# Facial Expression Animation through Action Units Transfer in Latent Space

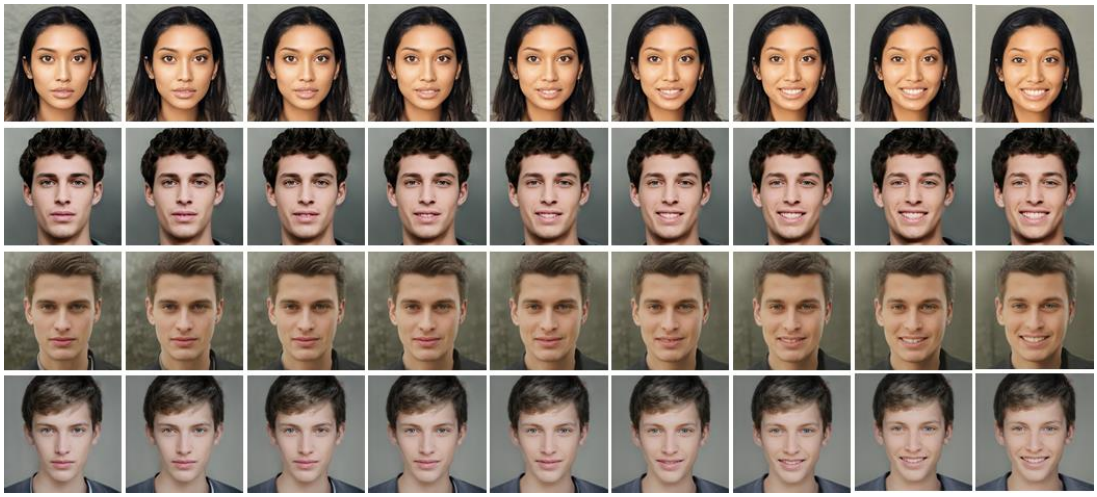

**Figure 1: Examples of facial expression animation generated automatically by our approach.** Each row demonstrates an animation from the neutral to happy expressions.

## Abstract

Automatic animation synthesis has attracted much attention from the community. As most existing methods take a small number of discrete expressions rather than continuous expressions, their integrity and reality of the facial expressions is often compromised. In addition, the easy manipulation with simple inputs and unsupervised processing, although being important to the automatic facial expression animation applications, was relatively less concerned. To address these issues, we propose an unsupervised continuous automatic facial expression animation approach through Action Units transfer in the latent space of Generative Adversarial Networks (GAN). The expression descriptor depicted with Action Units vector is transferred into the input image without the need of labelled pairs of images, their expressions and further network training. We also propose a new approach to quickly generate input image's latent code and cluster the boundaries of different Action Units attributes with their latent codes. Two latent code operators, vector addition and continuous interpolation, are leveraged for facial expression animation simulating align with the boundaries in the latent space. Experiments have shown that the proposed approach is effective on facial expression translation and animation synthesis.

**Keywords:** Facial expression, Facial animation, Action Units, Generative Adversarial Networks, Latent code encoding

## 1. Introduction

Facial expression animation is one of the powerful ways to insert the personalities into the computer generated characters [1]. It has created big impacts on applications in movie and other creative industries. The pioneering research of facial expression animation can be traced back to the work of Frederic I. Parke [2] in 1972. In the past decades, the facial expression animation research has been much conducted in computer graphics community [3], with well-known approaches proposed such as those based on mass-and-spring model [41] and

2D/3D morphing [42]. It has however received great attention in computer vision community in recent years because of the advance and success of deep learning techniques [4].

One of the challenging research topics on facial expression animation is to automate the process without manual intervention. Due to the diverseness of appearance from person to person, the same type of face expression such as *happiness* consists of an irregular structure, which makes its representation difficult. The development of CNN (Convolutional Neural Network) [5] and GAN (Generative Adversarial Networks) [6] has brought facial synthesis great promotion as they can dig deep image feature and generate high-reality fake images. Recently, the technology has been successfully applied to image-to-image translation to change the attributes of face, including gender, hair, age, and also facial expressions [7][8][9][10][11]. Most of the translation methods take the message-based approach [12] to describe facial behaviour and classify facial expressions into 7 basic emotions: anger, disgust, fear, happiness, sadness, surprise and contempt [13]. However, a discrete and low number of facial expression categories cannot fully depict the complex human expression of emotions.

According to the studies of psychologists, facial expressions are caused by a set of anatomically-motivated facial muscles. Paul Ekman and Wallace Friesen [14] proposed the Facial Action Coding System (FACS). In the system, facial expressions are decomposed into Action Units (AU) which are anatomically related to the contractions of specific facial muscles. Facial expressions can then be comprehensively described as a combination of different AUs. There are more than 30 AUs related to the contraction of specific facial muscles, and these AUs are diverse with different intensities. In fact, facial AUs can be combined into more than 7000 expressions. To this end, AU is more precise to describe facial expression than the message-based approaches. Fig.2 shows AU illustrations exampled by Ekman himself.

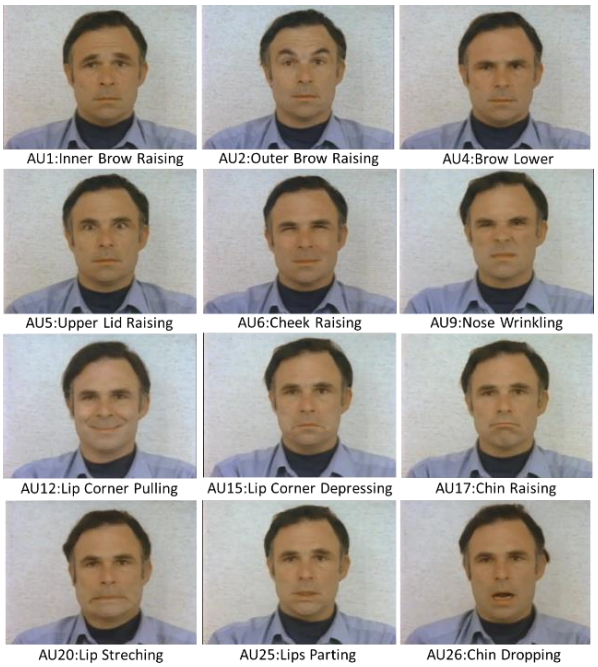


**Figure 2: Facial Action Units illustrations acted by Ekman [14].** 12 main AUs are listed with the essential muscle action explanations.

Taking advantages of FACS's powerful representation of expression and addressing the limitations of the message-based approach, we present a GAN based approach in this paper for automating the facial expression animation through anatomic muscle actions. Our approach is inspired by the success of StyleGAN [15], which generates facial images from latent code with unsupervised separation of high-level attributes. We synthesize AU properties in the latent space, by leveraging the latent code to embed attributes of face. Our approach is able to generate impressive continuous facial expressions, as shown in Fig. 1 Moreover, it is unsupervised and does not require the pairs of face images of the same person in different expressions or the expression annotation. Our contributions can be summarized as follows:

- A convolutional GAN based continuous facial expression animation synthesis framework is proposed to transfer AU description into facial images without requiring face alignment and extra network training. To measure the difference of AU attributes encoded in the latent space, we decouple the entangled attributes with subspace projection.
- A latent code encoding model is proposed to increase the convergence rate. We use the deep CNN network, VGG [40] to derive initial latent code. The VGG features are also leveraged to construct the loss function so that less iteration is required than the conventional pixel-wised distance methods.
- An approach based on two latent code operators is proposed for AUs transferring and continuous facial expression animation. Since linear subspaces align with different

AUs attributes emerging in the latent space, AUs boundaries are built for adding vector operator. Since the continuous facial expression changes according to linear latent codes, we leverage the interpolation operator based on AU transferring.

## 2. Related Work

Facial expressions play a major role in non-verbal communication, which carry two-thirds of human emotion [16]. In the span of time, significant efforts have been devoted to the development of automatic facial expression animation which can be treated as an unpaired image-to-image translation problem. They can be broadly grouped into two categories: the geometry-based and the vision-based.

### *2.1 Geometry-based methods*

Traditionally, the facial animation has been mainly addressed from a graphical perspective in which a 3D Morphable Model (3DMM) is first fitted to image and then re-rendered with a different facial expression [17]. The model is typically trained using spatially aligned 3D scans dataset [18][19][20]. These approaches usually involve three steps: image pre-processing, 3D modelling and expression fitting. As the facial attribute could be sensitive to affine transform and illumination variation, the pre-processing steps of face detection and alignment with landmarks [21][22] is required. 3D modelling then fits a 3D shape to match the input image with 3DMM method [23][24]. Finally, the expression coefficients are adjusted to perform animation followed with image or video rendering [25][26].

These geometry-based methods can provide high quality simulation of the input facial image, however they are unable to generate the expressions not shown by the image and unable to model parts not existing in the source image, such as the teeth when the mouth is closed. Moreover, it is difficult to acquire sufficient 3D face datasets for those methods requiring the network training.

### *2.2 Vision-based methods*

Generative Adversarial Networks (GAN) does a creative work on realistic fake image generation and opens a door to facial attribute editing [27][28][29][30]. GAN based facial attribute editing approaches have been proposed for unpaired image-to-image translation [31][8][9][10][11]. The attributes include gender, age, face color and facial expression [32][33][7]. However, these approaches depict expressions as a discrete emotion category thus fail to simulate continuous expression animation.

To tackle the limitation, an unsupervised continuous facial expression generating

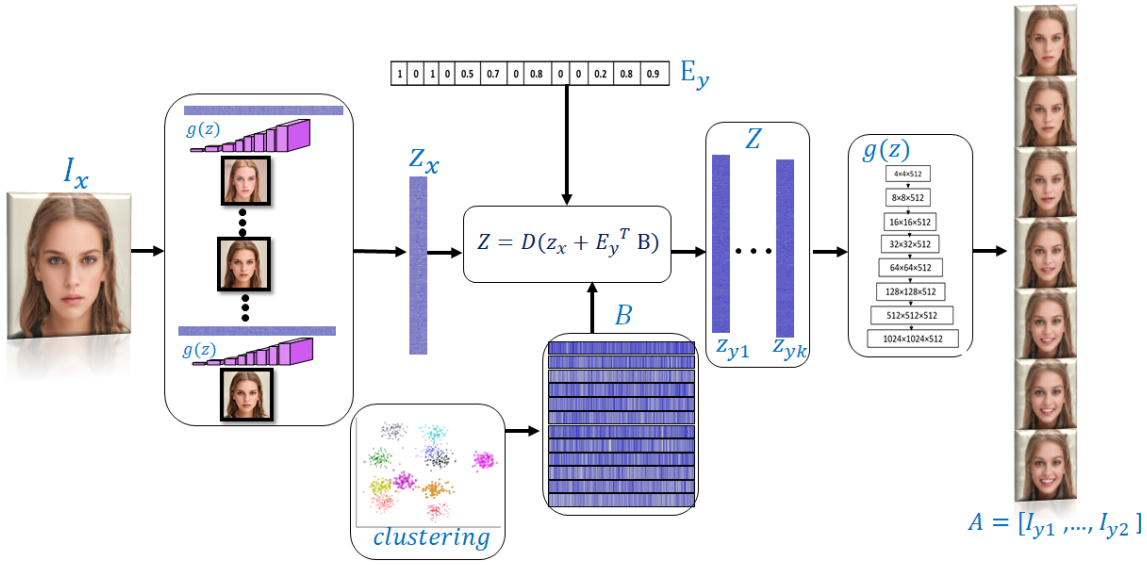


**Figure 3: The overview of our approach.** Firstly, the latent code $z_x$ of the facial image $I_x$ is encoded through iteration with the well-trained StyleGAN generator g(z). Secondly, the latent set $Z$ are generated in the latent space based on the specified facial expression descriptor $E_y$ and the AU boundaries set $B$. Lastly, the set of expressive latent codes is fed into the generator g(z) to produce the facial expression animation A.

approach based on GAN and AUs was proposed [4]. While it requires only a single face image, user must recognize the AU of the face's expression and provide its intensity, which is challenging for ordinary users.

To address the challenge, we exploit the operators of latent code in GAN network, which gives us two unique features compared with [4]. One is our framework does not require the AU occurrence vector of the input facial; the other is we use the well-trained StyleGAN generator to encode and decode the latent code without requiring facial images in dataset be labelled with AU vector for the training of generator and discriminator network.

# 3. Methodology

### *3.1 Facial expression transfer*

Automatic facial expression transfer is a very challenged task in facial animation, whilst it is much essential for delicate emotion understanding. The conventional seven basic expressions cannot depict all real actions. In this work, we propose a unified approach to animate facial expression animation through AUs transfer in the latent space. We embed an expression, which combined by AUs, into the facial image unsupervised. The pipeline of our approach is illustrated in Fig.3.

The input of the pipeline includes a facial image and an expression descriptor represented by a vector of AU intensities from 0 to 1. The expression descriptor is transmitted to the input facial image. The output of the pipeline is the facial animation corresponding with the expression descriptor. The addition operator of latent codes is leveraged for expression transfer, and the linear interpolation operator that aligns with different AUs latent boundaries is used to produce continuous facial expression animation.

We denote the input RGB facial image as $I_x \in \mathbb{R}^{W\times H\times 3}$ and the output RGB facial image as $I_y \in \mathbb{R}^{W\times H\times 3}$ , both with $W$ width and $H$ height. $I_y$ , which is from the same person as $I_x$, couples with the deterministic expression descriptor. $I_x$ of an arbitrary expression is provided by the user who is required to recognize the AU of the expression.

The pipeline consists of three main modules: latent code encoding, facial expression fitting and AU boundary building.

The conventional latent code encoding methods [34][35], which reverse the GAN generator process by iterating latent codes through to reduce the loss between the generated image and the ground truth image, are time-consuming and unstable because of the random initialization and pixel-wised loss function which drag down the convergence of the iteration. To address the issue, we initialize the latent code with the VGG full connect feature of the image. For the loss function we use the norm distance of VGG's full connected layer feature between the generated image and ground truth image. This improves the accuracy and efficiency of extracting image latent code. The latent code of $I_x$ is denoted as $z_x$, as illustrated in Fig. 3.

Facial expression fitting in latent space is the most important part in our pipeline. We bridge $z_x$ and $z_y$ with the expression descriptor $E_y = [e_1, \dots, e_m]$ under our fitting approach which is described in the next two sections, where $z_y$ is the latent code of the target facial image, $e_i \in [0,1]$ is the normalized AU boundary distances and $m$ is the number of AUs.

Based on the observation that linear subspace aligns with different facial AUs emerging in the latent space and continuous facial expression variation corresponding to linear latent code changing, we generate the facial expression animation through the linear interpolation of latent code. $D$ in Fig. 3 indicates the interpolation coefficients of the latent code transformation to simulate the intensity of the expressions. The output $A$ represents the continuous facial expressions contained in output animation which is transferred from the input facial image.

It has been widely observed that the facial image and its style can be disentangled in latent space [15]. There exists a hyperplane in the latent space between two semantic attributes [36]. Therefore, we can generate the latent codes of AU classification boundaries for facial expression animation. We define $B = [b_1, \dots, b_m]$ as the set of AUs boundaries in latent space, where $b_i$ is a unit normal vector to classify the hyperplane of the $i^{th}$ action unit. The boundary set specify the semantic range of

the AUs in latent space. The interpolation of expression in the boundary norm direction smooths the animation.

### *3.2 Latent code encoding*

The latent code encoding is an essential module in our approach which can be formulated as the function $g^{-1}: I \rightarrow z$, where $g^{-1}$ is an inverse process of GAN generating function, $I$ denotes facial image, and $z$ is the latent code of $I$ . Since it is impossible to deduce an inverse algebraic formula, we tackle it through a learning network. The learning network updates initial latent code depending on loss function under the feature distance from the generated image to the source image. The conventional methods [34][35] only consider the random initial latent code and pixel-wised image distance loss function. The convergence of the network is very slow and unstable. To tackle the issues, we introduce a deep convolution based loss function and initial latent code generating approach as illustrated in Fig.4.

Firstly, the last full connected layer with 1000-dimension feature map of VGG-16 is leveraged as the feature descriptor of images. The L1-norm distance between two feature descriptors of the generated image and the source image is exploited as the loss function:

$$\mathcal{L}(\mathrm{z}, \mathrm{I}) = L_1(Feature(g(z)), Feature(I)) \quad (1)$$

$L_1$ stands for the norm distance between the features. $Feature(\cdot)$ denotes the feature computed by the last full connected layer of VGG-16 network. $g(\cdot)$ represents the StyleGAN generator which operates on the latent code z to synthesize the generated image. This loss function has improved the convergence rate and more robust than pixel-to-pixel distance loss function, as demonstrated in our experiments.

Secondly, we bridge the source image with its initial latent code in Fig.4 through a deep convolution, by choosing a specific initial latent code instead of using a random value: $z_0 = v(I)$, where $z_0$ represents the initial latent code and $v(\cdot)$ denotes the trained convolution model. Therefore, the training dataset with pair data $\{z^m, I^m\}_{m=1}^{M}$ is needed, where $z^m$ is the $m^{th}$ latent code, $I^m$ is the $m^{th}$ image and $M$ is the number of data pairs. The random latent codes are fed into the generator, which synthesises images. The pair data are then used on model parameters training. The specified initial latent code has reduced the convergence time enormously.

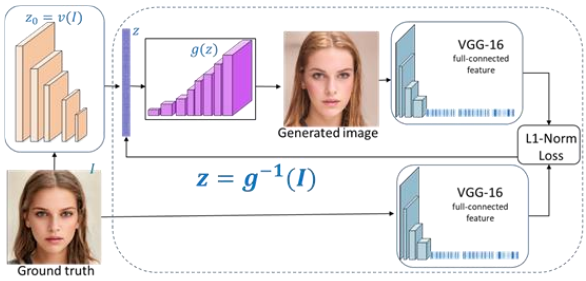


**Figure 4: The overview of latent code encoding.** The framework iterates its latent code with the change of the loss distance between features. The initial latent code comes from the well-trained deep convolutional network feature.

### *3.3 Facial action units transfer*

In this section we aim to find out the boundary hyperplane between different AUs in the latent space. We use Supported Vector Machine (SVM) to classify the latent codes from the neutral images to AU images and obtain the classification boundaries of each AU categories.

Through the AU latent codes boundary, we can attach the AU attribute to a facial image. For the boundary hyperplane $b$ between the latent codes with an AU characteristic and the neutral facial image latent codes, we define $\mathrm{e} = d(b, z)$ as the distance from the boundary to an arbitrary latent code $z$ as illustrated in Fig.5. The distance is along the normal of the boundary. $n$ is the unit normal of the boundary. Note that, e is not the real distance since it can be negative. e is positive when $z$ has the AU characteristic with the same direction with $n$ . And e will be negative when $z$ is to neutral which is on the opposite side of the boundary.

The vector addition operator can be used for expression transfer [31] although simply adding facial expression latent code to neutral facial latent code directly is barely satisfactory [31]. Thanks to the AU boundaries, the vector addition operator works effectively for AU transfer. To obtain a latent code with AU characteristic, we only need to add the distance from the boundary to the latent code:

$$f(z) = z + d(b, z) \cdot b \quad (2)$$

where $f(z)$ holds the same facial with $z$ except for the additional AU characteristic. The AU expression intensity is determined by $d$, and $b$ is the AU boundary vector. The latent code

transferring aligning with the boundary $b$ makes the expression transformation naturally and gives rise to a further smooth animation interpolation.

### *3.4 Facial animation synthesis*

Facial expression transfer is to learn a mapping $\mathcal{T}: I_x \xrightarrow{E_y} I_y$ from the facial image $I_x$ to $I_y$ with the expression $E_y$. $I_y$ is the output of $I_x$ with addition of the expression.

According to Rameen Abdal et al. [31], the style of image can be embedded in latent space for style transfer, including facial expressions. Hence the mapping from $I_x$ to $I_y$ can be translated as $\mathcal{F}: z_x \xrightarrow{E_y} z_y$ , where $z_x = g^{-1}(I_x)$ is the reverse GAN image generated process (detailed in the section 3.2), and $I_y = g(z_y)$ denotes the image generator function with the well-trained StyleGAN.

$E_y = [e_1, \dots, e_m]^T$ is the expression descriptor which indicates the vector of all AUs expression distances from the boundaries in the latent space, and $m$ is the number of AUs. The output latent code $z_y$ with expression can be obtained by input latent code $z_x$ under the following function$f_B$:

$$z_y = f_B(z_x) = z_x + {E_y}^T \mathrm{B} \qquad (3)$$

where $\mathrm{B} = [b_1, \dots, b_m]$ is the matrix made up by all boundaries.

The generator network can be used to generate the output expression facial image $I_y$ from the output latent code $z_y$, which is synthesized by the latent code of input image $I_x$ and the expression descriptor. We illustrate it with the following function:

$$I_y = g(f_B(g^{-1}(I_x))) = g(z_x + {E_y}^T \mathrm{B}) \qquad (4)$$

To generate facial expression animation, we apply linear interpolation to simulate the expression changing with different intensities. When two latent codes are linearly interpolated, the expression contained in them changes gradually and the appearance of the corresponding images changes continuously.

$$Z = h_D\big(f_B(z_x)\big) = D(z_x + {E_y}^T \mathrm{B}) \qquad (5)$$

where $\mathrm{D} = [\lambda_1, \dots, \lambda_k]^T$ indicates the linear coefficients of expression interpolation. $k$ is the number of generated expression latent codes. $Z = [z_1, \dots, z_k]$ contains all generated latent codes. D controls the frame frequency of the expression animation.

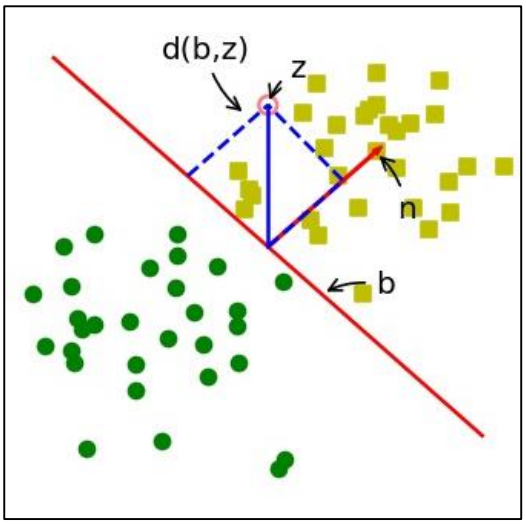


**Figure 5: The distribution of different AU latent codes.** The boundary $b$ is represented as red line, which is a separate hyperplane between the latent codes with an AU characteristic (yellow rectangles) and that of the neutral facial image (green circles). $n$ is the unit normal of the boundary represented as red arrow. $d(b, z)$ represents the distance from boundary to an arbitrary facial image with latent code $z$.

From the set of latent codes, we can generate the facial expression animation with pre-trained GAN network:

$$A = g(h_D\big(f_B(g^{-1}(I_x))\big)) \qquad (6)$$

Since the expression is combined with each AU boundary distance along boundary normal vector, the expression transfer path is as natural as the real expression animation.

## 4. Experiments

In this section we report and visualize the results of 3 experiments: facial image encoding, facial AU transfer and facial animation.

### *4.1 Experimental settings*

**Dataset:** The images are sampled from Denver Intensity of Spontaneous Facial Action database (DISFA) [37] which spontaneous emotion facial expressions were recorded while the subjects were watching YouTube videos. Twelve AUs were coded manually in DISFA. They are AU1, AU2, AU4, AU5, AU6, AU9, AU12, AU15, AU17, AU20, AU25, and AU26. There are only a few samples with intensity 5, despite it is the

highest intensity with the best AU representation, hence we sample the facial images with 4 and 5 AU intensity in DISFA.

**Network setting:** In our model, the network of StyleGAN is used twice. The first is for latent code encoding. The generator network [15] is applied to generate image from latent code which is updated under the loss. The VGG network is also used in this stage to obtain the initial latent code and to calculate loss together with the generator network. Both networks are well-trained with CelebA-HQ [38] dataset and ImageNet [39] dataset, so we do not need to take time to train them. The second time is in the last stage of our framework. The synthesized latent code is fed into the well-trained generator network to produce facial images with expression.

In StyleGAN, the initial latent space goes through a fully connected neutral network to the intermediate latent space which is the input of the generator. An important insight from [31] is that it is not easily possible to embed the expression attribute into initial latent space or intermediate latent space. Thus, the enhanced intermediate space becomes a good choice for our approach which is a concatenation of 18 different 512-dimensional intermediate latent vectors.

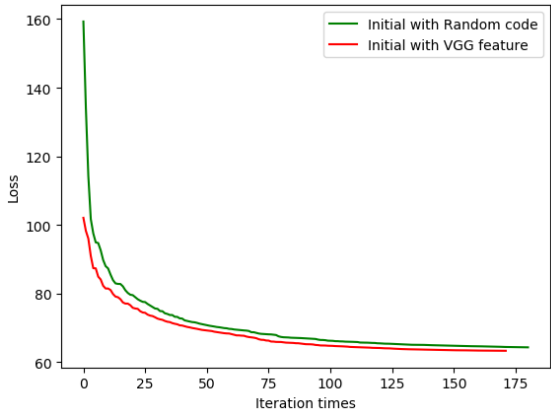


**Figure 6: The convergence of our latent code generating process.**

### *4.2 Encoding experiment*

In this experiment, we evaluate our approach's performance on latent code encoding, and compare it with the random initialization approach [34].

As showed in Fig.6, our approach has a smaller initial loss and faster convergence rate than the

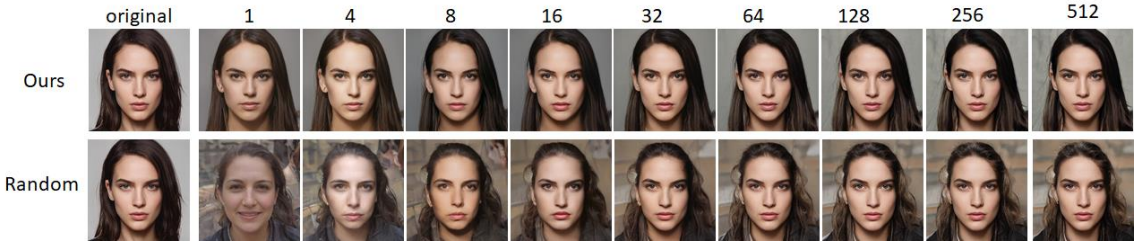


**Figure 7: The comparison of our latent code encoding result with that from the random initialization approach [34].** The number on the top of an image is the iteration times on generating the latent code for the image.

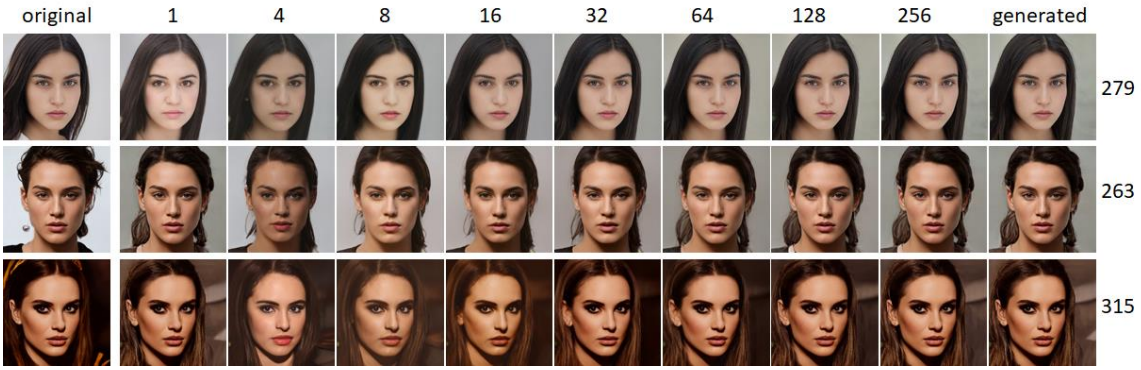


**Figure 8: The visualization of latent code encoding.** The first column is original images. The subsquent columns are the images generated by the initial latent code obtained by the VGG features to ensure quick convergence. The last column is the images generated with the final latent codes. On the tops and the right-most are the iteration times.

random initial code approach, thanks to the utilization of the VGG full connect feature to initialize the latent code. In the experiment we found that in most cases the iteration for our approach to reach the threshold was no more than $2^9 = 512$ times. In fact, as seen in Fig. 7, the generated images are very similar to the original images at the 16 times of iteration, which has demonstrated the quick convergence of our approach. On the contrary, the random initial code approach cannot converge even at 512 iterations. Fig. 8 shows some more encoding visualization results achieved using our approach. Again, the quick convergence is clearly demonstrated.

### *4.3 AU transfer and animation experiments*

In this section, we verify our animation approach through two experiments: single AU transfer interpolation and combined AU interpolation.

In Fig. 9, we demonstrate the AUs transfer animation. Each row is the animation images under the linear interpolation.

To experiment with different AUs, we chose AU4 and AU5 mainly happening on upper face, AU6 mainly happening on middle face, AU12 and AU25 mainly happening on lower face. It can be seen from Fig. 9 that our approach

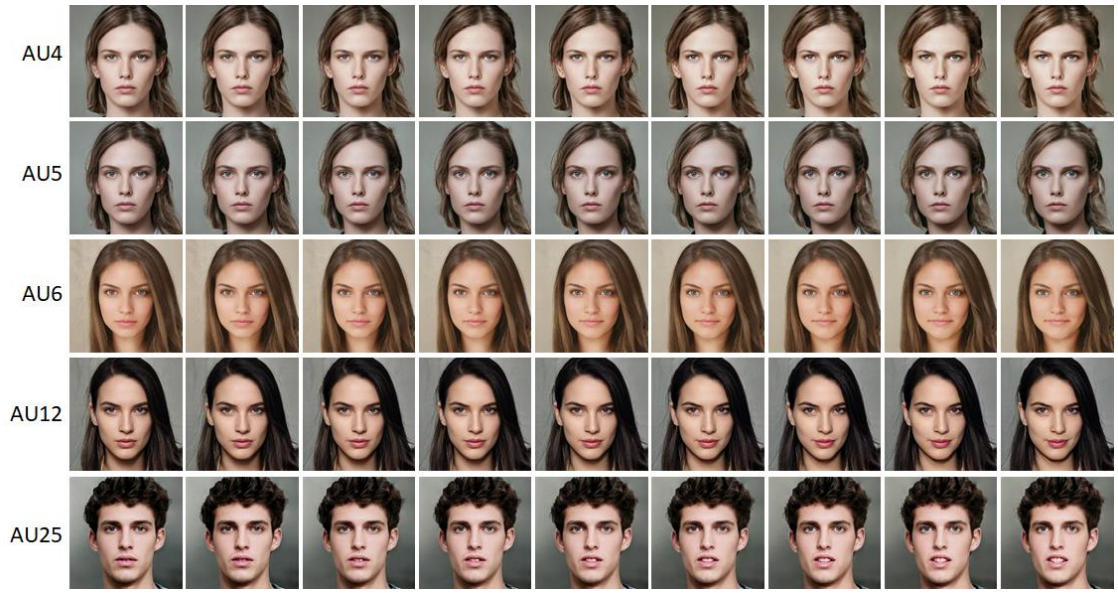


**Figure 9: The AU transfer animation.** Each row is the animation images with a specified AU transfer. AU4 indicates brown lower. AU5 indicates upper lid raising. AU6 indicates cheek raising. AU12 indicates lip corner pulling. AU25 indicates lips parting.


**Figure 10: Facial expressions interpolation results of the discrete facial expression with the specified AU combination.**

achieves better interpolation for AUs in AU4, AU12 and AU25, but less ideal for those in AU4 and AU6. We attribute it to the insufficient subject independent AU data in DISFA of these AUs. This is a part of limitations we will investigate in the future.

In Fig. 1, we demonstrate the compound AUs animations with the expression descriptor [0,0,0,0,1,0,1,0,0,0,1,0] to simulate the happy expression. It includes 12 AU occurrence descriptions which is similar to DISFA database [37] and those illustrated in Fig.2 by Ekman. In this expression descriptor, AU6 which indicates cheek raising is set as 1, and AU12 which indicates lip corner pulling is set as 1, and AU25 which indicates lips parting is set as 1, and the others are all set as 0 which means they have never happened on the animations.

Fig. 10 shows more animation results of the facial expressions. The results have demonstrated the effectiveness of our approach on generating high quality of facial expression animation with AU transfers.

## 5. Conclusion

In this paper, we proposed a framework to animate the facial expressions. Our framework is able to generate continuous facial expressions unsupervised, and also deal with convenient manipulation. The input of our approach is a facial image and an AU vector which transfers to the face in the input image. Our approach used the well-trained network thus saves time on training. The AU translation is done in the latent space instead of image space so that we can uncouple the semantic attributes containing AU feature in the latent space. The experiment results have demonstrated the high performance and effectiveness of our proposed approach.

## References


[1] Orvalho, Veronica & Orvalho, J. Character animation: Past, present and future. Business, Technological, and Social. Dimensions of Computer Games: Multidisciplinary Developments. 2011, 49-64.

[2] F. I. Parke, Computer Generated Animation of Faces. Proc. ACM annual conf., 1972.

[3] Alkawaz M H , Mohamad D , Rehman A , et al. Facial Animations: Future Research Directions & Challenges[J]. 3D Research, 2014, 5(2):12-91.

[4] 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, pages 835–851, 2018.

[5] Asifullah Khan, Anabia Sohail, Umme Zahoora, Aqsa Saeed Qureshi. A Survey of the Recent Architectures of Deep Convolutional Neural Networks. Artificial Intelligence Review. 2020, 1-62.

[6] Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Yoshua (2014). Generative Adversarial Networks (PDF). Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014). pp. 2672–2680.

[7] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multidomain image-to-image translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8789–8797, 2018.

[8] M. Liu, T. Breuel, and J. Kautz. Unsupervised image-to image translation networks. In Proceedings of the Neural Information Processing Systems Conference, pages 700–708, 2017.

[9] Ming-Yu Liu, Thomas Breuel , Jan Kautz. Unsupervised Image-to-Image Translation Networks. NIPS 2017.

[10] Andrés Romero, Pablo Arbeláez, Luc Van Gool, Radu Timofte. SMIT: Stochastic Multi-Label Image-to-Image Translation. ICCV Workshops, 2019.

[11] Wu, P., et al., RelGAN: Multi-Domain Image-to-Image Translation via Relative Attributes. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019: p. 5913-5921.

[12] Girard JM, Cohn JF, De la Torre F. Estimating smile intensity: A better way. Pattern Recognit Lett. 2015;66:13–21.

[13] P. Ekman and W. V. Friesen. Constants across cultures in the face and emotion. in Journal of Personality and Social Psychology, vol. 17, no. 2, pp. 124-129. 1971.

[14] P. Ekman and W. V. Friesen. Manual for the facial action coding system. Consulting Psychologists Press, 1978.

[15] Karras, T., S. Laine and T. Aila, A Style-Based Generator Architecture for Generative

Adversarial Networks. CVPR 2019.
[16] Birdwhistell, Ray. Kinesics and Context: Essays on Body Motion Communication. Philadelphia: University of Pennsylvania Press, 1970.
[17] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999.
[18] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3d facial expression database for visual computing. IEEE Transactions on Visualization and Computer Graphics, 20(3):413–425, 2014.
[19] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, P. Liu, and J. M. Girard, "Bp4d-spontaneous: a high-resolution spontaneous 3D dynamic facial expression database," IVC, vol. 32, no. 10, pp. 692–706, 2014.
[20] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale, "A highresolution 3D dynamic facial expression database," in FG, 2008, pp. 1–6.
[21] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in CVPR, 2013, pp. 532–539.
[22] Z. Mao, J. P. Siebert, W. P. Cockshott, and A. F. Ayoub, "Constructing dense correspondences to analyze 3D facial change," in ICPR, 2004, pp. 144–148.
[23] P. Huber, G. Hu, et al., A Multiresolution 3D Morphable Face Model and Fitting Framework, International Conference on Computer Vision Theory and Applications (VISAPP) 2016.
[24] Cao, Chen & Wu, Hongzhi & Weng, Yanlin & Shao, Tianjia & Zhou, Kun. (2016). Real-time Facial Animation with Image-based Dynamic Avatars. ACM Transactions on Graphics.
[25] Cao, Chen & Hou, Qiming & Zhou, Kun. Displaced Dynamic Expression Regression for Real-time Facial Tracking and Animation. ACM Transactions on Graphics. 2014.
[26] Geng, Z., C. Cao and S. Tulyakov, 3D Guided Fine-Grained Face Manipulation. CVPR 2019.
[27] Perarnau, G., van de Weijer, J., Raducanu, B., Alvarez, J.M.: Invertible conditional ǴANs for image editing. NIPS Workshop on Adversarial Training 2016.
[28] Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: ICLR . 2018.
[29] Shen, W., Liu, R.: Learning residual images for face attribute manipulation. In: CVPR, 2017.
[30] Li, M., Zuo, W., Zhang, D.: Deep identity-aware transfer of facial attributes. arXiv preprint arXiv:1610.05586 .2016.
[31] Abdal, R., Y. Qin and P. Wonka, Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space? CoRR, 2019.
[32] Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV, 2017.
[33] Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR, 2017.
[34] Bora, A., et al., Compressed Sensing Using Generative Models, in ICML'17. 2017. p. 537–546.
[35] ] Bojanowski, P., et al., Optimizing the Latent Space of Generative Networks. ICLR 2018.
[36] Shen, Y., et al., Interpreting the Latent Space of GANs for Semantic Face Editing. CVPR2020.
[37] S. M. Mavadati, M. H. Mahoor, K. Bartlett, P. Trinh, and J. F.Cohn. DISFA: A pontaneous facial action intensity database.TAC, 2013.
[38] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. CoRR, 2017.
[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database. IEEE Computer Vision and Pattern Recognition (CVPR), 2009.
[40] Simonyan, K. and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition. 2014.
[41] S. Platt, N. Badler, Animating facial expression. Computer Graphics, 1981, vol. 15(3) pp. 245-252
[42] Yu, H., Garrod, O.G., Schyns, P.G.: Perception-driven facial expression synthesis. Computers & Graphics 36(3) (2012)