread out all across the input face.

D. Training Details

D.1. Architecture Details

The texture generator G_{Texture}. The texture network is identical to the one used in GANimation [25] which builds upon the variation of the network proposed by Johnson et al. [17] and was used by Zhu et al. in [34] to achieve impressive results for image-to-image mapping.

The deformation generator G^{W}_{Def}. The deformation generator is identical to the texture generator except that we replace the last three convolutional layers (the layers responsible for upsampling) with Bilinear Upsampling layers.

The Discriminator D. The Discriminator has a PatchGAN [15] architecture where each element of the output matrix X_{ij} represents the probability of the overlapping patch ij to be real. We also add an AU output head to the penultimate layer of D that estimates the AU output $\mathbf{x} = (x_1, \ldots, x_N)$ of the input image $\mathcal{I}_{\mathbf{x}}$.

D.2. Coefficient Values

During training we use the following coefficient values for the losses

$$\lambda_{exp}^{G_{Comp}} = 4000.0 \tag{20}$$

$$\lambda_{exp}^{G_{Def}} = 1000.0 \tag{21}$$

$$\lambda_{cyc} = 100.0 \tag{22}$$

$$\lambda_{comp} = 10.0 \tag{23}$$

$$\lambda_{eye}^{\mathcal{G}} = 0.1 \tag{24}$$

$$\lambda_{TV}^{\mathcal{G}} = 1e-5 \tag{25}$$

$$\lambda_{eye}^{\mathcal{M}} = 0.1 \tag{26}$$

$$\lambda_{TV}^{\mathcal{M}} = 1e-5 \tag{27}$$

$$\tag{28}$$

E. User Study

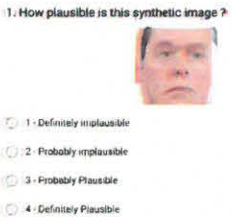


Figure 15: **User Study Stage 1.** Here we show an example of a question asked in the first stage of the user study.

We evaluate the realism of edits made by DefGAN and GANimation [25] independently and also perform a direct comparison of their edits using user studies. In the first stage, we ask users to rate the 'Plausibility' of an image given in the question, where the image shown is an edit made either by DefGAN or GANimation [25], an example form is given in Fig. 15 .

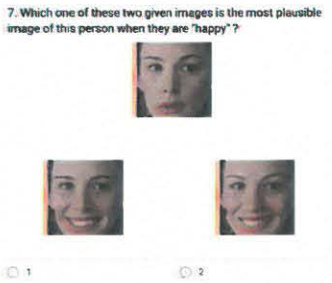


Figure 16: **User Study Stage 2.** Here we show an example of a question asked in the second stage of the user study.

In the next stage, we perform a direct comparison between the images edited by GANimation [25] and DefGAN and we ask the user to choose the image that is more 'plausible' and is also more faithful to the target expression, an example of the form is given in Fig. 16 . The order of the results shown was randomly chosen for each question.

F. Comparison with StarGAN

Here we compare against StarGAN [6] on the RaFD Dataset [23] and on in-the-wild images. In Fig. 17 , we see that both DefGAN and GANimation perform on par with StarGAN, especially considering StarGAN was trained on RaFD [23] while GANimation [25] and DefGAN were not. On the flip side, when edits are carried out on in-the-wild images as seen in Fig. 18 StarGAN performs much worse for the same reason.

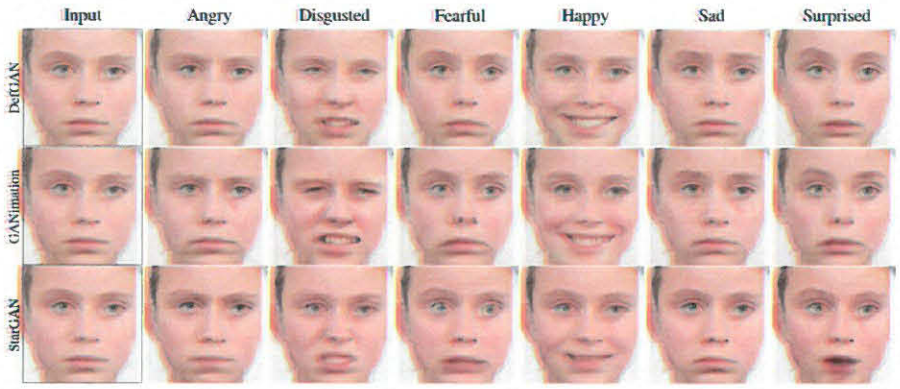


Figure 17: **StarGAN on RaFD.** In this figure we compare DefGAN, GANimation [25] and StarGAN [6] on the RaFD Dataset [23].

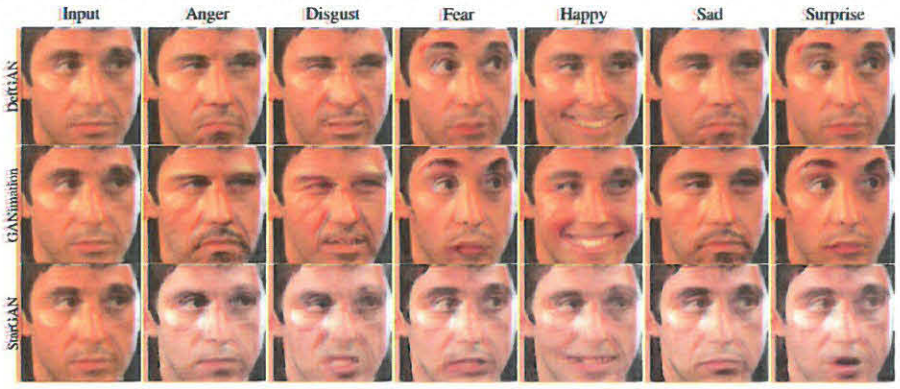


Figure 18: **StarGAN on in-the-wild.** In this figure we compare DefGAN, GANimation [25] and StarGAN [6] on an in-the-wild image.