International Journal of Computer Vision (2019) 127:743–762
https://doi.org/10.1007/s11263-019-01163-7

An Adversarial Neuro-Tensorial Approach for Learning Disentangled Representations

Mengjiao Wang[1] · Zhixin Shu[2] · Shiyang Cheng[1] · Yannis Panagakis[1,3] · Dimitris Samaras[2] · Stefanos Zafeiriou[1]

Received: 23 February 2018 / Accepted: 4 February 2019 / Published online: 16 February 2019

Abstract

Several factors contribute to the appearance of an object in a visual scene, including pose, illumination, and deformation, among others. Each factor accounts for a source of variability in the data, while the multiplicative interactions of these factors emulate the entangled variability, giving rise to the rich structure of visual object appearance. Disentangling such unobserved factors from visual data is a challenging task, especially when the data have been captured in uncontrolled recording conditions (also referred to as "in-the-wild") and label information is not available. In this paper, we propose a pseudo-supervised deep learning method for disentangling multiple latent factors of variation in face images captured in-the-wild. To this end, we propose a deep latent variable model, where the multiplicative interactions of multiple latent factors of variation are explicitly modelled by means of multilinear (tensor) structure. We demonstrate that the proposed approach indeed learns disentangled representations of facial expressions and pose, which can be used in various applications, including face editing, as well as 3D face reconstruction and classification of facial expression, identity and pose.

Keywords Adversarial autoencoder · Disentangled representation · Tensor decomposition

1 Introduction

The appearance of visual objects is significantly affected by multiple factors of variability such as, for example, pose, illumination, identity, and expression in case of faces. Each factor accounts for a source of variability in the data, while their complex interactions give rise to the observed entangled variability. Discovering the modes of variation, or in other words disentangling the latent factors of variations in visual data, is a very important problem in the intersection of statistics, machine learning, and computer vision.

Factor analysis (Fabrigar and Wegener 2011) and the closely related Principal Component Analysis (PCA) (Hotelling 1933) are probably the most popular statistical methods that find a single mode of variation explaining the data. Nevertheless, visual appearance (e.g., facial appearance) is affected by several modes of variations. Hence, methods such as PCA are not able to identify such multiple factors of variation. For example, when PCA is applied to facial images, the first principal component captures both pose and expressions variations.

An early approach for learning different modes of variation in the data is TensorFaces (Vasilescu and Terzopoulos 2002). In particular, TensorFaces is a strictly supervised method as it not only requires the facial data to be labelled (e.g., in terms of expression, identity, illumination etc.) but the data tensor must also contain all samples in all different variations. This is the primary reason that the use of such tensor decompositions is still limited to databases that have been captured in a strictly controlled environment, such as the Weizmann face database (Vasilescu and Terzopoulos 2002).

Recent unsupervised tensor decompositions methods (Tang et al. 2013; Wang et al. 2017b) automatically discover the modes of variation in unlabelled data. In particular, the most recent one (Wang et al. 2017b) assumes that the original visual data have been produced by a hidden multilinear structure and the aim of the unsupervised tensor decomposition is to discover both the underlying multilinear structure, as well as the corresponding weights (coefficients) that best explain

Communicated by Dr. Chellappa, Dr. Liu, Dr. Kim, Dr. Torre and Dr. Loy.

✉ Mengjiao Wang
m.wang15@imperial.ac.uk

[1] Imperial College London, London, UK

[2] Stony Brook University, Stony Brook, USA

[3] Middlesex University, London, UK

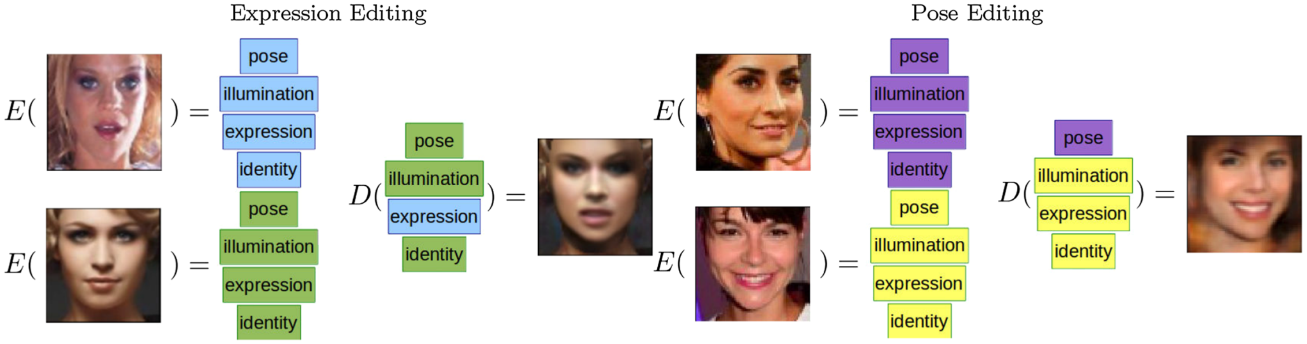

Fig. 1 Given a single in-the-wild image, our network learns disentangled representations for pose, illumination, expression and identity. Using these representations, we are able to manipulate the image and edit the pose or expression

the data. Special instances of the unsupervised tensor decomposition are the Shape-from-Shading (SfS) decompositions in Kemelmacher-Shlizerman (2013), Snape et al. (2015) and the multilinear decompositions for 3D face description in Wang et al. (2017b). In Wang et al. (2017b), it is shown that the method indeed can be used to learn representations where many modes of variation have been disentangled (e.g., identity, expression and illumination etc.). Nevertheless, the method in Wang et al. (2017b) is not able to find pose variations and bypasses this problem by applying it to faces which have been frontalised by applying a warping function [e.g., piece-wise affine warping (Matthews and Baker 2004)].

Another promising line of research for discovering latent representations is unsupervised Deep Neural Networks (DNNs). Unsupervised DNNs architectures include the Auto-Encoders (AE) (Bengio et al. 2013), as well as the Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) or adversarial versions of AE, e.g., the Adversarial Auto-Encoders (AAE) (Makhzani et al. 2015). Even though GANs, as well as AAEs, provide very elegant frameworks for discovering powerful low-dimensional embeddings without having to align the faces, due to the complexity of the networks, unavoidably all modes of variation are multiplexed in the latent-representation. Only with the use of labels it is possible to model/learn the manifold over the latent representation, usually as a post-processing step (Shu et al. 2017).

In this paper, we show that it is possible to learn a disentangled representation of the human face captured in arbitrary recording conditions in an pseudo-supervised manner[1] by imposing a multilinear structure on the latent representation of an AAE (Shu et al. 2017). To the best of our knowledge, this is the first time that unsupervised tensor decompositions have been combined with DNNs for learning disentangled representations. We demonstrate the power of the proposed approach by showing expression/pose transfer using only the latent variable that is related to expression/pose. We also demonstrate that the disentangled low-dimensional embeddings are useful for many other applications, such as facial expression, pose, and identity recognition and clustering. An example of the proposed approach is given in Fig. 1. In particular, the left pair of images have been decomposed, using the encoder of the proposed neural network $E(\cdot)$, into many different latent representations including latent representations for pose, illumination, identity and expression. Since our framework has learned a disentangled representation we can easily transfer the expression by only changing the latent variable related to expression and passing the latent vector into the decoder of our neural network $D(\cdot)$. Similarly, we can transfer the pose merely by changing the latent variable related to pose.

2 Related Work

Learning disentangled representations that explain multiple factors of variation in the data as disjoint latent dimensions is desirable in several machine learning, computer vision, and graphics tasks.

Indeed, bilinear factor analysis models (Tenenbaum and Freeman 2000) have been employed for disentangling two factors of variation (e.g., head pose and facial identity) in the data. Identity, expression, pose, and illumination variations are disentangled in Vasilescu and Terzopoulos (2002) by applying Tucker decomposition [also known as multilinear Singular Value Decomposition (SVD) (De Lathauwer et al. 2000)] into a carefully constructed tensor through label information. Interestingly, the modes of variation in well aligned images can be recovered via a multilinear matrix factorization (Wang et al. 2017b) without any supervision. However, inference in Wang et al. (2017b) might be ill-posed.

More recently, both supervised and unsupervised deep learning methods have been developed for disentangled representations learning. Transforming auto-encoders (Hinton

[1] Our methodology uses the information produced by an automatic 3D face fitting procedure (Booth et al. 2017) but it does not make use of any labels in the training set.

et al. 2011) is among the earliest methods for disentangling latent factors by means of auto-encoder capsules. In Desjardins et al. (2012) hidden factors of variation are disentangled via inference in a variant of the restricted Boltzmann machine. Disentangled representations of input images are obtained by the hidden layers of deep networks in Cheung et al. (2014) and through a higher-order Boltzmann machine in Reed et al. (2014). The Deep Convolutional Inverse Graphics Network (Kulkarni et al. 2015) learns a representation that is disentangled with respect to transformations such as out-of-plane rotations and lighting variations. Methods in Chen et al. (2016), Mathieu et al. (2016), Wang et al. (2017a), Tewari et al. (2017) and Tran et al. (2017) extract disentangled and interpretable visual representations by employing adversarial training. Recent works in face modeling (Tewari et al. 2018; Tran and Liu 2018) also employ self-supervision or pseudo-supervision to learn 3D Morphable Models from images. They rely on the use of a 3D to 2D image rendering layer to separate shape and texture. Contrarily to Tewari et al. (2018), Tran and Liu (2018) the proposed network does not render the 3D shape into a 2D image. Learning the components of a 3D morphable model is an additional advantage of the pseudo-supervision employed. The method in Shu et al. (2017) disentangles the latent representations of illumination, surface normals, and albedo of face images using an image rendering pipeline. Trained with pseudo-supervision, Shu et al. (2017) undertakes multiple image editing tasks by manipulating the relevant latent representations. Nonetheless, this editing approach still requires expression labelling, as well as sufficient sampling of a specific expression.

Here, the proposed network is able to edit the expression of a face image given another single in-the-wild face image of arbitrary expression. Furthermore, we are able to edit the pose of a face in the image which is not possible in Shu et al. (2017).

3 Proposed Method

In this section, we will introduce the main multilinear models used to describe three different image modalities, namely texture, 3D shape and 3D surface normals. To this end, we assume that for each different modality there is a different core tensor but all modalities share the same latent representation of weights regarding identity and expression. During training all the core tensors inside the network are randomly initialised and learnt end-to-end. In the following, we assume that we have a set of n facial images (e.g., in the training batch) and their corresponding 3D facial shape, as well as their normals per pixel (the 3D shape and normals have been produced by fitting a 3D model on the 2D image, e.g., Booth et al. 2017).

3.1 Facial Texture

The main assumption here follows from Wang et al. (2017b). That is, the rich structure of visual data is a result of multiplicative interactions of hidden (latent) factors and hence the underlying multilinear structure, as well as the corresponding weights (coefficients) that best explain the data can be recovered using the unsupervised tensor decomposition (Wang et al. 2017b). Indeed, following (Wang et al. 2017b), disentangled representations can be learnt (e.g., identity, expression, and illumination, etc.) from frontalised facial images. The frontalisation process is performed by applying a piecewise affine transform using the sparse shape recovered by a face alignment process. Inevitably, this process suffers from warping artifacts. Therefore, rather than applying any warping process, we perform the multilinear decomposition only on near frontal faces, which can be automatically detected during the 3D face fitting stage. In particular, assuming a near frontal facial image rasterised in a vector $\boldsymbol{x}_f \in \mathbb{R}^{k_x \times 1}$, given a core tensor $\mathcal{Q} \in \mathbb{R}^{k_x \times k_l \times k_{exp} \times k_{id}}$,[2] this can be decomposed as

$$\boldsymbol{x}_f = \mathcal{Q} \times_2 z_l \times_3 z_{exp} \times_4 z_{id}, \tag{1}$$

where $z_l \in \mathbb{R}^{k_l}$, $z_{exp} \in \mathbb{R}^{k_{exp}}$ and $z_{id} \in \mathbb{R}^{k_{id}}$ are the weights that correspond to illumination, expression and identity respectively. The equivalent form in case that we have a number of images in the batch stacked in the columns of a matrix $\mathbf{X}_f \in \mathbb{R}^{k_x \times n}$ is

$$\boldsymbol{X}_f = \boldsymbol{Q}_{(1)}(\boldsymbol{Z}_l \odot \boldsymbol{Z}_{exp} \odot \boldsymbol{Z}_{id}), \tag{2}$$

where $\boldsymbol{Q}_{(1)}$ is a mode-1 matricisation of tensor $\mathcal{Q}$ and $\boldsymbol{Z}_l$, $\boldsymbol{Z}_{exp}$ and $\boldsymbol{Z}_{id}$ are the corresponding matrices that gather the weights of the decomposition for all images in the batch. That is, $\boldsymbol{Z}_{exp} \in \mathbb{R}^{k_{exp} \times n}$ stacks the n latent variables of expressions of the images, $\boldsymbol{Z}_{id} \in \mathbb{R}^{k_{id} \times n}$ stacks the n latent variables of identity and $\boldsymbol{Z}_l \in \mathbb{R}^{k_l \times n}$ stacks the n latent variables of illumination.

3.2 3D Facial Shape

It is quite common to use a bilinear model for disentangling identity and expression in 3D facial shape (Bolkart and

[2] Tensors notation: Tensors (i.e., multidimensional arrays) are and denoted by calligraphic letters, e.g., $\boldsymbol{\mathcal{X}}$. The *mode-m matricisation* of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$ maps $\boldsymbol{\mathcal{X}}$ to a matrix $\mathbf{X}_{(m)} \in \mathbb{R}^{I_m \times \bar{I}_m}$. The *mode-m vector product* of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_M}$ with a vector $\mathbf{x} \in \mathbb{R}^{I_m}$, denoted by $\boldsymbol{\mathcal{X}} \times_n \mathbf{x} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N}$.
The *Kronecker product* is denoted by $\otimes$ and the *Khatri-Rao* (i.e., column-wise Kronecker product) product is denoted by $\odot$. More details on tensors and multilinear operators can be found in Kolda and Bader (2008).

Wuhrer 2016). Hence, for 3D shape we assume that there is a different core tensor $\mathcal{B} \in \mathbb{R}^{k_{3d} \times k_{exp} \times k_{id}}$ and each 3D facial shape $\boldsymbol{x}_{3d} \in \mathbb{R}^{k_{3d}}$ can be decomposed as:

$$\boldsymbol{x}_{3d} = \mathcal{B} \times_2 z_{exp} \times_3 z_{id}, \tag{3}$$

where z_{exp} and z_{id} are exactly the same weights as in the texture decomposition (2). The tensor decomposition for the n images in the batch is therefore written as as

$$X_{3d} = \boldsymbol{B}_{(1)}(\boldsymbol{Z}_{exp} \odot \boldsymbol{Z}_{id}), \tag{4}$$

where $\boldsymbol{B}_{(1)}$ is a mode-1 matricization of tensor $\mathcal{B}$.

3.3 Facial Normals

The tensor decomposition we opted to use for facial normals was exactly the same as the texture, hence we can use the same core tensor and weights. The difference is that since facial normals do not depend on illumination parameters (assuming a Lambertian illumination model), we just need to replace the illumination weights with a constant.[3] Thus, the decomposition for normals can be written as

$$\boldsymbol{X}_N = \boldsymbol{Q}_{(1)} \left(\frac{1}{k_l} \mathbb{1} \odot \boldsymbol{Z}_{exp} \odot \boldsymbol{Z}_{id} \right), \tag{5}$$

where $\mathbb{1}$ is a matrix of ones.

3.4 3D Facial Pose

Finally, we define another latent variable regarding 3D pose. This latent variable $z_p \in \mathbb{R}^9$ represents a 3D rotation. We denote by $\boldsymbol{x}^i \in \mathbb{R}^{k_x}$ an image at index i. The indexing is denoted in the following by the superscript. The corresponding z_p^i can be reshaped into a rotation matrix $\boldsymbol{R}^i \in \mathbb{R}^{3\times3}$. As proposed in Worrall et al. (2017), we apply this rotation to the feature of the image $\boldsymbol{x}^i$ created by 2-way synthesis (explained in Sect. 3.5). This feature vector is the i-th column of the feature matrix resulting from the 2-way synthesis $(\boldsymbol{Z}_{exp} \odot \boldsymbol{Z}_{id}) \in \mathbb{R}^{k_{exp}k_{id} \times n}$. We denote this feature vector corresponding to a single image as $(\boldsymbol{Z}_{exp} \odot \boldsymbol{Z}_{id})^i \in \mathbb{R}^{k_{exp}k_{id}}$. Next $(\boldsymbol{Z}_{exp} \odot \boldsymbol{Z}_{id})^i$ is reshaped into a $3 \times \frac{k_{exp}k_{id}}{3}$ matrix and left-multiplied by $\boldsymbol{R}^i$. After another round of vectorisation, the resulting feature $\in \mathbb{R}^{k_{exp}k_{id}}$ becomes the input of the decoders for normal and albedo. This transformation from feature vector $(\boldsymbol{Z}_{exp} \odot \boldsymbol{Z}_{id})^i$ to the rotated feature is called **rotation**.

3.5 Network Architecture

We incorporate the structure imposed by Eqs. (2), (4) and (5) into an auto-encoder network, see Fig. 2. For some matrices $\boldsymbol{Y}_i \in \mathbb{R}^{k_{yi} \times n}$, we refer to the operation $\boldsymbol{Y}_1 \odot \boldsymbol{Y}_2 \in \mathbb{R}^{k_{y1}k_{y2} \times n}$ as **2-way synthesis** and $\boldsymbol{Y}_1 \odot \boldsymbol{Y}_2 \odot \boldsymbol{Y}_3 \in \mathbb{R}^{k_{y1}k_{y2}k_{y3} \times n}$ as **3-way synthesis**. The multiplication of a feature matrix by $\boldsymbol{B}_{(1)}$ or $\boldsymbol{Q}_{(1)}$, mode-1 matricisations of tensors $\mathcal{B}$ and $\mathcal{Q}$, is referred to as **projection** and can be represented by an unbiased fully-connected layer.

Our network follows the architecture of Shu et al. (2017). The encoder E receives an input image $\boldsymbol{x}$ and the convolutional encoder stack first encodes it into z_i, an intermediate latent variable vector of size 128 × 1. z_i is then transformed into latent codes for background z_b, mask z_m, illumination z_l, pose z_p, identity z_{id} and expression z_{exp} via fully-connected layers.

$$E(\boldsymbol{x}) = [z_b, z_m, z_l, z_p, z_{id}, z_{exp}]^T. \tag{6}$$

The decoder D takes in the latent codes as input. z_b and z_m (128 × 1 vectors) are directly passed into convolutional decoder stacks to estimate background and face mask respectively. The remaining latent variables follow 3 streams:

1. z_{exp} (15 × 1 vector) and z_{id} (80 × 1 vector) are joined by 2-way synthesis and projection to estimate facial shape $\hat{\boldsymbol{x}_{3d}}$.
2. The result of 2-way synthesis of z_{exp} and z_{id} is rotated using z_p. The rotated feature is passed into 2 different convolutional decoder stacks: one for normal estimation and another for albedo. Using the estimated normal map, albedo, illumination component z_l, mask and background, we render a reconstructed image $\hat{\boldsymbol{x}}$.
3. z_{exp}, z_{id} and z_l are combined by a 3-way synthesis and projection to estimate frontal normal map and a frontal reconstruction of the image.

Streams 1 and 3 drive the disentangling of expression and identity components, while stream 2 focuses on the reconstruction of the image by adding the pose components. The decoder D then outputs the reconstructed image from the latent codes.

$$D(z_b, z_m, z_l, z_p, z_{id}, z_{exp}) = \hat{\boldsymbol{x}}. \tag{7}$$

Our input images are aligned and cropped facial images from the CelebA database (Liu et al. 2015) of size 64 × 64, so $k_x = 3 \times 64 \times 64$. $k_{3d} = 3 \times 9375$, $k_l = 9$, $k_{id} = 80$ and $k_{exp} = 15$. More details on the network such as the convolutional encoder stacks and decoder stacks can be found in the supplementary material.

[3] This is also the way that normals are computed in Wang et al. (2017b) up to a scaling factor

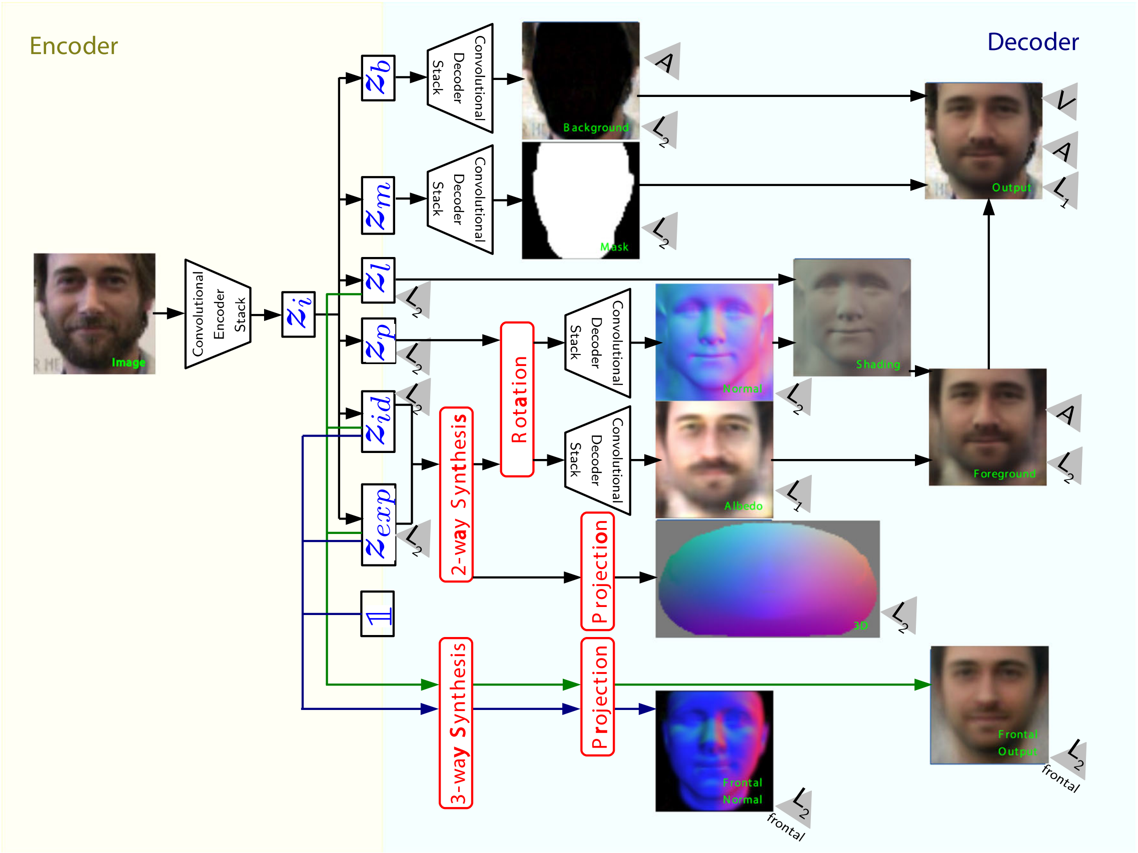

Fig. 2 Our network is an end-to-end trained auto-encoder. The encoder E extracts latent variables corresponding to illumination, pose, expression and identity from the input image $\boldsymbol{x}$. These latent variables are then fed into the decoder D to reconstruct the image. We impose a multilinear structure and enforce the disentangling of variations. The grey triangles represent the losses: adversarial loss A, verification loss V, L_1 and L_2 losses

3.6 Training

We use in-the-wild face images for training. Hence, we only have access to the image itself ($\boldsymbol{x}$) while ground truth labelling for pose, illumination, normal, albedo, expression, identity or 3D shape is unavailable. The main loss function is the reconstruction loss of the image x:

$$E_x = E_{recon} + \lambda_{adv} E_{adv} + \lambda_{veri} E_{veri}, \tag{8}$$

where $\hat{\boldsymbol{x}}$ is the reconstructed image, $E_{recon} = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_1$ is the reconstruction loss, λ_{adv} and λ_{adv} are regularisation weights, E_{adv} represents the adversarial loss and E_{veri} the verification loss. We use the pre-trained verification network $\mathcal{V}$ (Wu et al. 2015) to find face embeddings of our images $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$. As both images are supposed to represent the same person, we minimise the cosine distance between the embeddings: $E_{veri} = 1 - cos(\mathcal{V}(\boldsymbol{x}), \mathcal{V}(\hat{\boldsymbol{x}}))$. Simultaneously, a discriminative network $\mathcal{D}$ is trained to distinguish between the generated and real images (Goodfellow et al. 2014). We incorporate the discriminative information by following the auto-encoder loss distribution matching approach of Berthelot et al. (2017). The discriminative network $\mathcal{D}$ is itself an auto-encoder trying to reconstruct the input image $\boldsymbol{x}$ so the adversarial loss is $E_{adv} = \|\hat{\boldsymbol{x}} - \mathcal{D}(\hat{\boldsymbol{x}})\|_1$. $\mathcal{D}$ is trained to minimise $\|\boldsymbol{x} - \mathcal{D}(\boldsymbol{x})\|_1 - k_t\|\hat{\boldsymbol{x}} - \mathcal{D}(\hat{\boldsymbol{x}})\|_1$.

As fully unsupervised training often results in semantically meaningless latent representations, Shu et al. (2017) proposed to train with pseudo ground truth values for normals, lighting and 3D facial shape. We adopt here this technique and introduce further pseudo ground truth values for pose $\hat{\boldsymbol{x}_p}$, expression $\hat{\boldsymbol{x}_{exp}}$ and identity $\hat{\boldsymbol{x}_{id}}$. $\hat{\boldsymbol{x}_p}$, $\hat{\boldsymbol{x}_{exp}}$ and $\hat{\boldsymbol{x}_{id}}$ are obtained by fitting coarse face geometry to every image in the training set using a 3D Morphable Model (Booth et al. 2017). We incorporated the constraints used in Shu et al. (2017) for illumination, normals and albedo. Hence, the following new objectives are introduced:

$$E_p = \|z_p - \hat{\boldsymbol{x}_p}\|_2^2, \tag{9}$$

where $\hat{\boldsymbol{x}_p}$ is a 3D camera rotation matrix.

$$E_{exp} = \| fc(z_{exp}) - \boldsymbol{x}_{\hat{exp}} \|_2^2, \tag{10}$$

where fc(·) is a fully-connected layer and $\boldsymbol{x}_{\hat{exp}} \in \mathbb{R}^{28}$ is a pseudo ground truth vector representing 3DMM expression components of the image $\boldsymbol{x}$.

$$E_{id} = \| fc(z_{id}) - \boldsymbol{x}_{\hat{id}} \|_2^2 \tag{11}$$

where fc(·) is a fully-connected layer and $\boldsymbol{x}_{\hat{id}} \in \mathbb{R}^{157}$ is a pseudo ground truth vector representing 3DMM identity components of the image $\boldsymbol{x}$.

3.6.1 Multilinear Losses

Directly applying the above losses as constraints to the latent variables does not result in a well-disentangled representation. To achieve a better performance, we impose a tensor structure on the image using the following losses:

$$E_{3d} = \| \boldsymbol{x}_{\hat{3d}} - \mathcal{B} \times_2 z_{exp} \times_3 z_{id} \|_2^2, \tag{12}$$

where $\boldsymbol{x}_{\hat{3d}}$ is the 3D facial shape of the fitted model.

$$E_f = \| \boldsymbol{x}_f - \mathcal{Q} \times_2 z_l \times_3 z_{exp} \times_4 z_{id}) \|_2^2, \tag{13}$$

where $\boldsymbol{x}_f$ is a semi-frontal face image. During training, E_f is only applied on near-frontal face images filtered using $\boldsymbol{x}_{\hat{p}}$.

$$E_n = \| \boldsymbol{n}_{\hat{f}} - \mathcal{Q} \times_2 \frac{1}{k_l} \mathbf{1} \times_3 z_{exp} \times_4 z_{id}) \|_2^2 \tag{14}$$

where $\boldsymbol{n}_{\hat{f}}$ is a near frontal normal map. During training, the loss E_n is only applied on near frontal normal maps.

The model is trained end-to-end by applying gradient descent to batches of images, where Eqs. (12), (13) and (14) are written in the following general form:

$$E = \| \boldsymbol{X} - \boldsymbol{B}_{(1)} (\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(M)}) \|_F^2, \tag{15}$$

where M is the number of modes of variations, $\boldsymbol{X} \in \mathbb{R}^{k \times n}$ is a data matrix, $\boldsymbol{B}_{(1)}$ is the mode-1 matricisation of a tensor $\mathcal{B}$ and $\boldsymbol{Z}^{(i)} \in \mathbb{R}^{k_{zi} \times n}$ are the latent variables matrices.

The partial derivative of (15) with respect to the latent variable $\boldsymbol{Z}^{(i)}$ are computed as follows: Let $\hat{\boldsymbol{x}} = vec(\boldsymbol{X})$ be the vectorised $\boldsymbol{X}$, $\hat{z}^{(i)} = vec(\boldsymbol{Z}^{(i)})$ be the vectorised $\boldsymbol{Z}^{(i)}$, $\boldsymbol{Z}^{(\widehat{i-1})} = \boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(i-1)}$ and $\boldsymbol{Z}^{(\widehat{i+1})} = \boldsymbol{Z}^{(i+1)} \odot \cdots \odot \boldsymbol{Z}^{(M)}$, then (15) is equivalent with:

$$\begin{aligned} &\| \hat{\boldsymbol{x}} - (\boldsymbol{I} \otimes \boldsymbol{B}_{(1)}) vec(\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(M)}) \|_F^2 \\ &\quad = \| \hat{\boldsymbol{x}} - (\boldsymbol{I} \otimes \boldsymbol{B}_{(1)}) (\boldsymbol{I} \odot \boldsymbol{Z}^{(\widehat{i-1})}) \otimes \boldsymbol{I} \\ &\qquad \cdot \boldsymbol{I} \odot (\boldsymbol{Z}^{(\widehat{i+1})} (\boldsymbol{I} \otimes \nVdash)) \cdot \boldsymbol{z}^{(\hat{i})} \|_2^2 \end{aligned} \tag{16}$$

Consequently the partial derivative of (15) with respect to $\boldsymbol{Z}^{(i)}$ is obtained by matricising the partial derivative of (16) with respect to $\boldsymbol{Z}^{(i)}$. The derivation details are in the subsequent section.

3.6.2 Derivation Details

The model is trained end-to-end by applying gradient descent to batches of images, where (12), (13) and (14) are written in the following general form:

$$E = \| \boldsymbol{X} - \boldsymbol{B}_{(1)} (\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(M)}) \|_F^2, \tag{15}$$

where $\boldsymbol{X} \in \mathbb{R}^{k \times n}$ is a data matrix, $\boldsymbol{B}_{(1)}$ is the mode-1 matricisation of a tensor $\mathcal{B}$ and $\boldsymbol{Z}^{(i)} \in \mathbb{R}^{k_{zi} \times n}$ are the latent variables matrices.

The partial derivative of (15) with respect to the latent variable $\boldsymbol{Z}^{(i)}$ are computed as follows: Let $\hat{\boldsymbol{x}} = vec(\boldsymbol{X})$ be a vectorisation of $\boldsymbol{X}$, then (15) is equivalent with:

$$\begin{aligned} &\| \boldsymbol{X} - \boldsymbol{B}_{(1)} (\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(M)}) \|_F^2 \\ &\quad = \| vec(\boldsymbol{X} - \boldsymbol{B}_{(1)} (\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(M)})) \|_2^2 \\ &\quad = \| \hat{\boldsymbol{x}} - vec(\boldsymbol{B}_{(1)} (\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(M)})) \|_2^2, \end{aligned} \tag{17}$$

as both the Frobenius norm and the L_2 norm are the sum of all elements squared.

$$\begin{aligned} &\| \hat{\boldsymbol{x}} - vec(\boldsymbol{B}_{(1)} (\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(M)})) \|_2^2 \\ &\quad = \| \hat{\boldsymbol{x}} - (\boldsymbol{I} \otimes \boldsymbol{B}_{(1)}) vec(\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(M)}) \|_2^2, \end{aligned} \tag{18}$$

as the property $vec(\boldsymbol{B}\boldsymbol{Z}) = (\boldsymbol{I} \otimes \boldsymbol{B}) vec(\boldsymbol{Z})$ holds Neudecker (1969).

Using $vec(\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)}) = (\boldsymbol{I} \odot \boldsymbol{Z}^{(1)}) \otimes \boldsymbol{I} \cdot vec(\boldsymbol{Z}^{(2)})$ (Roemer 2012) and let $\boldsymbol{Z}^{(\widehat{i-1})} = \boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(i-1)}$ and $\boldsymbol{Z}^{(\hat{i})} = \boldsymbol{Z}^{(i)} \odot \cdots \odot \boldsymbol{Z}^{(M)}$ the following holds:

$$\begin{aligned} &\| \hat{\boldsymbol{x}} - (\boldsymbol{I} \otimes \boldsymbol{B}_{(1)}) vec(\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)} \odot \cdots \odot \boldsymbol{Z}^{(M)}) \|_2^2 \\ &\quad = \| \hat{\boldsymbol{x}} - (\boldsymbol{I} \otimes \boldsymbol{B}_{(1)}) (\boldsymbol{I} \odot \boldsymbol{Z}^{(\widehat{i-1})}) \otimes \boldsymbol{I} \cdot vec(\boldsymbol{Z}^{(\hat{i})}) \|_2^2 \end{aligned} \tag{19}$$

Using $vec(\boldsymbol{Z}^{(1)} \odot \boldsymbol{Z}^{(2)}) = \boldsymbol{I} \odot (\boldsymbol{Z}^{(2)} (\boldsymbol{I} \otimes \nVdash)) \cdot vec(\boldsymbol{Z}^{(1)})$ (Roemer 2012) and let $\boldsymbol{Z}^{(\widehat{i+1})} = \boldsymbol{Z}^{(i+1)} \odot \cdots \odot \boldsymbol{Z}^{(M)}$:

$$\begin{aligned} &\| \hat{\boldsymbol{x}} - (\boldsymbol{I} \otimes \boldsymbol{B}_{(1)}) (\boldsymbol{I} \odot \boldsymbol{Z}^{(\widehat{i-1})}) \otimes \boldsymbol{I} \cdot vec(\boldsymbol{Z}^{(\hat{i})}) \|_2^2 \\ &\quad = \| \hat{\boldsymbol{x}} - (\boldsymbol{I} \otimes \boldsymbol{B}_{(1)}) (\boldsymbol{I} \odot \boldsymbol{Z}^{(\widehat{i-1})}) \otimes \boldsymbol{I} \\ &\qquad \cdot \boldsymbol{I} \odot (\boldsymbol{Z}^{(\widehat{i+1})} (\boldsymbol{I} \otimes \nVdash)) \cdot vec(\boldsymbol{Z}^{(i)}) \|_2^2 \end{aligned} \tag{20}$$

Let $\hat{z}^{(i)} = vec(\boldsymbol{Z}^{(i)})$ be a vectorisation of $\boldsymbol{Z}^{(i)}$, this becomes:

$$\|\hat{\boldsymbol{x}} - (\boldsymbol{I} \otimes \boldsymbol{B}_{(1)})(\boldsymbol{I} \odot \boldsymbol{Z}^{(\widehat{i-1})}) \otimes \boldsymbol{I} \\ \cdot \boldsymbol{I} \odot (\boldsymbol{Z}^{(\widehat{i+1})}(\boldsymbol{I} \otimes \nVdash)) \cdot z^{\widehat{(i)}}\|_2^2 \tag{16}$$

We then compute the partial derivative of (16) with respect to $z^{\widehat{(i)}}$:

$$\frac{\partial \|\hat{\boldsymbol{x}} - \boldsymbol{A}z^{\widehat{(i)}}\|_2^2}{\partial z^{\widehat{(i)}}} = 2\boldsymbol{A}^T(\boldsymbol{A} \cdot z^{\widehat{(i)}} - \hat{\boldsymbol{x}}), \tag{21}$$

where $\boldsymbol{A} = (\boldsymbol{I} \otimes \boldsymbol{B}_{(1)})(\boldsymbol{I} \odot \boldsymbol{Z}^{(\widehat{i-1})}) \otimes \boldsymbol{I} \cdot \boldsymbol{I} \odot (\boldsymbol{Z}^{(\widehat{i+1})}(\boldsymbol{I} \otimes \nVdash))$.

The partial derivative of (15) with respect to $\boldsymbol{Z}^{(i)}$ is obtained by matricising (21).

To efficiently compute the above mentioned operations, Tensorly (Kossaifi et al. 2016) has been employed.

4 Proof of Concept Experiments

We develop a lighter version of our proposed network, a proof-of-concept network (visualised in Fig. 3), to show that our network is able to learn and disentangle pose, expression and identity.

In order to showcase the ability of the network, we leverage our newly proposed 4DFAB database (Cheng et al. 2018), where subjects were invited to attend four sessions at different times in a span of five years. In each experiment session, the subject was asked to articulate 6 different facial expressions (*anger, disgust, fear, happiness, sadness, surprise*), and we manually select the most expressive mesh (i.e. the apex frame) for this experiment. In total, 1795 facial meshes from 364 recording sessions (with 170 unique identities) are used. We keep 148 identities for training and leave 22 identities for testing. Note that there are no overlapping of identities between both sets. Within the training set, we synthetically augment each facial mesh by generating new facial meshes with 20 randomly selected expressions. Our training set contains in total 35900 meshes. The test set contains 387 meshes. For each mesh, we have the ground truth facial texture as well as expression and identity components of the 3DMM model.

4.1 Disentangling Expression and Identity

We create frontal images of the facial meshes. Hence there is no illumination or pose variation in this training dataset. We train a lighter version of our network by removing the illumination and pose streams, a proof-of-concept network, visualised in Fig. 3, on this synthetic dataset.

4.1.1 Expression Editing

We show the disentanglement between expression and identity by transferring the expression of one person to another.

For this experiment, we work with unseen data (a hold-out set consisting of 22 unseen identities) and no labels. We first encode both input images $\boldsymbol{x}^i$ and $\boldsymbol{x}^j$:

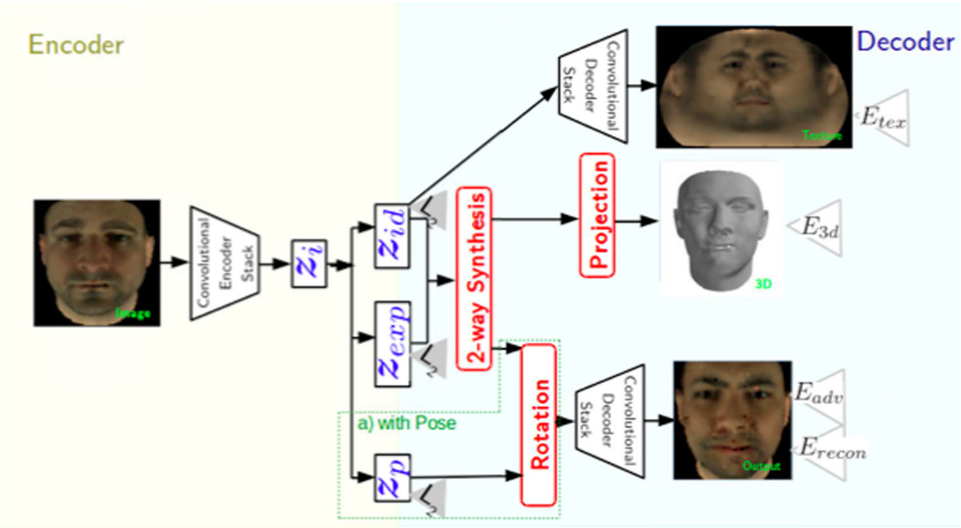

Fig. 3 Our proof-of-concept network is an end-to-end trained auto-encoder. The encoder E extracts latent variables corresponding to expression and identity from the input image $\boldsymbol{x}$. These latent variables are then fed into the decoder D to reconstruct the image. A separate stream also reconstructs facial texture from z_{id}. We impose a multilinear structure and enforce the disentanglement of variations. In the extended version a) the encoder also extracts a latent variable corresponding to pose. The decoder takes in this information and reconstructs an image containing pose variations

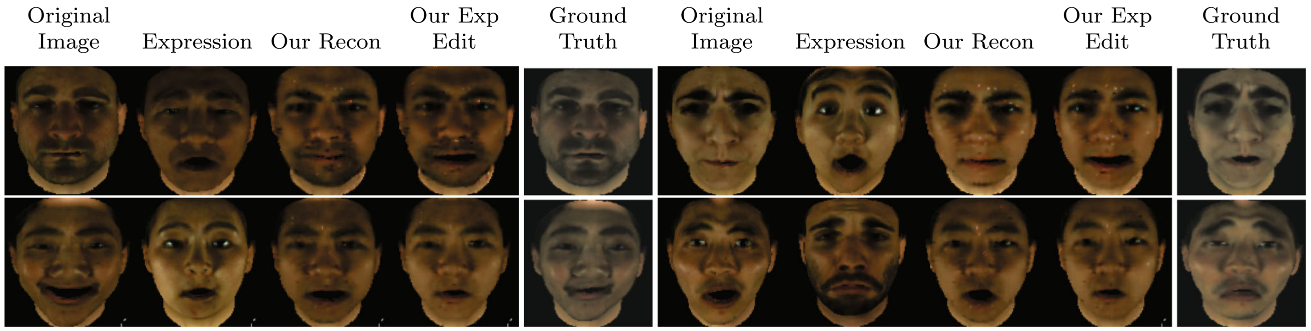

Fig. 4 Our network is able to transfer the expression from one face to another by disentangling the expression components of the images. The ground truth has been computed using the ground truth texture with synthetic identity and expression components

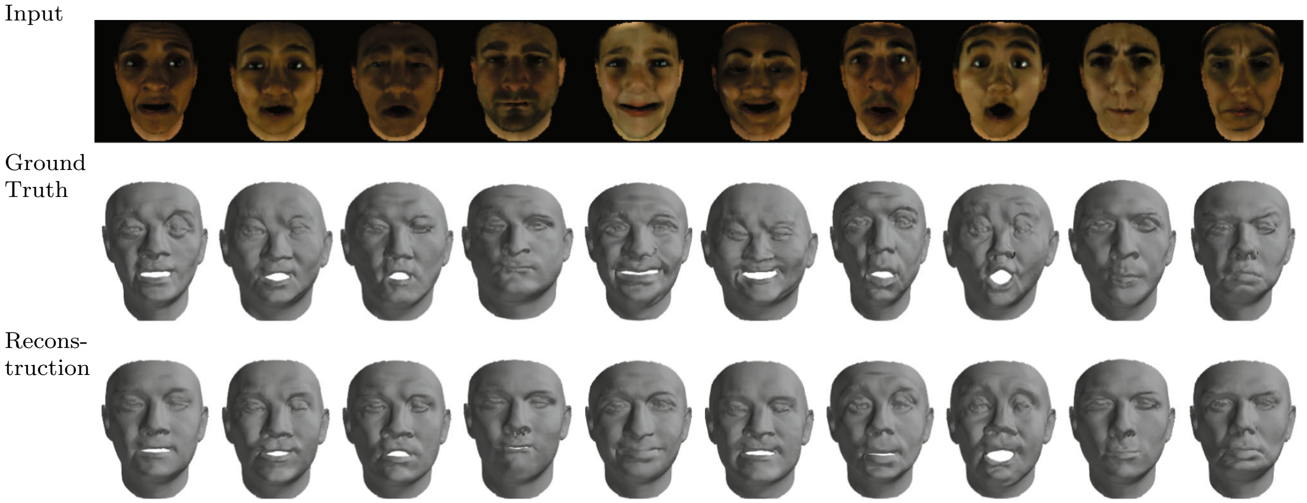

Fig. 5 Given a single image, we infer meaningful expression and identity components to reconstruct a 3D mesh of the face. We compare the reconstruction (last row) against the ground truth (2nd row)

$$\begin{aligned} E(\boldsymbol{x}^i) &= z^i_{exp}, z^i_{id}, \\ E(\boldsymbol{x}^j) &= z^j_{exp}, z^j_{id}, \end{aligned} \tag{22}$$

where $E(\cdot)$ is our encoder and z_{exp} and z_{id} are the latent representations of expression and identity respectively.

Assuming we want $\boldsymbol{x}^i$ to emulate the expression of $\boldsymbol{x}^j$, we decode on:

$$D(z^j_{exp}, z^i_{id}) = \boldsymbol{x}^{ji}, \tag{23}$$

where $D(\cdot)$ is our decoder. The resulting $\boldsymbol{x}^{ji}$ becomes our edited image where $\boldsymbol{x}^i$ has the expression of $\boldsymbol{x}^j$. Figure 4 shows how the network is able to separate expression and identity. The edited images clearly maintain the identity while expression changes.

4.1.2 3D Reconstruction and Facial Texture

The latent variables z_{exp} and z_{id} that our network learns are extremely meaningful. Not only can they be used to reconstruct the image in 2D, but also they can be mapped into the expression ($\boldsymbol{x}_{exp}$) and identity ($\boldsymbol{x}_{id}$) components of a 3DMM model. This mapping is learnt inside the network. By replacing the expression and identity components of a mean face shape with $\boldsymbol{x}_{\hat{exp}}$ and $\hat{\boldsymbol{x}_{id}}$, we are able to reconstruct the 3D mesh of a face given a single input image. We compare these reconstructed meshes against the ground truth 3DMM used to create the input image in Fig. 5.

At the same time, the network is able to learn a mapping from z_{id} to facial texture. Therefore, we can predict the facial texture given a single input image. We compare the reconstructed facial texture with the ground truth facial texture in Fig. 6.

4.2 Disentangling Pose, Expression and Identity

Our synthetic training set contains in total 35900 meshes. For each mesh, we have the ground truth facial texture as well as expression and identity components of the 3DMM, from which we create a corresponding image with one of 7 given

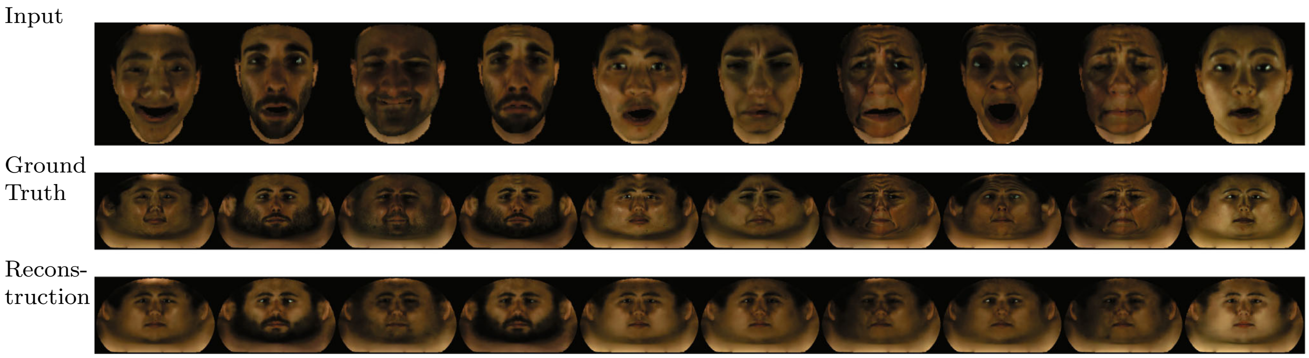

Fig. 6 Given a single image, we infer the facial texture. We compare the reconstructed facial texture (last row) against the ground truth texture (2nd row)

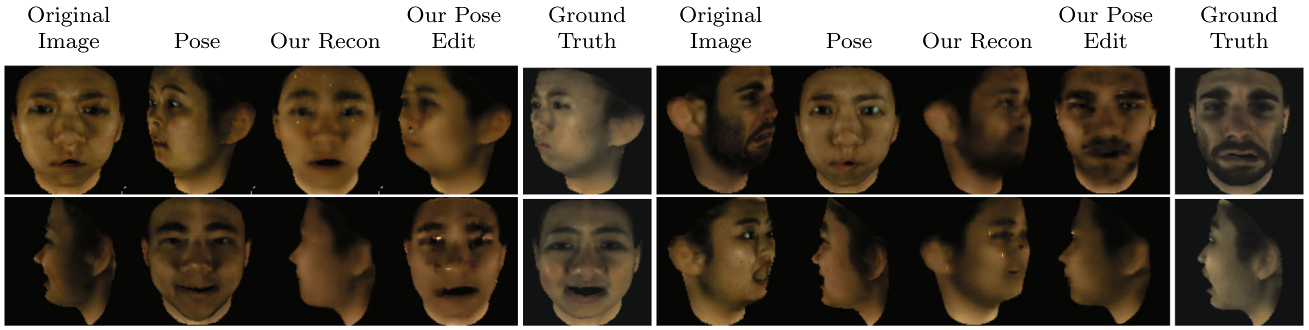

Fig. 7 Our network is able to transfer the pose from one face to another by disentangling the pose, expression and identity components of the images. The ground truth has been computed using the ground truth texture with synthetic pose, identity and expression components

poses. As there is no illumination variation in this training set, we train a proof-of-concept network by removing the illumination stream, visualised in Fig. 3a, on this synthetic dataset.

4.2.1 Pose Editing

We show the disentanglement between pose, expression and identity by transferring the pose of one person to another. Figure 7 shows how the network is able to separate pose from expression and identity. This experiment highlights the ability of our proposed network to learn large pose variations even from profile to frontal faces.

5 Experiments in-the-Wild

We train our network on in-the-wild data and perform several experiments on unseen data to show that our network is indeed able to disentangle illumination, pose, expression and identity.

We edit expression or pose by swapping the latent expression/pose component learnt by the encoder E [Eq. (6)] with the latent expression/pose component predicted from another image. We feed the decoder D [Eq. (7)] with the modified latent component to retrieve our edited image.

5.1 Expression, Pose and Identity Editing in-the-Wild

Given two in-the-wild images of faces, we are able to transfer the expression, pose of one person to another. We are also able to swap the face of the person from one image to another. Transferring the expression from two different facial images without fitting a 3D model is a very challenging problem. Generally, it is considered in the context of the same person under an elaborate blending framework (Yang et al. 2011) or by transferring certain classes of expressions (Sagonas et al. 2017).

For this experiment, we work with completely unseen data (a hold-out set of CelebA) and no labels. We first encode both input images $\boldsymbol{x}^i$ and $\boldsymbol{x}^j$:

$$\begin{aligned} E(\boldsymbol{x}^i) &= z^i_{exp}, z^i_{id}, z^i_p \\ E(\boldsymbol{x}^j) &= z^j_{exp}, z^j_{id}, z^j_p, \end{aligned} \tag{24}$$

where $E(\cdot)$ is our encoder and z_{exp}, z_{id}, z_p are the latent representations of expression, identity and pose respectively.

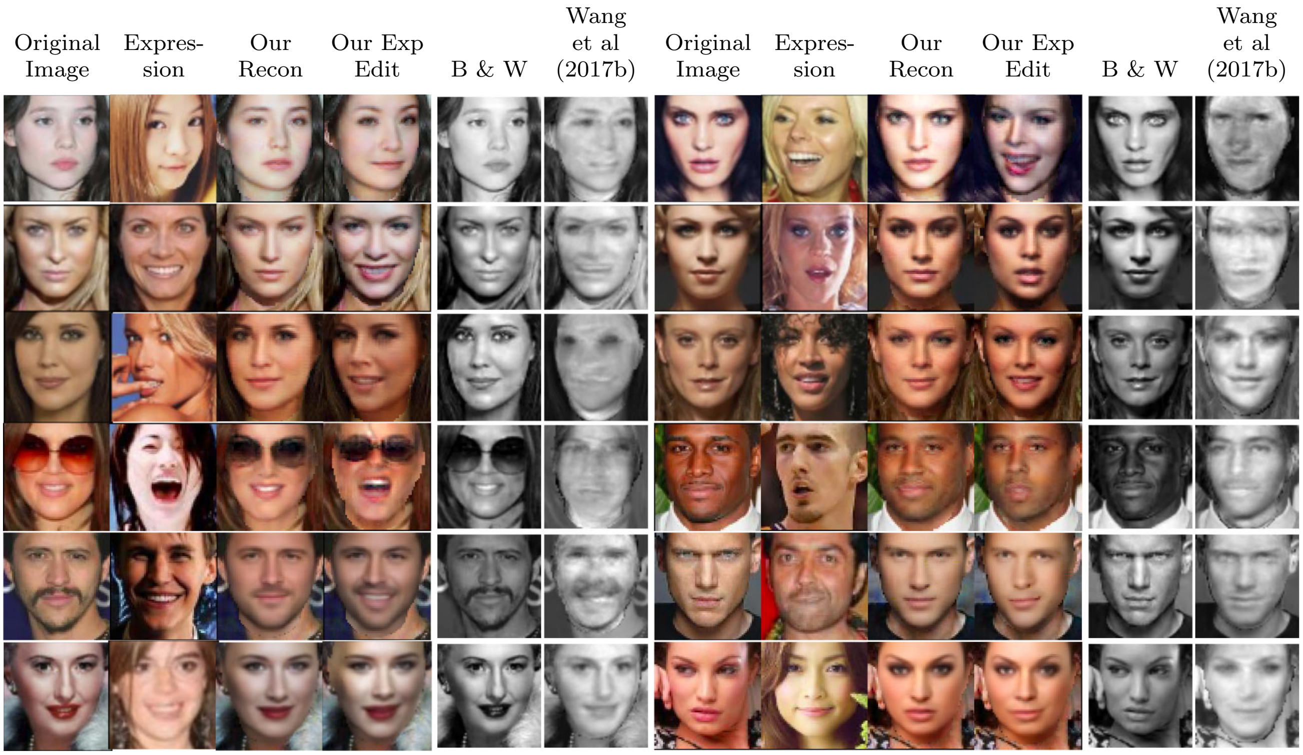

Fig. 8 We compare our expression editing results with Wang et al. (2017b). As Wang et al. (2017b) is not able to disentangle pose, editing expressions from images of different poses returns noisy results

Assuming we want $\boldsymbol{x}^i$ to take on the expression, pose or identity of $\boldsymbol{x}^j$, we then decode on:

$$\begin{aligned} D(z^j_{exp}, z^i_{id}, z^i_p) &= \boldsymbol{x}^{jii} \\ D(z^i_{exp}, z^i_{id}, z^j_p) &= \boldsymbol{x}^{iij} \\ D(z^i_{exp}, z^j_{id}, z^i_p) &= \boldsymbol{x}^{iji} \end{aligned} \tag{25}$$

where $D(\cdot)$ is our decoder.

The resulting $\boldsymbol{x}^{jii}$ then becomes our result image where $\boldsymbol{x}^i$ has the expression of $\boldsymbol{x}^j$. $\boldsymbol{x}^{jii}$ is the edited image where $\boldsymbol{x}^i$ changed to the pose of $\boldsymbol{x}^j$. $\boldsymbol{x}^{iji}$ is the edit where $\boldsymbol{x}^i$'s face changed to the face of $\boldsymbol{x}^j$.

As there is currently no prior work for this expression editing experiment without fitting an AAM (Cootes et al. 2001) or 3DMM, we used the image synthesised by the 3DMM fitted models as a baseline, which indeed performs quite well. Compared with our method, other very closely related works (Wang et al. 2017b; Shu et al. 2017) are not able to disentangle illumination, pose, expression and identity. In particular, Shu et al. (2017) disentangles illumination of an image while Wang et al. (2017b) disentangles illumination, expression and identity from "frontalised" images. Hence they are not able to disentangle pose. None of these methods can be applied to the expression/pose editing experiments on a dataset that contains pose variations such as CelebA. If Wang et al. (2017b) is applied directly on our test images, it would not be able to perform expression editing well, as shown by Fig. 8.

For the 3DMM baseline, we fit a shape model to both images and extract the expression components of the model. This fitting step has high overhead of 20 s per image. We then generate a new face shape using the expression components of one face and the identity components of another face in the same 3DMM setting. This technique has much higher overhead than our proposed method as it requires time-consuming 3DMM fitting of the images. Our expression editing results and the baseline results are shown in Fig. 9. Though the baseline is very strong, it does not change the texture of the face which can produce unnatural looking faces shown with original expression. Also, the baseline method can not fill up the inner mouth area. Our editing results show more natural looking faces.

For pose editing, the background is unknown once the pose has changed, thus, for this experiment, we mainly focus on the face region. Figure 10 shows our pose editing results. For the baseline method, we fit a 3DMM to both images and estimate the rotation matrix. We then synthesise $\boldsymbol{x}_i$ with the rotation of $\boldsymbol{x}_j$. This technique has high overhead as it requires expensive 3DMM fitting of the images.

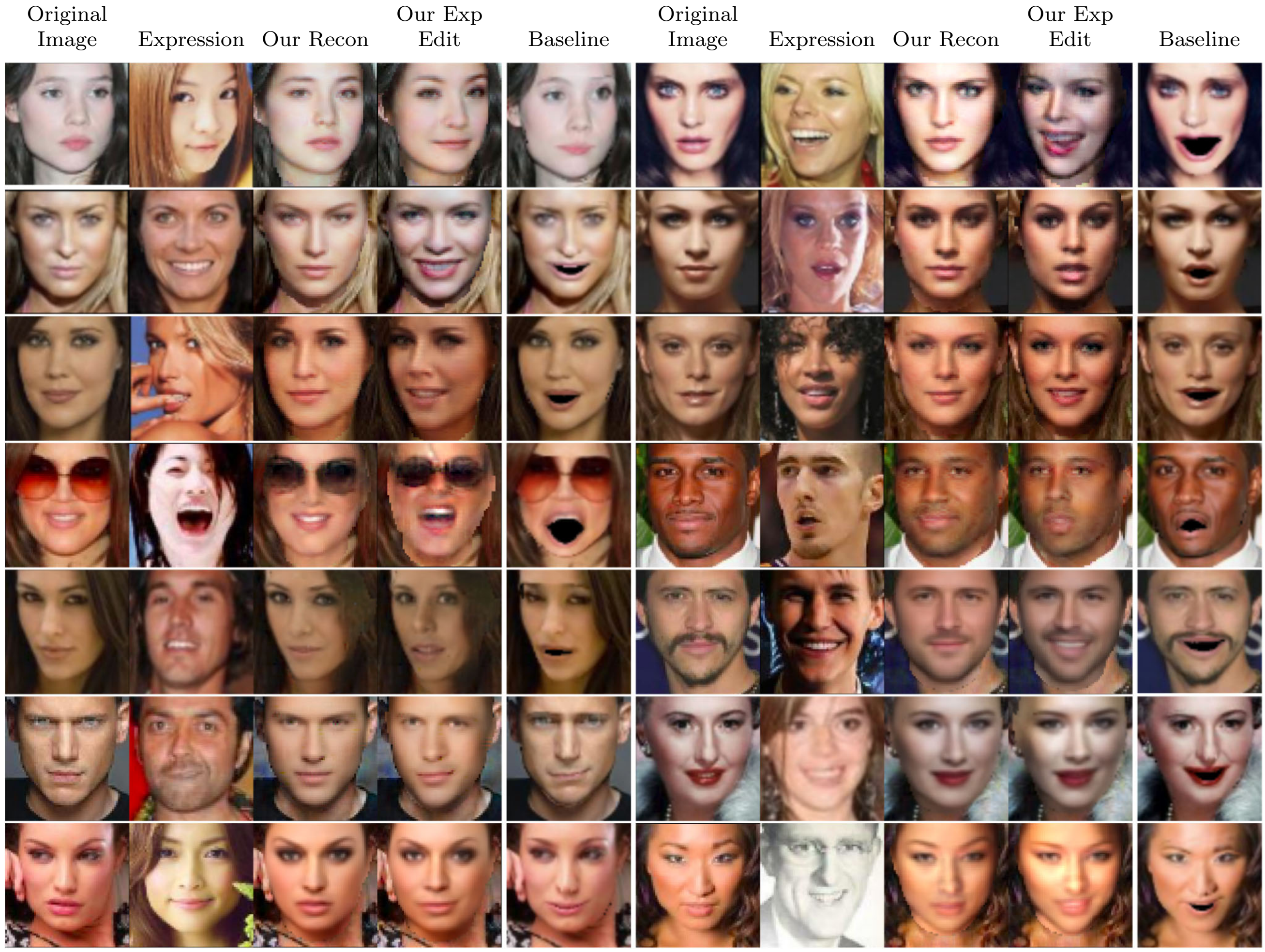

Fig. 9 Our network is able to transfer the expression from one face to another by disentangling the expression components of the images. We compare our expression editing results with a baseline where a 3DMM has been fit to both input images

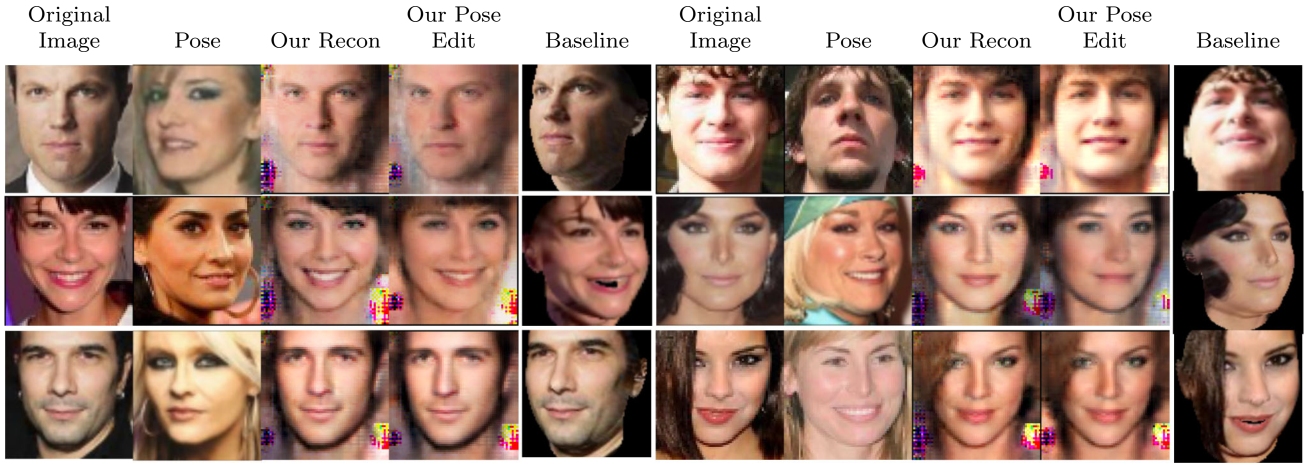

Fig. 10 Our network is able to transfer the pose of one face to another by disentangling the pose components of the images. We compare our pose editing results with a baseline where a 3DMM has been fit to both input images

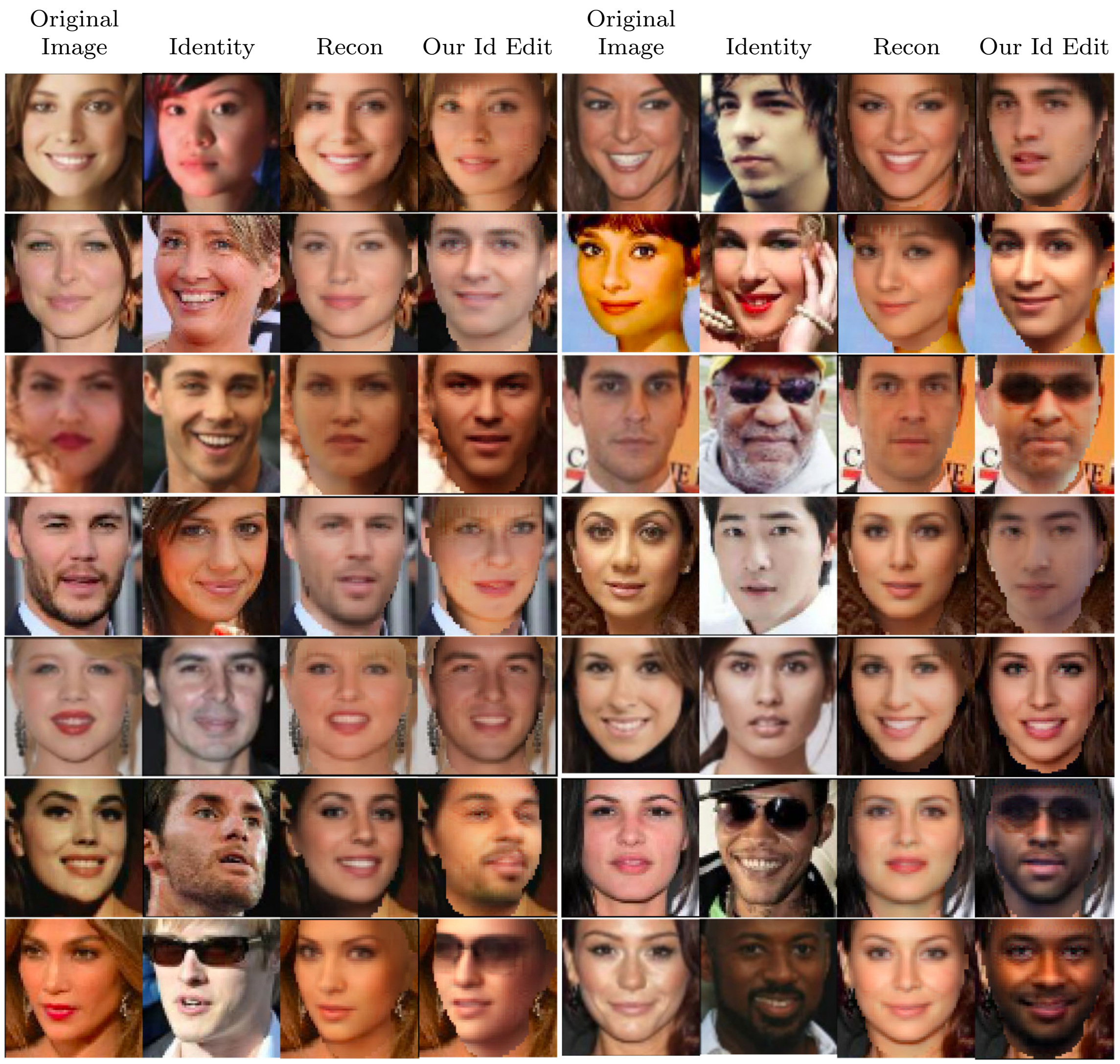

Fig. 11 Our network is also able to transfer the identity of one image to another by disentangling the identity components of the images

Figure 11 shows our results on the task of face swapping where the identity of one image has been swapped with the face of another person from the second image.

5.1.1 Quantitative Studies

We conducted a quantitative measure on the expression editing experiment. We ran a face recognition experiment on 50 pairs of images where only the expression has been transferred. We then passed them to a face recognition network (Deng et al. 2018) and extracted their respective embeddings. All 50 pairs of embeddings had cosine similarity larger than 0.3. In comparison, We selected 600 pairs of different people from CelebA and computed their average cosine similarity which is 0.062. The histogram of these cosine similarities is visualised in Fig. 12. This indicates that the expression editing does conserve identity in terms of machine perception.

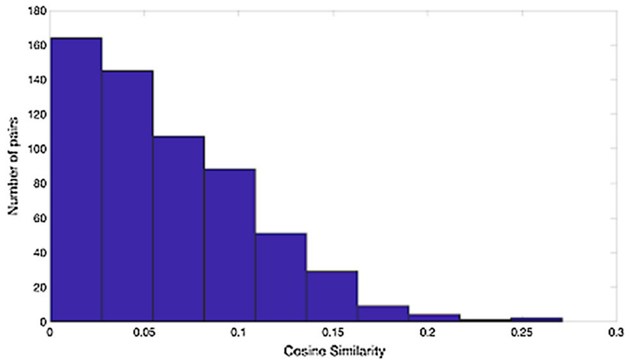

Fig. 12 Histogram of cosine similarities on 600 pairs of “non-same” people from CelebA

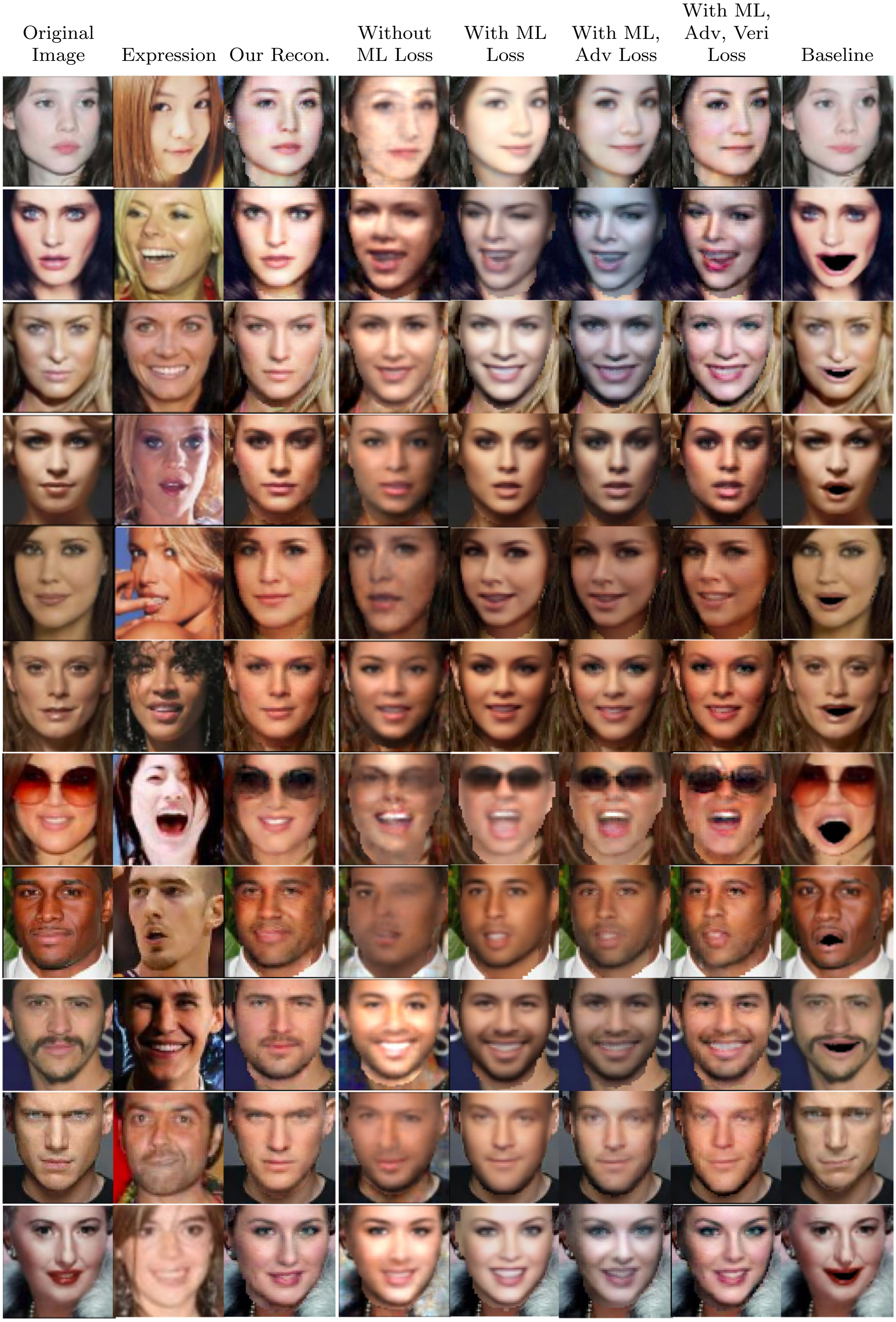

Fig. 13 Ablation study on different losses (multilinear, adversarial, verification) for expression editing. The results show that incorporating multilinear losses indeed helps the network to better disentangle the expression variations

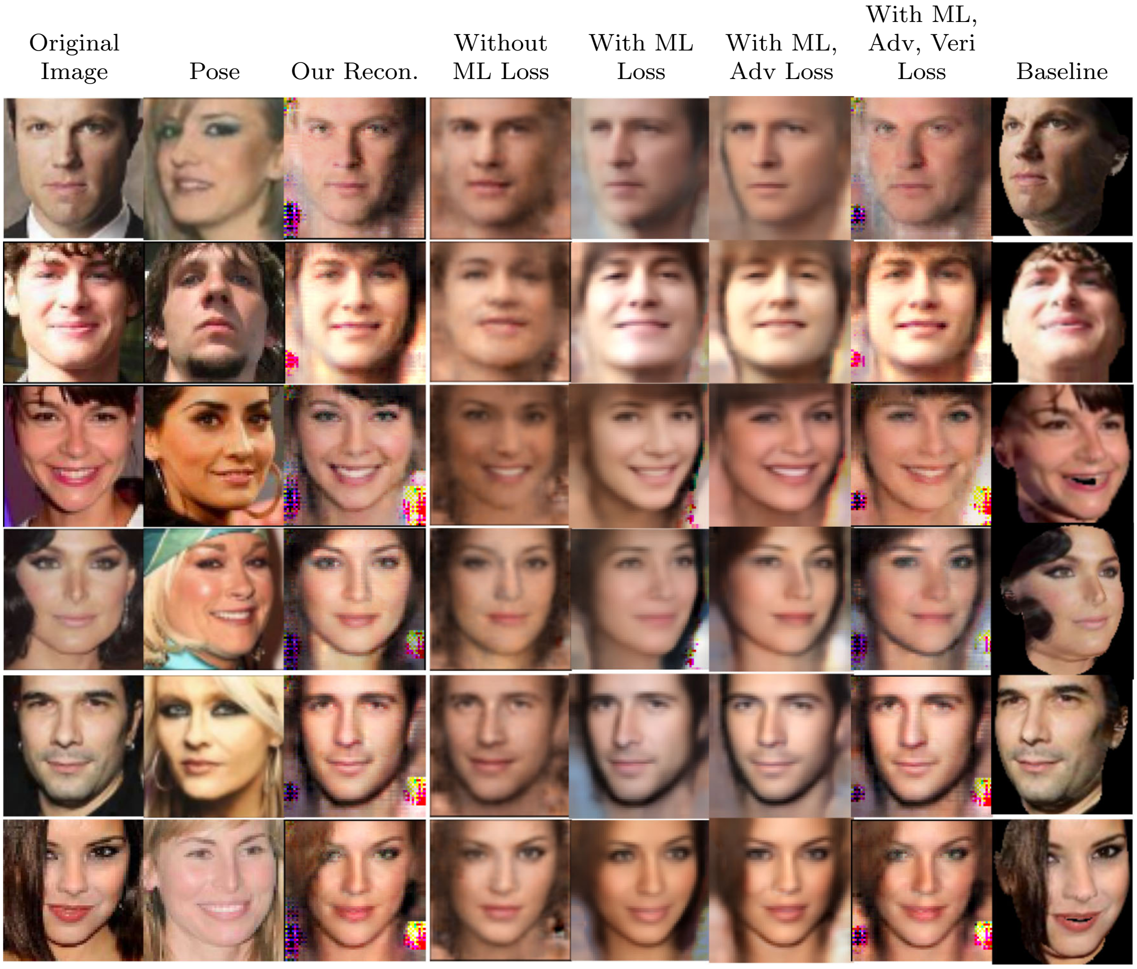

Fig. 14 Ablation study on different losses (multilinear, adversarial, verification) for facial pose editing. The results show that incorporating multilinear losses helps the network to better disentangle the pose variations

5.1.2 Ablation Studies

We performed a series of ablation studies. We first trained a network without multilinear losses by simply feeding the concatenated parameters $\boldsymbol{p} = [z_{pose}, z_{exp}, z_{id}]$ to the decoder, thus the training of the network is only driven by the reconstruction loss and pseudo-supervision from 3DMM on pose, expression and identity latent variables, i.e., z_{pose}, z_{exp} and z_{id}. Next, we started to incorporate other losses (i.e., multilinear losses, adversarial loss, verification loss) step by step in the network and trained different models. In this way, we can observe at each step how additional loss may improve the result.

In Figs. 13 and 14, we compare the expression and pose editing results. We find that the results without multilinear losses shows some entanglement of the variations in terms of illumination, identity, expression and pose. In particular, the entanglement with illumination is strong, examples can be found in second and ninth row of Fig. 13. Indeed, by incorporating multilinear losses in the network, the identity and expression variations are better disentangled. Furthermore, the incorporation of adversarial and verification losses enhances the quality of images, making them look more

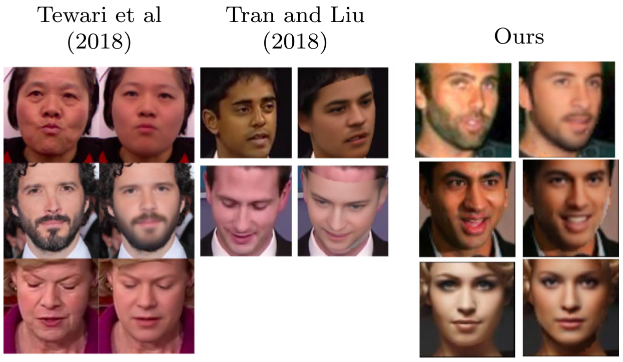

Fig. 15 Texture reconstruction compared with Tewari et al. (2018), Tran and Liu (2018). Tewari et al. (2018), Tran and Liu (2018) have been trained with images of higher resolutions of 240 × 240 and 128 × 128 respectively. In comparison our model has only been trained with images of size 64 × 64 pixels

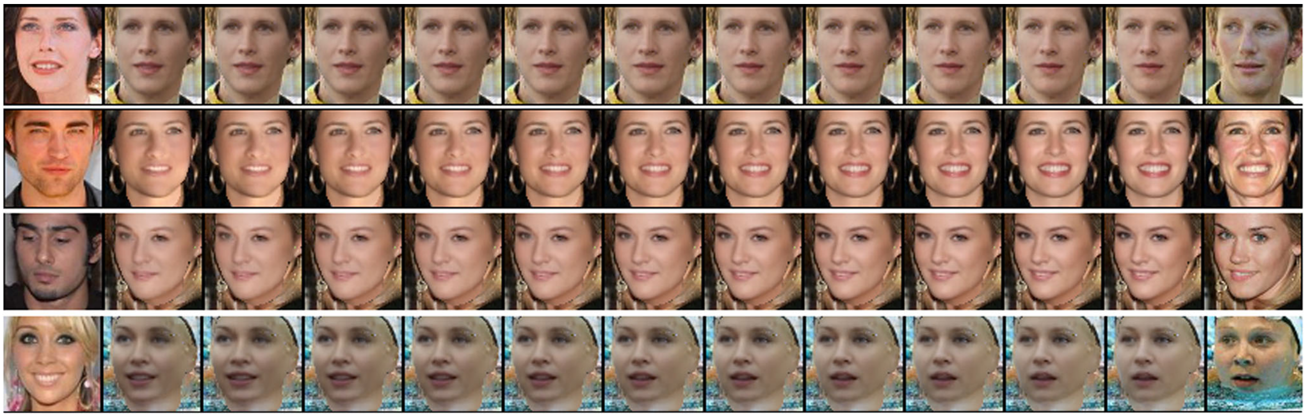

Fig. 16 Expression interpolation

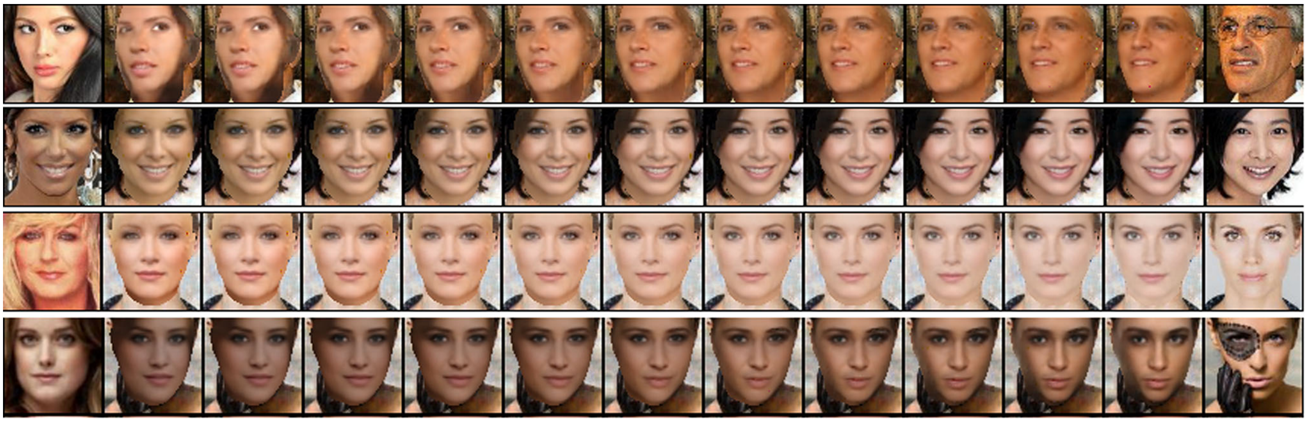

Fig. 17 Identity interpolation

realistic but do not contribute in a meaningful way to the disentanglement.

5.1.3 Discussion on Texture Quality

It has to be noted that our baseline 3DMM method Booth et al. (2017) does not change facial texture. It directly samples the original texture and maps it to a 3D face. Hence, the texture quality is exactly the same as that of the original image as no low-dimensional texture representation is used. In terms of texture quality, direct texture mapping has an edge over our proposed method which models the texture using a low-dimensional representation. But direct texture mapping is also prone to artefacts and does not learn the new expression in the texture. Looking at Fig. 9 column 2, rows 4, 5 and 7, we observe that the texture itself did not change in the baseline result. The eyes and cheeks did not adjust to show a smiling or neutral face. The expression change results from the change in the 3D shape but the texture itself remained the same as in the input. Low-dimensional texture representation does not have this issue and can generate new texture with changed expression.

Generally methods similar to ours which estimate facial texture is not able to extract the same amount of details as the original image. Figure 15 visualises how our texture reconstruction compares to state-of-the-art works which have been trained on images of higher resolutions.

5.2 Expression and Identity Interpolation

We interpolate z^i_{exp} / z^i_{id} of the input image $\boldsymbol{x}^i$ on the right-hand side to the z^t_{exp} / z^t_{id} of the target image $\boldsymbol{x}^t$ on the left-hand side. The interpolation is linear and at 0.1 interval. For the interpolation we do not modify the background so the background remains that of image $\boldsymbol{x}^i$.

For expression interpolation, we expect the identity and pose to stay the same as the input image $\boldsymbol{x}^i$ and only the expression to change gradually from the expression of the input image to the expression of the target image $\boldsymbol{x}^t$. Figure 16 shows the expression interpolation. We can clearly see the change in expression while pose and identity remain constant.

For identity interpolation, we expect the expression and pose to stay the same as the input image $\boldsymbol{x}^i$ and only the

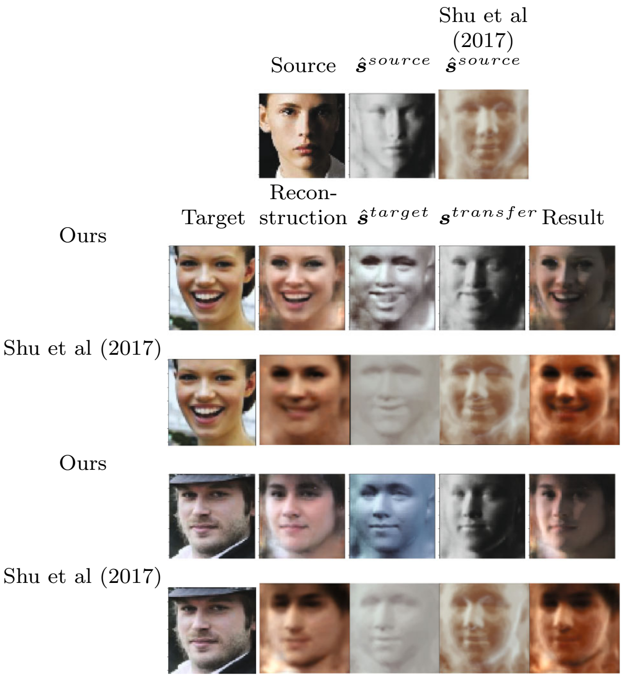

Fig. 18 Using the illumination and normals estimated by our network, we are able to relight target faces using illumination from the source image. The source $\hat{s}^{source}$ and target shading $\hat{s}^{target}$ are displayed to visualise against the new transferred shading $s^{transfer}$. We compare against Shu et al. (2017)

identity to change gradually from the identity of the input image to the identity of the target image $\boldsymbol{x}^t$. Figure 17 shows the identity interpolation. We can clearly observe the change in identity while other variations remain limited.

5.3 Illumination Editing

We transfer illumination by estimating the normals $\hat{\boldsymbol{n}}$, albedo $\hat{\boldsymbol{a}}$ and illumination components $\hat{\boldsymbol{l}}$ of the source ($\boldsymbol{x}^{source}$) and target ($\boldsymbol{x}^{target}$) images. Then we use $\hat{\boldsymbol{n}}^{target}$ and $\hat{\boldsymbol{l}}^{source}$ to compute the transferred shading $\boldsymbol{s}^{transfer}$ and multiply the new shading by $\hat{\boldsymbol{a}}^{target}$ to create the relighted image result $\boldsymbol{x}^{transfer}$. In Fig. 18 we show the performance of our method and compare against Shu et al. (2017) on illumination transfer. We observe that our method outperforms Shu et al. (2017) as we obtain more realistic looking results.

5.4 3D Reconstruction

The latent variables z_{exp} and z_{id} that our network learns are extremely meaningful. Not only can they be used to reconstruct the image in 2D, they can be mapped into the expression ($\boldsymbol{x}_{exp}$) and identity ($\boldsymbol{x}_{id}$) components of a 3DMM. This mapping is learnt inside the network. By replacing the expression and identity components of a mean face shape with $\boldsymbol{x}_{\hat{exp}}$ and $\boldsymbol{x}_{\hat{id}}$, we are able to reconstruct the 3D mesh of a face given a single in-the-wild 2D image. We compare these reconstructed meshes against the fitted 3DMM to the input image.

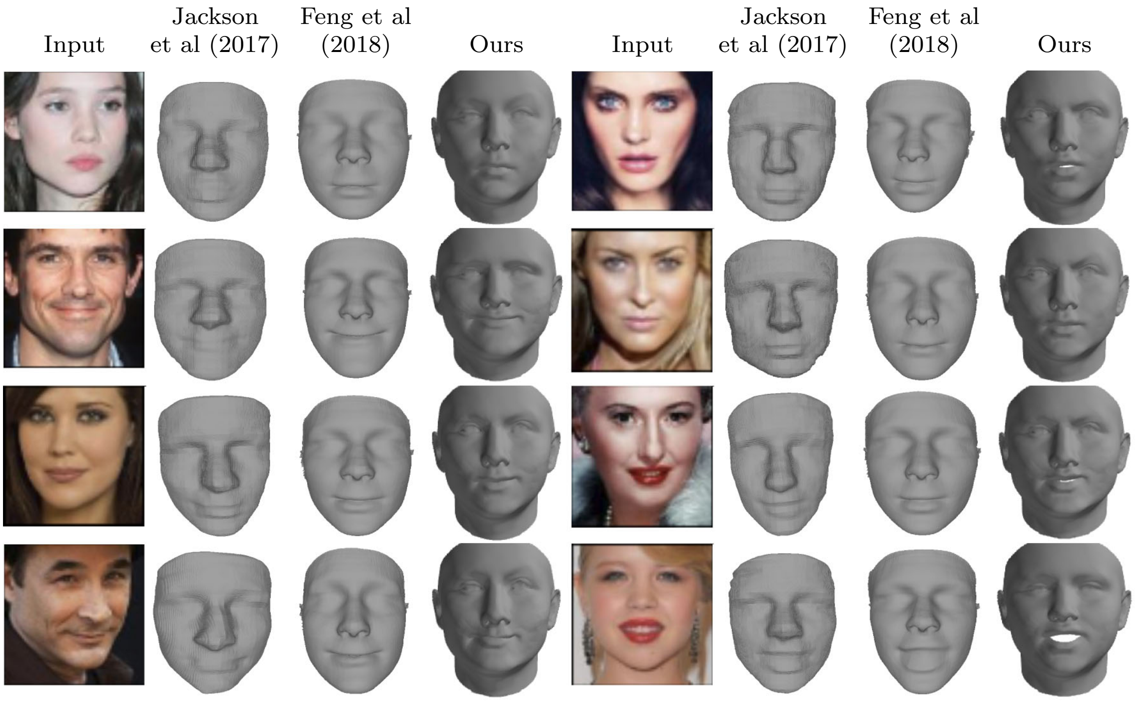

Fig. 19 Given a single image, we infer meaningful expression and identity components to reconstruct a 3D mesh of the face. We compare our 3D estimation against recent works (Jackson et al. 2017; Feng et al. 2018)

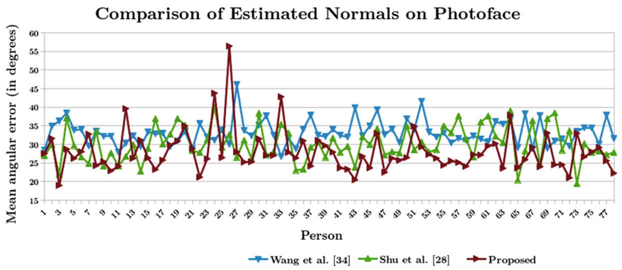

Fig. 20 Comparison of the estimated normals obtained using the proposed model vs the ones obtained by Wang et al. (2017b) and Shu et al. (2017)

Table 1 Angular error for the various surface normal estimation methods on the Photoface (Zafeiriou et al. 2013) dataset. We also show the proportion of the normals below 35° and 40°

Method	Mean ± Std against Woodham (1980)	< 35° (%)	< 40° (%)
Wang et al. (2017b)	33.37° ± 3.29°	75.3	96.3
Shu et al. (2017)	30.09° ± 4.66°	84.6	98.1
Proposed	28.67° ± 5.79°	89.1	96.3

The results of the experiment are visualised in Fig. 19. We observe that the reconstruction is comparable to other state-of-the-art techniques (Jackson et al. 2017; Feng et al. 2018). None of the techniques though capture well the identity of the person in the input image due to a known weakness in 3DMM.

5.5 Normal Estimation

We evaluate our method on the surface normal estimation task on the Photoface (Zafeiriou et al. 2013) dataset which has information about illumination. Assuming the normals found using calibrated Photometric Stereo (Woodham 1980) as "ground truth", we calculate the angular error between our estimated normals and the "ground truth". Figure 20 and Table 1 quantitatively evaluates our proposed method against prior works (Wang et al. 2017b; Shu et al. 2017) in the normal estimation task. We observe that our proposed method performs on par or outperforms previous methods.

Table 2 Classification accuracy results: we try to classify 54 identities using z_{id}, 6 expressions using z_{exp} and 7 poses using z_p. We compare against standard baseline methods such as SIFT and CNN

Features	Identity (%)	Expression (%)	Pose (%)
SIFT and visual bag of words, K = 50	14.60	58.33	55.50
SIFT and visual bag of words, K = 100	18.71	59.36	59.46
Standard CNN model	94.68	96.54	98.78
Ours ($z_{identity}$, $z_{expression}$, z_{pose})	88.29	84.85	95.55

Table 3 Identity classification accuracy results: we classify 54 identities using z_{id} with and without verification loss

Features	Identity (%)
Without verification loss	87.94
Ours ($z_{identity}$)	**88.29**
Without verification loss (frontal only)	99.96
Ours ($z_{identity}$, frontal only)	**99.98**

Top performing values are given in bold

Table 4 Classification accuracy results in comparison with Wang et al. (2017b): as Wang et al. (2017b) works on frontal images, we only consider frontal images in this experiment. We try to classify 54 identities using z_{id} versus $\boldsymbol{C}$, 6 expressions using z_{exp} versus $\boldsymbol{E}$ and 16 illumination using z_{ill} versus $\boldsymbol{L}$

	$z_{identity}$ (%)	$\boldsymbol{C}$ (Wang et al. 2017b) (%)
Identity		
Accuracy	**99.33**	19.18
	$z_{expression}$ (%)	$\boldsymbol{E}$ (Wang et al. 2017b) (%)
Expression		
Accuracy	**78.92**	35.49
	$z_{illumination}$ (%)	$\boldsymbol{L}$ (Wang et al. 2017b) (%)
Illumination		
Accuracy	**64.11**	48.85

Top performing values are given in bold

5.6 Quantitative Evaluation of the Latent Space

We want to test whether our latent space corresponds well to the variation that it is supposed to learn. For our quantitative experiment, we used Multi-PIE (Gross et al. 2010) as our test dataset. This dataset contains labelled variations in identity, expressions and pose. Disentanglement of variations in Multi-PIE is particularly challenging as its images are captured under laboratory conditions which is quite different from that of our training images. As a matter of fact, the expressions contained in Multi-PIE do not correspond to the 7 basic expressions and can be easily confused.

We encoded 10,368 images of the Multi-PIE dataset with 54 identities, 6 expressions and 7 poses and trained a linear SVM classifier using 90% of the identity labels and the latent variables z_{id}. We then test on the remaining 10% z_{id} to check whether they are discriminative for identity classification. We use 10-fold cross-validation to evaluate the accuracy of the learnt classifier. We repeat this experiment for expression with z_{exp} and pose with z_p respectively. Our results in Table 2 show that our latent representation is indeed

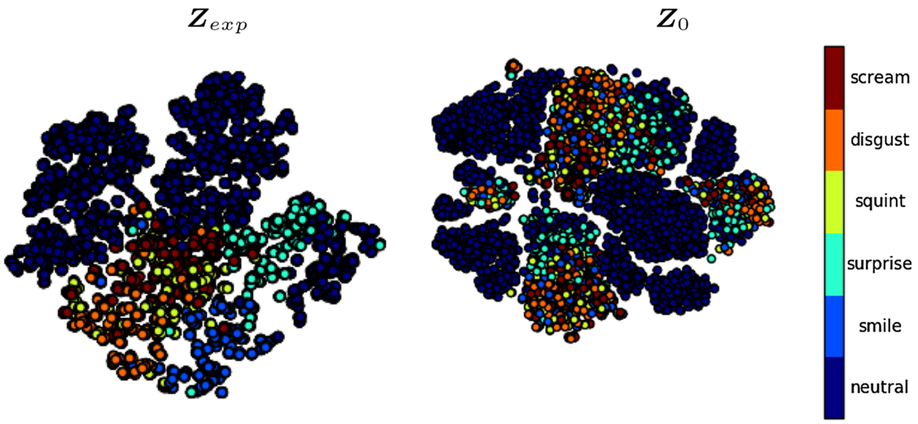

Fig. 21 Visualisation of our $\boldsymbol{Z}_{exp}$ and baseline $\boldsymbol{Z}_0$ using t-SNE. Our latent $\boldsymbol{Z}_{exp}$ clusters better with regards to expression than the latent space $\boldsymbol{Z}_0$ of an auto-encoder

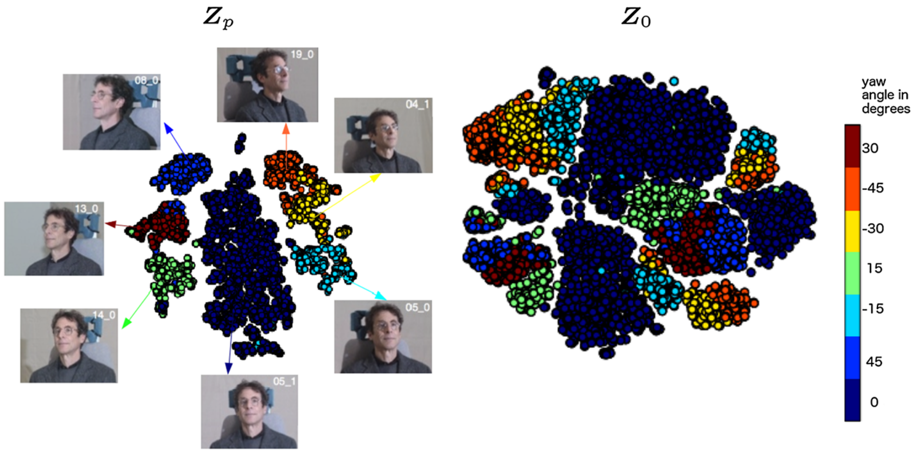

Fig. 22 Visualisation of our $\boldsymbol{Z}_p$ and baseline $\boldsymbol{Z}_0$ using t-SNE. It is evident that the proposed disentangled $\boldsymbol{Z}_p$ clusters better with regards to pose than the latent space $\boldsymbol{Z}_0$ of an auto-encoder

discriminative. We compare against some standard baselines such as Bag-of-Words (BoWs) models with SIFT feature (Sivic and Zisserman 2009) and standard CNN. Our model does not outperform the standard CNN model, which is fully supervised and requires a separate model for each variation classification. Still our results are a strong indication that the latent representation found is discriminative. This experiment showcases the discriminative power of our latent representation on a previously unseen dataset.

As an ablation study, we test the accuracy of the identity classification of z_{id} from a model trained without the verification. The results in Table 3 show that though adding the verification loss improves the performance, the gain is not significant enough to prove that this loss is a substantial contributor of the information.

In order to quantitatively compare with Wang et al. (2017b), we run another experiment on only frontal images of the dataset with 54 identities, 6 expressions and 16 illuminations. The results in Table 4 shows how our proposed model outperforms (Wang et al. 2017b) in these classification tasks. Our latent representation has stronger discriminative power than the one learnt by Wang et al. (2017b).

We visualise, using t-SNE (Maaten and Hinton 2008), the latent $\boldsymbol{Z}_{exp}$ and $\boldsymbol{Z}_p$ encoded from Multi-PIE according to their expression and pose label and compare against the latent representation $\boldsymbol{Z}_0$ learnt by an in-house large-scale adversarial auto-encoder of similar architecture trained with 2 million faces (Makhzani et al. 2015). Figures 21 and 22 show that even though our encoder has not seen any images of Multi-PIE, it manages to create informative latent representations that cluster well expression and pose (contrary to the representation learned by the tested auto-encoder).

6 Limitations

Some of our results do still show entanglement in the variations. Sometimes despite only aiming to change expression only, pose or illumination have been modified as well. This happens mainly in very challenging scenarios where for example one of the image shows extreme lighting conditions, is itself black and white or displays large pose variations. Due to the dataset (CelebA) we used, we do struggle with large pose variations. The proof of concept experiments do show that this is possible to be learned with a more balanced dataset.

7 Conclusion

We proposed the first, to the best of our knowledge, attempt to jointly disentangle modes of variation that correspond to expression, identity, illumination and pose using no explicit labels regarding these attributes. More specifically, we proposed the first, as far as we know, approach that combines a powerful Deep Convolutional Neural Network (DCNN) architecture with unsupervised tensor decompositions. We demonstrate the power of our methodology in expression and pose transfer, as well as discovering powerful features for pose and expression classification. For future work, we believe that designing networks with skip connections for better reconstruction quality and which at the same time can learn a representation space where some of the variations are disentangled would be a promising research direction.

Acknowledgements Mengjiao Wang was supported by an EPSRC DTA from Imperial College London. This work was partially funded by a gift from Adobe, NSF grants CNS-1718014 and DMS 1737876, the Partner University Fund, and the SUNY2020 Infrastructure Transportation Security Center awarded to Zhixin Shu, as well as by a Google Faculty Award and EPSRC Fellowship DEFORM: Large Scale Shape Analysis of Deformable Models of Humans (EP/S010203/1) awarded to Dr. Zafeiriou. We thank Amazon Web Services for providing computational resources.

References

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828.

Berthelot, D., Schumm, T., & Metz, L. (2017). Began: Boundary equilibrium generative adversarial networks. arXiv preprint arXiv:1703.10717.

Bolkart, T., & Wuhrer, S. (2016). A robust multilinear model learning framework for 3D faces. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4911–4919).

Booth, J., Antonakos, E., Ploumpis, S., Trigeorgis, G., Panagakis, Y., & Zafeiriou, S. (2017). 3D face morphable models "in-the-wild". arXiv preprint arXiv:1701.05360.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems* (pp. 2172–2180).

Cheng, S., Kotsia, I., Pantic, M., & Zafeiriou, S. (2018). 4DFAB: A large scale 4D database for facial expression analysis and biometric applications. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5117–5126).

Cheung, B., Livezey, J. A., Bansal, A. K., & Olshausen, B. A. (2014). Discovering hidden factors of variation in deep networks. arXiv preprint arXiv:1412.6583.

Cootes, T. F., Edwards, G. J., & Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(6), 681–685.

De Lathauwer, L., De Moor, B., & Vandewalle, J. (2000). A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, *21*(4), 1253–1278.

Deng, J., Guo, J., & Zafeiriou, S. (2018). Arcface: Additive angular margin loss for deep face recognition. CoRR. arXiv:1801.07698.

Desjardins, G., Courville, A., & Bengio, Y. (2012). Disentangling factors of variation via generative entangling. arXiv Preprint arXiv:1210.5474.

Fabrigar, L. R., & Wegener, D. T. (2011). *Exploratory factor analysis*. Oxford: Oxford University Press.

Feng, Y., Wu, F., Shao, X. H., Wang, Y. F., & Zhou, X. (2018). Joint 3D face reconstruction and dense alignment with position map regression network. In *The European conference on computer vision (ECCV)*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

Gross, R., Matthews, I., Cohn, J., Kanade, T., & Baker, S. (2010). Multi-PIE. *Image and Vision Computing*, *28*(5), 807–813.

Hinton, G. E., Krizhevsky, A., & Wang, S. D. (2011). Transforming auto-encoders. In *International conference on artificial neural networks* (pp. 44–51). Springer.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*(6), 417.

Jackson, A. S., Bulat, A., Argyriou, V., & Tzimiropoulos, G. (2017). Large pose 3D face reconstruction from a single image via direct volumetric CNN regression. In *International conference on computer vision*.

Kemelmacher-Shlizerman, I. (2013). Internet based morphable model. In *Proceedings of the IEEE international conference on computer vision* (pp. 3256–3263).

Kolda, T. G., & Bader, B. W. (2008). Tensor decompositions and applications. *SIAM Review*, *51*(3), 455–500. https://doi.org/10.1137/07070111X.

Kossaifi, J., Panagakis, Y., Pantic, M. (2016). Tensorly: Tensor learning in python. ArXiv e-print.

Kulkarni, T. D., Whitney, W. F., Kohli, P., & Tenenbaum, J. (2015). Deep convolutional inverse graphics network. In *Advances in neural information processing systems* (pp. 2539–2547).

Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of international conference on computer vision (ICCV)*.

Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). Adversarial autoencoders. arXiv Preprint arXiv:1511.05644.

Mathieu, M. F., Zhao, J. J., Zhao, J., Ramesh, A., Sprechmann, P., & LeCun, Y. (2016). Disentangling factors of variation in deep representation using adversarial training. In *Advances in neural information processing systems* (pp. 5040–5048).

Matthews, I., & Baker, S. (2004). Active appearance models revisited. *International Journal of Computer Vision*, *60*(2), 135–164.

Neudecker, H. (1969). Some theorems on matrix differentiation with special reference to Kronecker matrix products. *Journal of the American Statistical Association*, *64*(327), 953–963.

Reed, S., Sohn, K., Zhang, Y., & Lee, H. (2014). Learning to disentangle factors of variation with manifold interaction. In E. P. Xing & T. Jebara (Eds.), *Proceedings of machine learning research, proceedings of the 31st international conference on machine learning, PMLR, Bejing, China* (Vol. 32, pp. 1431–1439).

Roemer, F. (2012). Advanced algebraic concepts for efficient multichannel signal processing. Ph.D. thesis, Universitätsbibliothek Ilmenau.

Sagonas, C., Panagakis, Y., Leidinger, A., Zafeiriou, S., et al. (2017). Robust joint and individual variance explained. In *Proceedings of IEEE international conference on computer vision & pattern recognition (CVPR)*.

Shu, Z., Yumer, E., Hadap, S., Sunkavalli, K., Shechtman, E., & Samaras, D. (2017). Neural face editing with intrinsic image disentangling. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

Sivic, J., & Zisserman, A. (2009). Efficient visual search of videos cast as text retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(4), 591–606.

Snape, P., Panagakis, Y., & Zafeiriou, S. (2015). Automatic construction of robust spherical harmonic subspaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 91–100).

Tang, Y., Salakhutdinov, R., & Hinton, G. (2013). Tensor analyzers. In *International conference on machine learning* (pp. 163–171).

Tenenbaum, J. B., & Freeman, W. T. (2000). Separating style and content with bilinear models. *Neural Computation*, *12*(6), 1247–1283. https://doi.org/10.1162/089976600300015349.

Tewari, A., Zollöfer, M., Kim, H., Garrido, P., Bernard, F., Perez, P., & Christian, T. (2017). MoFA: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *IEEE international conference on computer vision (ICCV)*.

Tewari, A., Zollhöfer, M., Garrido, P., Bernard, F., Kim, H., Pérez, P., & Theobalt, C. (2018). Self-supervised multi-level face model learning for monocular reconstruction at over 250 Hz. In *IEEE conference on computer vision and pattern recognition (CVPR)*.

Tran, L., & Liu, X. (2018). Nonlinear 3D face morphable model. In *Proceeding of IEEE computer vision and pattern recognition, Salt Lake City, UT*.

Tran, L., Yin, X., & Liu, X. (2017). Disentangled representation learning gan for pose-invariant face recognition. In *CVPR* (Vol. 4, p. 7).

Vasilescu, M. A. O., & Terzopoulos, D. (2002). Multilinear analysis of image ensembles: Tensorfaces. In *European conference on computer vision* (pp. 447–460). Springer.

Wang, C., Wang, C., Xu, C., & Tao, D. (2017a). Tag disentangled generative adversarial network for object image re-rendering. In *Proceedings of the twenty-sixth international joint conference on artificial intelligence, IJCAI-17* (pp. 2901–2907). https://doi.org/10.24963/ijcai.2017/404.

Wang, M., Panagakis, Y., Snape, P., & Zafeiriou, S. (2017b). Learning the multilinear structure of visual data. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4592–4600).

Woodham, R. J. (1980). Photometric method for determining surface orientation from multiple images. *Optical Engineering*, *19*(1), 191,139.

Worrall, D. E., Garbin, S. J., Turmukhambetov, D., & Brostow, G. J. (2017). Interpretable transformations with encoder–decoder networks. In *IEEE international conference on computer vision (ICCV)*.

Wu, X., He, R., & Sun, Z. (2015). A lightened CNN for deep face representation. arXiv Preprint arXiv:1511.02683.

Yang, F., Wang, J., Shechtman, E., Bourdev, L., & Metaxas, D. (2011). Expression flow for 3D-aware face component transfer. In *ACM transactions on graphics (TOG)* (Vol. 30, p. 60). ACM.

Zafeiriou, S., Atkinson, G. A., Hansen, M. F., Smith, W. A. P., Argyriou, V., Petrou, M., et al. (2013). Face recognition and verification using photometric stereo: The photoface database and a comprehensive evaluation. *IEEE Transactions on Information Forensics and Security*, *8*(1), 121–135. https://doi.org/10.1109/TIFS.2012.2224109.