IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, USA, June 2024.

FSRT: Facial Scene Representation Transformer for Face Reenactment from Factorized Appearance, Head-pose, and Facial Expression Features

Andre Rochow
rochow@ais.uni-bonn.de

Max Schwarz
schwarz@ais.uni-bonn.de

Sven Behnke
behnke@cs.uni-bonn.de

Autonomous Intelligent Systems - Computer Science Institute VI and Center for Robotics, University of Bonn, Germany
Lamarr Institute for Machine Learning and Artificial Intelligence, Germany

Abstract

The task of face reenactment is to transfer the head motion and facial expressions from a driving video to the appearance of a source image, which may be of a different person (cross-reenactment). Most existing methods are CNN-based and estimate optical flow from the source image to the current driving frame, which is then inpainted and refined to produce the output animation. We propose a transformer-based encoder for computing a set-latent representation of the source image(s). We then predict the output color of a query pixel using a transformer-based decoder, which is conditioned with keypoints and a facial expression vector extracted from the driving frame. Latent representations of the source person are learned in a self-supervised manner that factorize their appearance, head pose, and facial expressions. Thus, they are perfectly suited for cross-reenactment. In contrast to most related work, our method naturally extends to multiple source images and can thus adapt to person-specific facial dynamics. We also propose data augmentation and regularization schemes that are necessary to prevent overfitting and support generalizability of the learned representations. We evaluated our approach in a randomized user study. The results indicate superior performance compared to the state-of-the-art in terms of motion transfer quality and temporal consistency.[1]

1. Introduction

Face reenactment is a special case of the motion transfer task [38, 39]. Its objective is to synthesize a realistic facial animation combining the appearance given by one or more images of a source person and the facial expression and head motion of a driving video, which may be of a different person. The driving frame is used to transform the source image to the desired expression and head pose. When the driving video is of the same person (self-reenactment), applications include, e.g., low-bandwidth video conferencing.

[1]Code & Video: https://andrerochow.github.io/fsrt

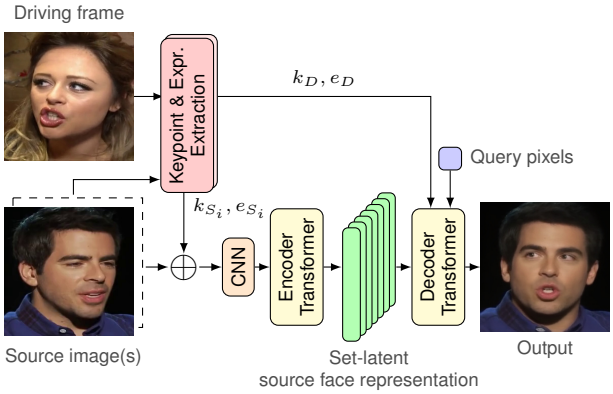

Figure 1. Overview of our method (relative motion transfer). The source image(s) are encoded along with keypoints k_S, capturing head pose, and facial expression vectors e_S to a set-latent representation of the source person. The decoder attends this representation for a query pixel, conditioned on keypoints k_D and a facial expression vector e_D extracted from the driving frame. $\oplus$ denotes pixel-wise concatenation. Images from the VoxCeleb test set [27].

The more interesting and challenging case is when the driving video is of a different person (cross-reenactment) since, if successful, only one or few images of the source person are required to create a realistic facial animation.

Most face reenactment methods are CNN-based [10, 11, 11, 13, 38, 39, 48, 49, 53, 56–58]; and many of these utilize optical flow between the source and driving frames for morphing the source image, followed by a refinement stage [11, 38, 39, 49, 57].

Inspired by recent successes in scene reconstruction [35, 36], we apply a transformer-based architecture to face reenactment that encodes the face of the source person as a set of latent vectors (see Fig. 1). This representation is learned in a self-supervised way. We then sample each target pixel location with a transformer-based decoder conditioned on keypoints and an expression vector that are extracted from the driving frame. The learned set-latent representation of the source person factorizes their head pose, appearance, and facial expression, which enables accurate head motion

and facial expression transfer, also for cross-reenactment.

While our method yields state-of-the-art results in absolute motion transfer (i.e., when the driving keypoints are used unmodified), it especially increases robustness in the case of relative motion transfer [38], a mode which reduces unwanted leaking of face shape from the driving frame. Relative motion transfer is initialized by finding a source and driving frame with well-matched head poses and expressions and then operates by applying motions to the source frame in relative fashion.

Many methods encode facial expression by keypoints, which are augmented with local spatial transformations [11, 38, 57]. Here, relative motion is encoded as relative transforms from the initial driving frame and is applied in the source frame. Of course, this is highly sensitive to the initialization. By decoupling head pose from expression and describing expression in an absolute manner, our approach reduces the influence of initialization on expression transfer.

Finally, we note that our approach makes few assumptions since there is no explicit model of motion or correspondence. Instead, the set-latent representation of the source person is learned in a self-supervised way using conditioning with keypoints and a facial expression vector from the source on the encoder and from the driving frame on the decoder, respectively.

In summary, our contributions include:

1. A novel transformer-based architecture for face reenactment which learns a global set-latent representation of source images and allows rendering conditioned on keypoints and expression vectors,
2. a latent expression description invariant to appearance, head shape, and head pose, greatly improving cross-reenactment,
3. augmentation and regularization methods for training that support the separation of facial expression information (expression vector), appearance (set-latent representation), and head pose (keypoints) without overfitting,
4. application of adversarial and perceptual losses to scene representation transformers, which greatly improves realism and sharpness,
5. a detailed evaluation, where we outperform state-of-the-art in motion transfer quality and temporal consistency, including a user study which also shows superiority in subjective preference trials.

2. Related Work

Face Image Synthesis. Face image synthesis deals with the generation of new non-existing faces [12] even from text input [14, 22], completion of missing facial regions of known faces [21, 61], or manipulation of expression and appearance of known faces in manual [14, 16, 19] or automatic manner [4, 6, 30, 37].

In contrast to these methods, our approach combines the continuously changing head pose and expression from the driving video with the appearance of the source, so that a natural and temporally consistent video stream is produced.

Talking Head Synthesis and Face Reenactment. Talking Head Synthesis aims to make head poses, emotions, and especially speech controllable. Here, lip movement is mainly reconstructed from audio [25, 32, 44–47, 53, 55, 59, 60]. Closely related, face reenactment [41], which is a motion transfer task, aims to apply the motion given by a driving frame to the appearance defined by a source image. An even more challenging problem is VR facial animation, where the driving face is additionally occluded by a head-mounted display (HMD) [24, 31, 33, 34, 42, 52]. Here, especially the alignment problem between facial images and mouth camera images makes it difficult to generate training data.

Generally, there are different types of driving information being used. Where some methods only utilize facial keypoints [13, 39, 48, 56, 58], other methods additionally use image features from a driving frame [10, 11, 38, 40, 49, 57]. Also, audio can be used if available [1].

Some methods [13, 58] utilize external 3D keypoints extraction [3] for face reenactment. Hsu et al. [13] use a separate generator to predict more accurate driving keypoints, given initial keypoints and a source image. Siarohin et al. [39] learn keypoints self-supervised and use them to predict a deformation grid of the source image into the driving keypoints. To resolve keypoint ambiguities, Siarohin et al. [38] estimate local affine transformations into a canonical space for each keypoint area. This allows far more facial expressions to be represented. Based on [38], Hong et al. [11] learn depth maps, which they use to predict more accurate keypoints and dense depth-aware attention maps, which can attend to important semantic facial areas. Zhao and Zhang [57] use a motion estimation based on thin-plate spline transformations to produce a more flexible optical flow. They use multi-resolution depth maps and occlusion masks to inpaint missing regions more realistically.

However, driving motion does not necessarily has to be described by keypoints [29, 40, 41, 53]. Wiles et al. [53] directly predict the sampling coordinates to a canonical embedding of a face. A separate driving network then predicts the mapping from the embedded face to the driving frame. Siarohin et al. [40] bypass the keypoint estimation step from predicted heatmaps and consider them as regions. They estimate the principal components of the region to predict an in-plane rotation and scaling, which is used to estimate a more accurate pixel-wise optical flow. Pang et al. [29] learn a disentanglement of pose and expression, so that different driving sources can be used. Very recently, Li et al. [20] learned to embed a source image into a canonical volume and predict the deformation of individual sampled rays to estimate the optical flow.

Wang et al. [50] and Gong et al. [8] replace the motion network proposed by Siarohin et al. [39] with custom modules (Linear Motion Decomposition and a transformer module enabling domain switching, respectively), but remain fundamentally based on CNNs (in encoder and decoder) and a warp-and-refine architecture.

Unlike related methods, we use a transformer-style architecture to predict a latent scene representation of the source images and learn expression vectors which are decoupled from appearance and head pose information. Instead of modeling optical flow and motion explicitly, we learn to attend the latent scene representation, with keypoints and latent expression vectors extracted from a driving frame. This allows us to generate results of higher accuracy, while significantly improving the temporal consistency.

Scene Representation Transformers. While transformers [43] were originally developed for natural language processing, vision transformers have also been highly successful [7, 23]. Recently, Sajjadi et al. [35] proposed Scene Representation Transformers (SRT) to learn an internal scene representation encoded in a set of latents vectors. Given these latent vectors and a camera pose, SRT allow novel-view rendering without explicitly modeling the scene geometry. Based on this, Sajjadi et al. [36] propose a slot attention module to instead predict an object-centric slot scene representation, in which different objects are separated without any supervision.

In this work, we reformulate SRT [35] for the face reenactment task and demonstrate how to efficiently model and query dynamics in the learned face representation. Unlike [35], we aim to reconstruct photorealistic faces. To this end, we propose training with perceptual [17] and adversarial losses [9], which significantly improves the output quality.

3. Method

The SRT architecture [35] encodes one or more posed images to an internal representation and reconstructs views from arbitrary viewpoints. We adapt the architecture and the input representation in such a way that we learn an internal representation from one or more unposed facial images. Reconstruction then allows free choice of head pose and facial expression. Given a set-latent representation of an encoded face, head pose and facial expression can be controlled by ten keypoints and a latent expression vector. Abstractly, the internal representation can also be understood as an embedding that separates the appearance of a person from the head pose and expression.

Given an input representation $\{R_{S_i}\}$ (see Sec. 3.1), the transformer encoder $\mathcal{E}$ (Fig. 2 and Sec. 3.1) produces a set-latent scene representation

$$\{z_z \in \mathbb{R}^d\} = \mathcal{E}(\text{CNN}(\{R_{S_i}\})), \tag{1}$$

where CNN (Sec. 3.1) is a convolutional feature extractor backbone (shared in case of multiple input images), as proposed by [35]. Given this set-latent representation and the query $Q_{I_D}(q)$ (see Sec. 3.1), the transformer decoder $\mathcal{D}$ predicts the pixel color

$$C(q) = \mathcal{D}(Q_{I_D}(q) \mid \{z_z\}). \tag{2}$$

Our full architecture is visualized in Fig. 2.

3.1. Input and Query Representation

Given are one more facial source images I_{S_i}. We encode each image through ten keypoints k_{S_i}, computed by a keypoint detector, and a latent expression vector e_{S_i} which we learn in self-supervised manner. The keypoints are normalized to $(-1, 1)$ and positionally encoded [26]

$$f(p, s_O, O) = \bigoplus_{m=s_O}^{s_O+O-1} \sin(2^m \pi p) \oplus \cos(2^m \pi p) \tag{3}$$

to obtain

$$\gamma_{\text{key}}(k_{S_i}) = \bigoplus_{j=0}^{n_\mathcal{K}} f\left(k_{S_i}^{(j)}, s_{O_{\text{key}}}, O_{\text{key}}\right) \tag{4}$$

where $n_\mathcal{K}$ is the number of keypoints, O_{key} is the number of octaves per keypoint, $s_{O_{\text{key}}}$ is the keypoint start octave, and $\bigoplus$ is the vector concatenation.

During face reenactment, keypoints might move out of the image boundaries $(-1, 1)$. Due to this, we set $s_{O_{\text{key}}} = -1$ to add an additional negative octave, which extends the interval of uniquely encodable values to $(-2, 2)$. We generate training samples with keypoints outside the image by estimating them before cropping.

Similar to the keypoints, each image pixel with normalized coordinate $p = (x, y)$ receives a 2D positional encoding [26]

$$\gamma_{\text{pix}}(p, O_{\text{pix}}, s_{O_{\text{pix}}}) = f\left(p, s_{O_{\text{pix}}}, O_{\text{pix}}\right). \tag{5}$$

The input representation of the source images I_{S_i} at pixel $p = (x, y)$ is then given by

$$R_{S_i}(p) = [c_p, \gamma_{\text{pix}}(p), \gamma_{\text{key}}(k_{S_i}), e_{S_i}], \tag{6}$$

where c_p is the RGB color at pixel p and e_{S_i} is the latent expression vector extracted from I_{S_i} (see Sec. 3.1). Note that each pixel is conditioned with the full keypoint encoding and the full latent expression vector, which quickly leads to a large input representation.

The decoder is queried for every output pixel $q = (x', y')$. Instead of the camera pose, as in [35], each positionally-encoded query pixel is additionally conditioned with the desired target keypoints k_D (i.e. the driving keypoints) and the latent expression vector e_D of the

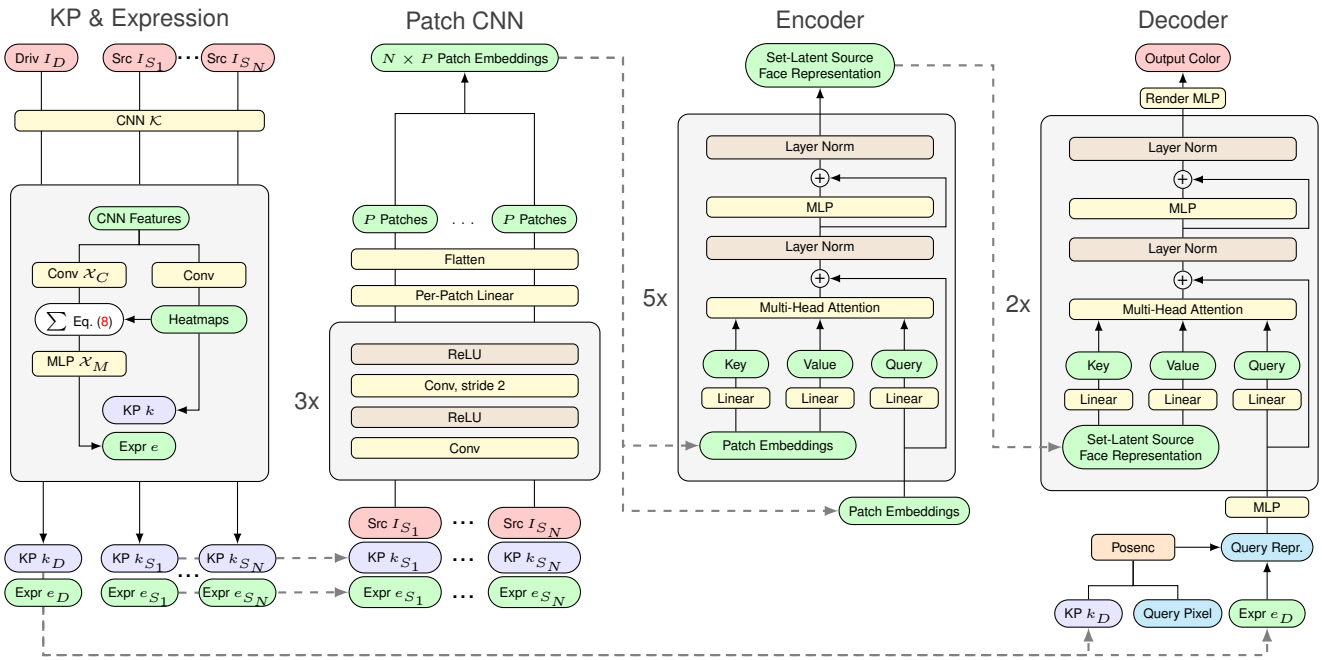

Figure 2. Architecture details. Given the driving frame and source images, we extract facial keypoints and latent expression vectors. Extracted source information are used to generate the input representation of the Patch CNN. The encoder infers the set-latent source face representation from the patch embeddings as in SRT [35]. The decoder is applied for each query pixel individually and is conditioned with the driving keypoints and the latent driving expression vector. For further implementation details we refer to the Supplementary Material.

desired expression (i.e. the latent expression vector of the driving frame).

Hence, the pixel-wise query representation is

$$Q_{I_D}(q) = [\gamma_{\text{pix}}(q), \gamma_{\text{key}}(k_D), e_D]. \tag{7}$$

Intuitively, the decoder attends to the most important features of the representation $\{z_z\}$ to render the source image appearance with the desired motion given by k_D and e_D.

Note that our method does not require camera extrinsics or intrinsics, since we operate directly in pixel space.

Keypoint Detector. The keypoint detector $\mathcal{K}$ is a fully convolutional hourglass network [28], as proposed by Siarohin et al. [38, 39]. For each input image I, it predicts heatmaps $H_I^{(i)} \in [0,1]^{H \times W}$, $i = 1, \ldots, n_{\mathcal{K}}$, which define the pixelwise presence confidence of keypoint k_i. For all experiments, we fix the number of keypoints to $n_{\mathcal{K}} = 10$.

Expression Network. We assume that the last feature maps $F_{I,\mathcal{K}}$ predicted by the keypoint detector capture local image properties. Given this assumption we build an expression network $\mathcal{X}$ that recycles $F_{I,\mathcal{K}}$ and the predicted keypoint heatmaps H_I for an image I (see Fig. 2). We filter $F_{I,\mathcal{K}}$ by a learned 7×7 convolution $\mathcal{X}_C$, producing $F_{I,\mathcal{X}}$ with shape $[n_f, n_{\mathcal{K}}, h', w']$. To focus on facial features, we utilize H_I to aggregate the features spatially for each keypoint k_i:

$$\vec{f}_{I,\mathcal{X}}^{(i)} = \bigoplus_{c=1}^{64} \left[\sum_{y=0}^{h'} \sum_{x=0}^{w'} H_I^{(i)}(x,y) F_{I,\mathcal{X}}^{(c)}(i,x,y) \right] \in \mathbb{R}^{n_f} \tag{8}$$

to obtain

$$f_{I,\mathcal{X}} = \bigoplus_{i=1}^{n_{\mathcal{K}}} \left[\vec{f}_{I,\mathcal{X}}^{(i)} \right] \in \mathbb{R}^{n_{\mathcal{K}} \cdot n_f}, \tag{9}$$

where c is the channel index, i is the keypoint index, and $\bigoplus$ is the concatenation operation. Using a 4-layer MLP $\mathcal{X}_M$, we compute the latent expression vector

$$e_I = \mathcal{X}_M \left(f_{I,\mathcal{X}} \right). \tag{10}$$

The expressional information of all important keypoints areas are thus spread throughout e_I. Additionally, keypoint regions that do not contain important expression information can be filtered out by combining the local information.

Patch CNN. As in [35], the shared CNN is designed to reduce the spatial dimension of the input data and fuse patch information. In each block, the height and width are reduced by factor two and the number of feature maps is doubled. For all experiments, we use three CNN blocks followed by a final convolution with kernel size one, which generates the number of feature maps $n_{\mathcal{E}}^{fm}$

needed for the transformer encoder. The features with shape $[bs, n_{\mathcal{E}}^{fm}, H/8, W/8]$ are then reshaped to $[bs, \frac{H+W}{8}, n_{\mathcal{E}}^{fm}]$ which is the patch embedding input to the encoder.

Encoder. Following [35], the standard transformer alternates global self-attention (between all tokens) and small MLP networks (see Fig. 2). Following [36], we drop source ID embeddings and reduce the number of attention blocks to five. Through self-attention across all patch embeddings, the encoder learns a set-latent scene representation $\{z_z\}$ in which each vector z_z captures global scene and dynamics information. Note that the cardinality of the set-latent representation scales linearly with the number of source images.

Decoder. The transformer decoder does not use self-attention, but instead attends the set-latent scene representation with the query Q_{I_D}. This is repeated for two times, followed by a render MLP that predicts the final output color at a certain pixel location. For better performance, the query is first fed through a small 2-layer MLP that spreads the information in all dimensions, as proposed by [36]. Furthermore, we also use a final 5-layer render MLP that predicts the output color given the output of the attention module. Intuitively, the decoder learns to attend to the most important features of $\{z_z\}$ to infer the pixel information of the encoded facial image with the requested head pose and facial expression. Note that unlike SRT, we do not only request a novel view of a static scene but also certain dynamics within the scene, such as mouth movement. It is therefore necessary for the encoder to output a highly flexible scene representation (see experiments with small decoder in Sec. 4.2).

Due to the transformer design, the decoder can handle $\{z_z\}$ of any cardinality. Thus, a trained encoder and decoder can operate on a flexible number of source images.

For a given source face, we only need to predict the set-latent scene representation once and each query pixel is estimated independently of the others. This is an advantage over CNN-based approaches [11, 38, 39, 49, 57], because it allows the model throughput to be scaled linearly with the number of available GPUs. Only one copy of the decoder needs to reside on each GPU.

3.2. Augmentation and Regularization

Ideally, the network should learn to decouple appearance, pose, and expression information into set-latents z_z, keypoints k_I, and expression vector e_I, respectively. This separation of concerns enables cross-reenactment. In practice, the method is prone to overfitting, since we can only train in the self-reenactment regime, where ground truth is available. This results in latents that jointly encode appearance, pose and expression, which is visible when cross-reenacting to a different person. Artifacts appear in the background area around the face and the model also deforms the source person to be closer to the face shape of the driving frame (see Fig. 3). Hence, we do not reach the intended separation level. To combat this, we implement several data augmentation and regularization measures.

Color Augmentation. To prevent colors leaking from the driving frame to the output image, we apply color jitter augmentation on the source images. Specifically, we create two color-jittered versions I_S^{A1}, I_S^{A2} of the input image I_S. The expression network $\mathcal{X}$ is run to extract expression vectors $e_{I_S^{A1}}, e_{I_S^{A2}}$. An additional regularization term is added to enforce invariance to color jitter:

$$\mathcal{L}_{\text{aug}} = \frac{1}{|e|} \left\| e_{I_S^{A1}} - e_{I_S^{A2}} \right\|_2^2. \tag{11}$$

While the encoder $\mathcal{E}$ is always trained with RGB colors from I_S^{A1}, it receives the expression vector $e_{I_S^{A2}}$. This further improves color invariance.

Cropping. To reduce background information in the output of $\mathcal{X}$, we further randomly crop the driving frame (just for $\mathcal{X}$). Here, we define $\Omega(\cdot)$, which selects a random crop with awareness of facial keypoints as proposed by [3]. This crop is then scaled back to the original size, which can change the aspect ratio. We add a loss term

$$\mathcal{L}_{\text{aug,D}} = \frac{1}{|e|} \left\| e_{I_D^A} - e_{\Omega\left(I_D^{A3}\right)} \right\|_2^2 \tag{12}$$

on the expression vectors of color-jittered versions I_D^A, I_D^{A3}, in which A is either A_1 or A_2. Adding $\Omega(\cdot)$ to the loss term encourages that $\mathcal{X}$ extracts scale-invariant expression information only from the face region. Primarily, the expression vector of $\Omega\left(I_D^{A3}\right)$ is also utilized for decoding. However, in 25% of cases, $e_{I_D^A}$ is selected, which employs the same color-jittering (A_1 or A_2) applied to the source images.

Statistical Regularization. Data augmentation alone is not enough to completely prevent head pose, expression, and appearance information from being jointly encoded (see Fig. 3). We take inspiration from VICReg [2], a method for regularization of unsupervised feature learning based on invariance, variance, and covariance, but adapt it to encourage the focus on expression information. Invariance against augmentations is already covered by Eqs. (11) and (12).

The covariance part aims to decorrelate along the feature dimension. Intuitively, decorrelation encourages separation of expression from head pose, shape, and appearance and enables the network to drop non-expressional information (which is already encoded in keypoints and scene representation $\{z_z\}$). Given a batch of source and driving images concatenated in the batch dimension

$$\begin{aligned} E = [&e_{S_1}^{(1)}, \ldots, e_{S_{(n_{src})}}^{(1)}, e_D^{(1)}, \\ &\vdots \\ &e_{S_1}^{(bs)}, \ldots, e_{S_{(n_{src})}}^{(bs)}, e_D^{(bs)}] \end{aligned} \tag{13}$$

with shape $[(n_{src}+1)bs, c]$, we estimate the covariance of the individual dimensions $\mathrm{Cov}(E)$. The covariance loss is

$$\mathcal{L}^E_{\text{Cov}} = \frac{1}{c}(\underbrace{\sum_{i\neq j}[\mathrm{Cov}(E)]^2_{i,j}}_{\text{off diagonal}} + \underbrace{\sum_{k}[\mathrm{Cov}(E)]^2_{k,k}}_{\text{diagonal (variance)}}). \quad (14)$$

In contrast to VICReg, we minimize the diagonal variance terms as well, which represents an additional information bottleneck. In experiments, this regularization was helpful.

Since we train with supervision, there is no risk of ending up in a mode collapse, so the batch-variance criterion of VICReg is not required. Conversely, we found that encouraging variance *along the feature dimension* of each vector with a hinge loss (penalizing vanishing features)

$$\mathcal{L}^E_{\text{Var}} = \frac{1}{|E|}\sum_{e\in E}\max\left(0,\ 1-\sqrt{\mathrm{Var}(e)+\epsilon}\right), \quad (15)$$

leads to better results and a more stable training. Finally, we define $\mathcal{L}_{\text{Cov}}{=}\mathcal{L}^{E_1}_{\text{Cov}}{+}\mathcal{L}^{E_2}_{\text{Cov}}$ and $\mathcal{L}_{\text{Var}}{=}\mathcal{L}^{E_1}_{\text{Var}}{+}\mathcal{L}^{E_2}_{\text{Var}}$, where E_1 and E_2 are the differently augmented variants of E.

3.3. Training

We use the VoxCeleb dataset [27] and prepare it as suggested by Siarohin et al. [38]. It consists of ∼3000 videos from 419 different identities divided into a total of ∼17.000 utterances with a resolution of 256×256. During training, we sample n_{src} source frames and one driving frame from the same video. Keypoints are extracted using the detector network of [38], which is not trained further.

We train the rest of our method in three distinct phases. In all phases, we apply the regularization loss

$$\mathcal{L}_{\text{reg}} = \tfrac{\lambda_{\text{aug}}}{2}\left(\overline{\mathcal{L}_{\text{aug}}} + \overline{\mathcal{L}_{\text{aug,D}}}\right) + \lambda_{\text{Cov}}\mathcal{L}_{\text{Cov}} + \lambda_{\text{Var}}\mathcal{L}_{\text{Var}}, \quad (16)$$

where $\overline{\mathcal{L}_{\text{aug}}}$ and $\overline{\mathcal{L}_{\text{aug,D}}}$ are the mean values of $\mathcal{L}_{\text{aug}}$ and $\mathcal{L}_{\text{aug,D}}$ calculated over the entire batch. We start in Phase I with warm-up training, optimizing the MSE loss [35]

$$\mathcal{L}_{\text{MSE}} = \mathbb{E}_{q\sim I_D}\|\mathcal{D}(q) - I_D(q)\|_2^2, \quad (17)$$

where we approximate $\mathbb{E}_{q\sim I_D}$ with 4096 sampled pixels.

Using only pixel-wise losses leads to blurry images (see Fig. 3). To address this issue, we propose to add the perceptual loss $\mathcal{L}_P$ [17] in Phase II to generate more details. During our experiments, we found that the batch size must be large enough to avoid local minima and poor performance. Since training on the full frames already exceeds 80GB with a batch size of four, we compute gradients only sub-sampled to 128^2 pixels and compute the remaining pixels without gradient information. We apply a random pixel offset to ensure that all positions are covered during training. This trick allows us to estimate image-based losses without requiring costly gradient estimation for the entire image.

Finally, in Phase III, we then add adversarial losses $\mathcal{L}_A$, which guide the model to predict realistic images. Following Siarohin et al. [38], we utilize a CNN-based keypoint-aware discriminator $\mathcal{A}$ with 4 blocks and also add a feature matching loss $\mathcal{L}^F_A$ between the discriminator maps predicted from the generated image and the ground truth image.

The final loss in phase three is thus:

$$\mathcal{L} = \mathcal{L}_{\text{reg}} + \lambda_{\text{MSE}}\mathcal{L}_{\text{MSE}} + \lambda_P\mathcal{L}_P + \lambda_A\mathcal{L}_A + \lambda^F_A\mathcal{L}^F_A. \quad (18)$$

In our experiments, we train with a batch size of 24 on three NVIDIA A100 GPUs (80GB), for 200k iterations in Phase I, 300k iterations in Phase II, and approximately 500k iterations in Phase III, depending on the validation performance (see Suppl. Material). We set $\lambda_{\text{MSE}} = 1$, $\lambda_P = 0.01$, $\lambda_A = 0.001$, $\lambda^F_A = 0.01$, $\lambda_{\text{aug}} = \lambda_{\text{Cov}} = 1$, and $\lambda_{\text{Var}} = 0.2$. Especially Phase I is very important to avoid overfitting. When skipped, we experience extremely slow training progress and easily end up in a bad local minimum.

3.4. Inference

For self-reenactment, the inference follows the training pipeline. In contrast, for cross-reenactment, the driving frame comes from a different person. This means that keypoints k_D can be taken as-is (absolute motion transfer) or adapted (relative motion transfer). This adaption is calibrated from a selected driving frame that best matches the head pose and expression (measured through the normalized keypoints of Bulat and Tzimiropoulos [3]). Following Siarohin et al. [38], the scale is estimated by comparing the volume of the convex hull of head keypoints. Driving keypoint movement is then scaled correctly and added to the keypoints of the source image.

In both cases, the facial expression vector does not depend on pose, shape, or appearance and is applied as-is, which is a particular advantage of our method.

4. Experiments

In this section, we carry out various experiments on the official VoxCeleb test dataset [27] with image size 256^2. Additional results are reported in the Supplementary Material.

4.1. Self-reenactment

In self-reenactment, the source image is selected as the first frame in the driving video. In the case of two source images, we also select the last frame. We then reconstruct every tenth frame within the video, ensuring that each driving frame is at least ten frames apart from the closest source image. We compare the animations to ground truth using the Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) [51], mean L1 error, and Average Keypoint Distance (AKD). To compute the AKD, we utilize external facial keypoints provided by Bulat and Tzimiropoulos [3].

Method	#KP	SSIM↑	PSNR↑	L1↓	AKD↓
DPE [29]	0	.7180	22.94	.0484	3.07
FOMM [38]	10	.7310	22.90	.0470	2.26
DaGAN [11]	15	.7563	23.51	.0450	2.10
DaGAN/dv2 [11][1]	15	.7346	22.81	.0493	2.50
OSFS [49][2]	15	.7327	22.97	.0471	2.33
TSMM [57]	50	.7660	23.76	.0433	**2.00**
Ours/2-Src	10	**.7891**	**25.00**	**.0360**	2.04
Ours	10	.7576	23.67	.0421	2.13
$\|e\| = 128$	10	.7558	23.56	.0428	2.16
$\|e\| = 64$	10	.7535	23.61	.0430	2.18
small $\mathcal{D}$	10	.7548	23.60	.0430	2.17
$n_{\mathcal{K}} = 0$	0	.7445	23.56	.0436	2.64

Table 1. Self-reenactment results (including ablations) on the official VoxCeleb test set [27]. Underlined values are the second best.

[1] Uses depth network trained on VoxCeleb2 [5] for inference
[2] Third-party implementation

In Tab. 1, we compare with related methods. Our multi-source ablation outperforms related methods in terms of accuracy. For single source images, we achieve state-of-the-art performance. While TSMM [57] slightly outperforms our method in SSIM, AKD, and PSNR, we note that they inpaint only disoccluded regions of the detected background. This produces nearly perfect reconstructions in static background areas that are also visible in the source image. Furthermore, our method generalizes much better for cross-reenactment and produces more temporally consistent animations, as highlighted with our user study (see Tab. 2).

While a low AKD and high SSIM value are good for self-reenactment, they often indicate that face shapes predicted by a model are highly dependent on the driving face structure. This, however, is detrimental for cross-reenactment, where the source appearance may be distorted by poorly matching driving keypoints (see shape deformations of related methods in Fig. 4). Also, for relative motion, the alignment assumption (explained in Sec. 3.4) is often not perfectly satisfied, leading to poorly matching keypoints. Our method is more robust to this (see Fig. 5), because we do not use the structure of the driving frame to estimate the optical flow and we encode appearance information invariant to the driving keypoints in the set-latent scene representation.

4.2. Ablation Study

We run an ablation study to compare quantitative results (see Tab. 1). A qualitative comparison and implementation details are reported in the Supplementary Material.

Do we need keypoints? We report an ablation without keypoint encoding (Ours/$n_{\mathcal{K}} = 0$), i.e. all pose information is carried implicitly in the latent vector e, removing factorization of pose and expression, which makes relative motion transfer impossible. As can be seen in Tab. 1, this change results in worse self-reenactment performance. See Suppl. Material for qualitative cross-reenactment comparisons.

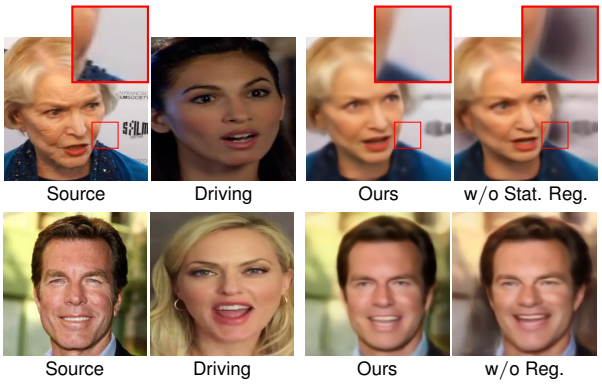

Figure 3. Regularization benefit in Phase I training (relative motion transfer). If trained without statistical regularization (w/o Stat. Reg.), artifacts originating from the driving frame are visible in the background around the face boundary. When dropping regularization entirely (w/o Reg.), color distortions, background artifacts, and shape deformations are clearly visible. The lower sequence uses a source image from the CelebA-HQ dataset [18].

What size should latent vectors have? Our reference model is trained with one source image and a latent expression vector of size $|e|$=256. As mentioned in Sec. 3.2 and visualized in Fig. 3, training without our proposed regularization leads to overfitting, resulting in shape deformation, color distortion, and background artifacts. With $|e|$=128 and $|e|$=64 latent expression dimensions, the self-reenactment performance decreases only slightly, showing that we can significantly reduce the amount of driving information being transmitted (e.g. for low-bandwidth videoconferencing) without losing much accuracy. In cross-reenactment, we also noticed a slight degradation in the transmission of facial expressions. However, the results are still good.

How efficient is the set-latent representation for decoding? We train a model (Ours/small $\mathcal{D}$) with a significantly smaller decoder (see Supplementary Material). Interestingly, we achieve a self-reenactment performance close to the reference model. This indicates that the set-latent representation is very efficient to decode and already models facial dynamics. However, the sharpness of fine details, such as hair, is slightly degraded. Reducing the decoder capacity increases throughput from 11 fps to 23 fps, enabling real-time application on a single NVIDIA RTX 4090 GPU. As mentioned in Sec. 3.1, we can scale throughput linearly with the number of GPUs without introducing additional latency.

Can we improve results with multiple source images? Unlike state-of-the-art methods [11, 29, 38, 49, 57], our architecture allows the use of an arbitrary and flexible number of source images when available (e.g., when extracted from a video). Multiple source images can help the model understand person-specific face dynamics. In this experiment, we train a model ablation (Ours/2-Src) with two source images. As Tab. 1 shows, the results are significantly im-

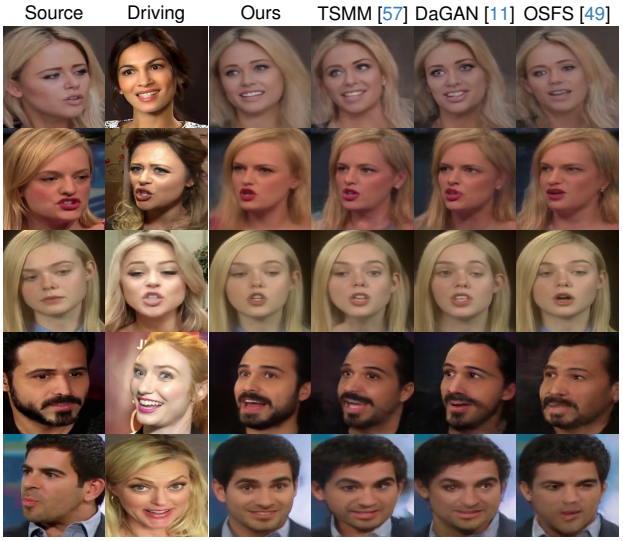

Figure 4. Cross-reenactment comparison with absolute motion transfer on the VoxCeleb test set [27]. We generate more accurate expressions with less shape deformations (higher ID preservation).

	FOMM[38]	DaGAN[11]	[11]/dv2	TSMM[57]	OSFS[49]	DPE[29]
Rel.	97% (20)	98% (20)	95% (19)	97% (20)	87% (19)	
Abs.	94% (20)	99% (20)	96% (20)	92% (19)	94% (19)	95% (20)

Table 2. Cross-reenactment user study. Pairwise preferences between SOTA and our method. Higher values show higher preference for our videos. DPE [29] has no relative mode. (·) shows the amount out of 20 scenes for which we got the majority of votes.

proved. Interestingly, the model generalizes to more than two source images even without explicit training.

4.3. Cross-reenactment

Our main motivation is to perform cross-reenactment. We sample 20 source images and driving videos from the official VoxCeleb [27] test set and compare our videos to state-of-the-art animations in a pairwise user study in Tab. 2. To be fair, we only use a single source image. We also present qualitative results in Figs. 4 and 5 and report additional user study information in the Supplementary Material. In general, our method is better at cross-ID motion transfer, while producing more consistent and natural results. For additional qualitative results on CelebV [54], CelebA-HQ [18], and VoxCeleb2 [5] see our Supplementary Material.

Absolute Motion. When the driving keypoints are simply copied, users mainly prefer the animations generated by our method (see Tab. 2). Since our method is more robust to poorly matching keypoints, we produce fewer shape deformations than other keypoint-based methods (see Fig. 4). Furthermore, we consistently animate larger pose offsets.

Relative Motion. More interesting and challenging is animating with relative motion. Here, best performance can be achieved when the facial expression representation is decoupled from head pose and shape. As Tab. 2 illustrates, we significantly outperform state-of-the-art methods. When analyzing the results, we noticed that related methods show poor performance when there is no good match for the source expression and head pose in the driving video.

Figure 5. Comparison with SOTA in cross-reenactment with relative motion transfer. Our method is more robust to the alignment assumption for relative motion transfer, generates more accurate expressions, and handles larger pose offsets. All images are from the VoxCeleb test set [27], except the lower block, which shows generalization to source images from the CelebA-HQ dataset [18].

5. Limitations

Our method struggles to generate out-of-distribution expressions such as sticking out the tongue or looking back. While we produce sharper mouth and eye regions, details in the background and hair are sometimes slightly reduced, compared to CNN-based methods. We believe that the model allocates most of its capacity to the face. Compared to CNN approaches that simply learn to forward background pixels from the input, our model must encode the background in the set-latents and reconstruct it by attending the correct features. Increasing model capacity or optimizing the query representation might lead to improvements.

6. Conclusion

We have proposed a state-of-the-art method for face reenactment. To our knowledge, this is the first transformer-based architecture for this purpose. We learn latent expression features that are free of appearance, shape or pose information, making them perfectly suited for cross-reenactment. Our method achieves fast inference speed, which allows real-time application. We proposed a regularization and training scheme which are necessary to guide the network to represent the scene as desired. Future work could investigate further improving the animation quality of fine details (e.g. in the hair) and utilizing volume rendering techniques to reconstruct geometry.

References

[1] Madhav Agarwal, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. Audio-visual face reenactment. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5178–5187, 2023. 2

[2] Adrien Bardes, Jean Ponce, and Yann Lecun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022. 5

[3] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks). In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017. 2, 5, 6

[4] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8789–8797, 2018. 2

[5] J. S. Chung, A. Nagrani, and A. Zisserman. VoxCeleb2: Deep speaker recognition. In *Conference of the International Speech Communication Association (INTERSPEECH)*, 2018. 7, 8, 3, 4, 9, 10, 11, 12

[6] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3D imitative-contrastive learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5154–5163, 2020. 2

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 3

[8] Yuan Gong, Yong Zhang, Xiaodong Cun, Fei Yin, Yanbo Fan, Xuan Wang, Baoyuan Wu, and Yujiu Yang. ToonTalker: Cross-domain face reenactment. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7690–7700, 2023. 3

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 3

[10] Sungjoo Ha, Martin Kersner, Beomsu Kim, Seokjun Seo, and Dongyoung Kim. MarioNETte: Few-shot face reenactment preserving identity of unseen targets. In *AAAI Conference on Artificial Intelligence*, pages 10893–10900, 2020. 1, 2

[11] Fa-Ting Hong, Longhao Zhang, Li Shen, and Dan Xu. Depth-aware generative adversarial network for talking head video generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3397–3406, 2022. 1, 2, 5, 7, 8, 3, 9, 10, 11

[12] Xianxu Hou, Ke Sun, Linlin Shen, and Guoping Qiu. Improving variational autoencoder with deep feature consistent and generative adversarial training. *Neurocomputing*, 341: 183–194, 2019. 2

[13] Gee-Sern Hsu, Chun-Hung Tsai, and Hung-Yi Wu. Dual-generator face reenactment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 642–650, 2022. 1, 2

[14] Ziqi Huang, Kelvin CK Chan, Yuming Jiang, and Ziwei Liu. Collaborative diffusion for multi-modal face generation and editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6080–6090, 2023. 2

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134, 2017. 1

[16] Youngjoo Jo and Jongyoul Park. SC-FEGAN: Face editing generative adversarial network with user's sketch and color. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1745–1753, 2019. 2

[17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, pages 694–711. Springer, 2016. 3, 6

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018. 7, 8, 4, 9, 11

[19] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. MaskGAN: Towards diverse and interactive facial image manipulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[20] Weichuang Li, Longhao Zhang, Dong Wang, Bin Zhao, Zhigang Wang, Mulin Chen, Bang Zhang, Zhongjian Wang, Liefeng Bo, and Xuelong Li. One-shot high-fidelity talking-head synthesis with deformable neural radiance field. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17969–17978, 2023. 2

[21] Yijun Li, Sifei Liu, Jimei Yang, and Ming-Hsuan Yang. Generative face completion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3911–3919, 2017. 2

[22] Xinmiao Lin, Yikang Li, Jenhao Hsiao, Chiuman Ho, and Yu Kong. Catch missing details: Image reconstruction with frequency augmented variational autoencoder. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1736–1745, 2023. 2

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021. 3

[24] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *ACM Transactions on Graphics (ToG)*, 37(4):1–13, 2018. 2

[25] Yifeng Ma, Suzhen Wang, Zhipeng Hu, Changjie Fan, Tangjie Lv, Yu Ding, Zhidong Deng, and Xin Yu. StyleTalk: One-shot talking head generation with controllable speaking styles. In *AAAI Conference on Artificial Intelligence*, pages 1896–1904, 2023. 2

[26] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3

[27] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. VoxCeleb: A Large-Scale Speaker Identification Dataset. In *Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2616–2620, 2017. 1, 6, 7, 8, 2, 3, 4, 5, 12

[28] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision (ECCV)*, pages 483–499, 2016. 4, 1

[29] Youxin Pang, Yong Zhang, Weize Quan, Yanbo Fan, Xiaodong Cun, Ying Shan, and Dong-ming Yan. DPE: Disentanglement of pose and expression for general video portrait editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436, 2023. 2, 7, 8, 3, 11

[30] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. GANimation: Anatomically-aware facial animation from a single image. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[31] Alexander Richard, Colin Lea, Shugao Ma, Juergen Gall, Fernando De la Torre, and Yaser Sheikh. Audio-and gaze-driven facial animation of codec avatars. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 41–50, 2021. 2

[32] Alexander Richard, Michael Zollhöfer, Yandong Wen, Fernando De la Torre, and Yaser Sheikh. MeshTalk: 3D face animation from speech using cross-modality disentanglement. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1173–1182, 2021. 2

[33] Andre Rochow, Max Schwarz, Michael Schreiber, and Sven Behnke. VR facial animation for immersive telepresence avatars. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2167–2174, 2022. 2

[34] Andre Rochow, Max Schwarz, and Sven Behnke. Attention-based VR facial animation with visual mouth camera guidance for immersive telepresence avatars. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1276–1283, 2023. 2

[35] Mehdi SM Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, et al. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6229–6238, 2022. 1, 3, 4, 5, 6

[36] Mehdi S. M. Sajjadi, Daniel Duckworth, Aravindh Mahendran, Sjoerd van Steenkiste, Filip Pavetić, Mario Lučić, Leonidas J. Guibas, Klaus Greff, and Thomas Kipf. Object scene representation transformer. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 1, 3, 5

[37] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of GANs for semantic face editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9243–9252, 2020. 2

[38] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 1, 2, 4, 5, 6, 7, 8, 3, 9, 10

[39] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2377–2386, 2019. 1, 2, 3, 4, 5

[40] Aliaksandr Siarohin, Oliver J Woodford, Jian Ren, Menglei Chai, and Sergey Tulyakov. Motion representations for articulated animation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13653–13662, 2021. 2

[41] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2Face: Real-time face capture and reenactment of RGB videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2387–2395, 2016. 2

[42] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. FaceVR: Real-time gaze-aware facial reenactment in virtual reality. *ACM Transactions on Graphics (ToG)*, 37:1–15, 2018. 2

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 3

[44] Duomin Wang, Yu Deng, Zixin Yin, Heung-Yeung Shum, and Baoyuan Wang. Progressive disentangled representation learning for fine-grained controllable talking head synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17979–17989, 2023. 2

[45] Jiadong Wang, Xinyuan Qian, Malu Zhang, Robby T Tan, and Haizhou Li. Seeing what you said: Talking face generation guided by a lip reading expert. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14653–14662, 2023.

[46] Jiayu Wang, Kang Zhao, Shiwei Zhang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. LipFormer: High-fidelity and generalizable talking face generation with a pre-learned facial codebook. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13844–13853, 2023.

[47] Suzhen Wang, Lincheng Li, Yu Ding, Changjie Fan, and Xin Yu. Audio2Head: Audio-driven one-shot talking-head generation with natural head motion. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2021. 2

[48] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. Few-shot video-to-video synthesis. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 1, 2

[49] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *IEEE/CVF Conference on Computer Vision and Pat-*

tern Recognition (CVPR), pages 10039–10049, 2021. 1, 2, 5, 7, 8, 3, 9, 10, 11

[50] Yaohui Wang, Di Yang, Francois Bremond, and Antitza Dantcheva. Latent image animator: Learning to animate images via latent space navigation. In *International Conference on Learning Representations (ICLR)*, 2022. 3

[51] Zhou Wang. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 6

[52] Shih-En Wei, Jason Saragih, Tomas Simon, Adam W Harley, Stephen Lombardi, Michal Perdoch, Alexander Hypes, Dawei Wang, Hernan Badino, and Yaser Sheikh. VR facial animation via multiview image translation. *ACM Transactions on Graphics (ToG)*, 38(4):1–16, 2019. 2

[53] Olivia Wiles, A Koepke, and Andrew Zisserman. X2Face: A network for controlling face generation using images, audio, and pose codes. In *European Conference on Computer Vision (ECCV)*, pages 670–686, 2018. 1, 2

[54] Wayne Wu, Yunxuan Zhang, Cheng Li, Chen Qian, and Chen Change Loy. ReenactGAN: Learning to reenact faces via boundary transfer. In *European Conference on Computer Vision (ECCV)*, pages 603–619, 2018. 8, 4, 11

[55] Chao Xu, Junwei Zhu, Jiangning Zhang, Yue Han, Wenqing Chu, Ying Tai, Chengjie Wang, Zhifeng Xie, and Yong Liu. High-fidelity generalized emotional talking face generation with multi-modal emotion space learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6609–6619, 2023. 2

[56] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9459–9468, 2019. 1, 2

[57] Jian Zhao and Hui Zhang. Thin-plate spline motion model for image animation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3657–3666, 2022. 1, 2, 5, 7, 8, 3, 9, 10, 11

[58] Ruiqi Zhao, Tianyi Wu, and Guodong Guo. Sparse to dense motion transfer for face image animation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1991–2000, 2021. 1, 2

[59] Hang Zhou, Yu Liu, Ziwei Liu, Ping Luo, and Xiaogang Wang. Talking face generation by adversarially disentangled audio-visual representation. In *AAAI Conference on Artificial Intelligence*, pages 9299–9306, 2019. 2

[60] Hang Zhou, Yasheng Sun, Wayne Wu, Chen Change Loy, Xiaogang Wang, and Ziwei Liu. Pose-controllable talking face generation by implicitly modularized audio-visual representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4176–4186, 2021. 2

[61] Tong Zhou, Changxing Ding, Shaowen Lin, Xinchao Wang, and Dacheng Tao. Learning oracle attention for high-fidelity face completion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7680–7689, 2020. 2

FSRT: Facial Scene Representation Transformer for Face Reenactment from Factorized Appearance, Head-pose, and Facial Expression Features

Supplementary Material

7. Implementation Details

We present important training and architecture details, including the parameter values that were used.

7.1. Architecture Details

Keypoint Detector. The keypoint detector is used as-is from [38] and not trained further. It consists of a 5-block Hourglass network [28] with a block expansion of 32 and a maximum feature map size of 1024. For keypoint extraction, the images are resized to 64×64. After decoding, the heatmaps are predicted by a final 7×7 convolution. Keypoint locations are given by the centroids of the corresponding heatmap.

Latent Expression Extractor. The latent expression extractor $\mathcal{X}$ has a single 7×7 convolutional layer that predicts $n_f = 32$ individual feature maps for each keypoint. For each keypoint, the individual feature maps computed by the keypoint detector are aggregated in x and y direction with the weights of the corresponding heatmap. After aggregating the features of each keypoint individually, the information is concatenated and fused to predict a global expression vector. The fusion is performed by a 4-layer MLP with $(640 - 1280 - 640)$ hidden units and $|e|$ output neurons.

Input and Query Representation. For both, the positional encoding in the input and query representation, we set the number of octaves to $\mathcal{O}_{pix} = 16$ and $\mathcal{O}_{key} = 4$ with start octaves $s_{\mathcal{O}_{pix}} = -1$ and $s_{\mathcal{O}_{key}} = -1$. Together with a latent expression dimension of $|e| = 256$, this results in a query representation of size $|Q_{I_D}| = 416$ and an input representation R_{S_i} with 419 input channels, since we also encode the RGB pixel color of the source image.

Patch CNN. In all experiments, we set the output feature dimension of the Patch CNN to $n_{\mathcal{E}}^{fm} = 768$. Since we are processing a very large number of input channels (419 when $|e| = 256$), we use a bottleneck of 96 feature maps in the first convolutional layer.

Encoder. The transformer encoder also has a feature dimension of 768. Each multi-head attention layer uses 12 heads with an attention dimension of 64. The encoder processes the patch embedding of each source image individually, so that the cardinality of the set-latent scene representation scales linearly with the number of source images. This allows a flexible number of source images to be used. In total, the encoder and Patch CNN (with $|e| = 256$) have 29,774,112 parameters.

Decoder. The decoder has a feature dimension equal to the size of the query representation $|Q_{I_D}|$. The input MLP (see decoder in Fig. 2) has two layers with 720 hidden units and $|Q_{I_D}|$ output neurons. In the attention blocks, we also use 12 heads with an attention dimension of 64. The MLP inside the attention block, which fuses the information from the individual heads, has two layers and $2|Q_{I_D}|$ hidden units. The final 5-layer render MLP has $(1536 - 1536 - 1536 - 768)$ hidden units and three output neurons for the RGB color.

For our small decoder ablation Ours/small$\mathcal{D}$, we reduce the number of heads from 12 to 6 and also halve the number of hidden units of the MLP inside the attention block. Finally, we replace the render MLP with a smaller 3-layer version with $(1536 - 768)$ hidden units. Compared to our standard decoder, the number of parameters is reduced from 15,310,131 to 6,012,723.

Discriminator. For the keypoint-aware discriminator $\mathcal{A}$, we use the implementation of Siarohin et al. [38] which is based on [15]. The input is an RGB image concatenated with ten heatmaps representing the driving keypoints. In total, we use four blocks, resulting in 512 output features with a downsampling factor of 16. For further implementation and loss details, we refer to Siarohin et al. [38].

7.2. Training Details

We train on three NVIDIA A100 (80GB) GPUs for about 23 days. We found that warming up (i.e. Phase I training, explained in Sec. 3.3) is essential to avoid ending up in local minima. Also, the batch size should be large enough. In our experiments we found out that 24 is sufficient. With a batch size of eight, training progressed slowly and appeared to be very unstable. Furthermore, we ended up in a local minimum with poor inference performance. When adding adversarial losses in training Phase III, we allow the discriminator to warm up for 500 iterations without computing gradients for the model. This is essential since otherwise the untrained discriminator will influence the current training progress with gradients of large magnitude.

Stopping Criterion. We extract a validation dataset, which we use to validate the self- and cross-reenactment performance. The self-reenactment performance is measured as in Tab. 1. For cross-reenactment, we randomly sample source images and driving videos. Model performance is judged visually by us. We found that it is not necessary to choose between good self- and cross-reenactment

performance, as both are typically correlated. We thus use self-reenactment scores as a way to find promising models and then verify cross-reenactment performance.

Visualizing Out-of-frame Motion. As explained in Sec. 3.1, we use a negative octave in the positional encoding of pixels and keypoints to uniquely encode values in $(-2, 2)$. However, the VoxCeleb dataset [27] (prepared as suggested by Siarohin et al. [38]) itself has no out-of-frame motion. Instead, we create out-of-frame motion by cropping the image with respect to the source image keypoints. We use external pre-estimated face keypoints [3] and select a random crop of all selected images (source and driving) such that all source keypoints are inside. Finally, the images are resized back to 256×256, which may change the aspect ratio and induces additional regularization. In some cases, the driving face will now be partially outside the image—generating corresponding training samples.

Since cropping will reduce the image resolution to less than 256^2, we download the dataset at the highest resolution possible so that the crop (before resizing) is ideally larger than 256^2 and no image detail is lost.

The keypoint detector can only predict keypoints within the image. Therefore, we detect keypoints of the uncropped images and use the cropping information to transform them into the cropped images.

Unlike the source keypoints, the latent expression vectors are extracted directly from the cropped source images. When extracting expression vectors from the driving frame, the differently augmented driving frame version (as explained in Sec. 3.2), ensures that the driving face is inside the image. In Fig. 6, we show that not addressing out-of-frame motion leads to poor results when keypoints are outside the image or close to the image boundaries.

7.3. User Study Details

We selected 30 different people to participate in the user study (see Tab. 2). Since we compared the methods in pairs, each participant was only allowed to judge one related method. Furthermore, each participant judged both relative motion transfer and absolute motion transfer. The face reenactment task was initially explained, and participants were instructed to base their decision on the following two criteria:

1. Does the motion transfer work well (including ID preservation)?
2. Does the animation look like a natural and consistent video?

Each participant was simultaneously shown the source image, the driving video, our result and the animation of the comparison method. In each of the 20 sequences, we randomized whether our method was shown on the left or on the right. Participants could only decide once the video had run through. However, the video automatically restarted, so that there was no overall time limit. A decision was made by clicking on the preferred video.

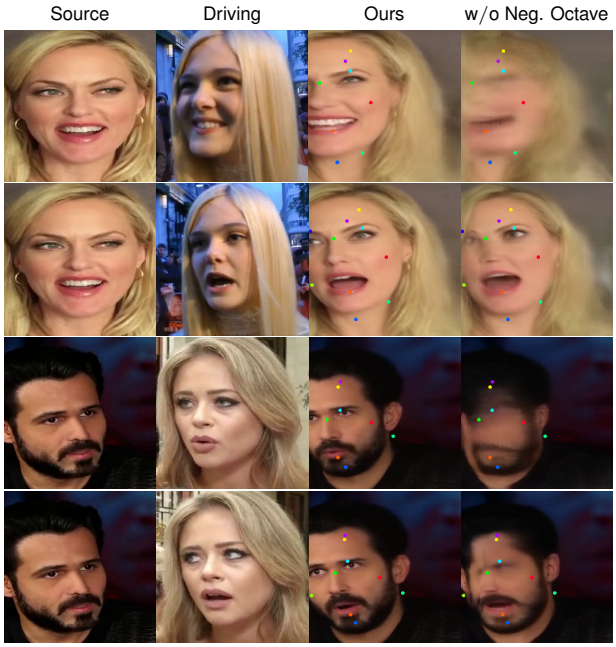

Figure 6. Out-of-frame motion with (Ours) and without explicit addressing keypoints outside the image (w/o Neg. Octave). Out-of-frame motion only occurs when relative motion transfer is used (see Sec. 3.4). The predicted images are visualized with the driving keypoints that were used in the decoder. Images from the VoxCeleb test set [27].

Method	SSIM↑	PSNR↑	L1↓	AKD↓
Ours	.7576	23.67	.0421	2.13
Ours/ $1 \rightarrow 2$-Src	.7181	23.06	.0453	2.42
Ours/ 2-Src	.7891	25.00	.0360	2.04
Ours/ $2 \rightarrow 3$-Src	.8092	25.80	.0325	2.00
Ours/ $2 \rightarrow 1$-Src	.7610	23.85	.0418	2.13

Ours/$t \rightarrow i$-Src means that the model trained with t source images is evaluated with i source images during inference.

Table 3. Self-reenactment results on the official VoxCeleb test set [27] when generalizing to a different number of source images without explicit training. Training with two source images increases self-reenactment performance, even when only one source image is used for inference.

8. Additional Experiments & Results

We report auxiliary experiments and more qualitative results here.

8.1. Flexibility in the Number of Source Images

We investigate the generalization behavior with respect to changing the number of source images during inference.

Here, our reference model was trained with a single source image and with two source images. As reported in Tab. 3, the model trained with two source images generalizes in both directions, with fewer and with more source images used for inference. Interestingly, when reducing the number of source images to one (line Ours$/2 \rightarrow$ 1-Src in Tab. 3) it even produces slightly better self-reenactment results than our model explicitly trained with only one source image (line Ours in Tab. 3). With three source images available for inference (line Ours$/2 \rightarrow$ 3-Src in Tab. 3), the performance increases further, indicating that additional source images can be added at inference as available.

The model trained with only one source image shows a significant drop in performance when the number of source images is increased during evaluation (line Ours$/1 \rightarrow$ 2-Src in Tab. 3). Therefore, if a flexible number of source images is desired, we recommend training with at least two source images. Alternatively, the number of source images can be chosen flexibly during training. To ensure that the data can still be batched, we recommend always selecting the maximum number of source images, but masking the set-latents of unnecessary source images in the attention module of the decoder.

8.2. Ablation Study

In Figs. 9 and 10 we present qualitative results of our ablations (see Sec. 4.2) in the cross- and self-reenactment situation, respectively. In terms of motion transfer accuracy, our reference model with $|e|$=256 produces slightly better results than models using $|e|$=64 or $|e|$=128.

By using two source images, information from both source images can be extracted and fused to produce more accurate animations. Especially if the second source image reveals occluded background or different head regions, less information has to be guessed by the model. As shown in Figs. 9 and 10, using multiple source images (Ours$/$2-Src) can help to produce animations with more detail in face, hair, and background.

Our ablation with a small decoder (Ours$/$small$\mathcal{D}$) has a motion transfer capability similar to our reference model (Ours), but with a slightly reduced sharpness in the animations.

8.3. Comparison with State-of-the-Art Methods

In Fig. 11 and Fig. 12 we present additional cross-reenactment results on the VoxCeleb test set [27] with relative and absolute motion transfer compared to all state-of-the-art methods from our user study (see Tab. 2). While TSMM [57], DaGAN [11], OSFS [49], and FOMM [38] are also keypoint based, DPE [29] uses a latent head pose description. This, however, eliminates the ability to perform relative motion transfer. As the visualizations show, our method produces significantly more natural results with higher ID preservation and more accurate and plausible motion transfers. Especially when there is a large pose offset, related methods often fail to produce satisfactory results. For animated results, see our project page.[2]

Figure 7. Out-of-distribution results with relative motion transfer generated by our method. The source images are extracted from popular paintings and the driving frames are from the VoxCeleb2 test set [5].

8.4. Out-of-Distribution Animation

As shown in Fig. 7, our model trained on VoxCeleb [27] generalizes to out-of-distribution source images extracted from popular paintings.

[2] https://andrerochow.github.io/fsrt

8.5. Generalizing to other Datasets

We report generalization examples of our models trained on VoxCeleb to other datasets at inference time. Specifically, we show the following source → driving combinations:

- CelebA-HQ [18] → VoxCeleb2 [5] in Fig. 13,
- VoxCeleb2 [5] → VoxCeleb2 [5] in Fig. 14, and
- CelebV [54] → CelebV [54] in Fig. 15.

We note that VoxCeleb2 covers a significantly larger number of identities in the test set compared to VoxCeleb. As the results show, our model generalizes to all of these combinations, while still producing more accurate animations compared to related methods.

8.6. Omitting Keypoints

We present qualitative results of our model ablation Ours/$n_{\mathcal{K}} = 0$ without keypoints in Fig. 15. Compared to our reference model (Ours), we found that the accuracy of the motion transfer is slightly reduced. In particular, the animated gaze direction seems to be less accurate (see third row in Fig. 15). Omitting the keypoints makes it impossible to perform relative motion transfer, since all pose information is implicitly encoded in the expression vector e.

In this variant, images input to the expression network are not augmented through cropping, since this makes recovery of the head pose impossible without keypoints. However, we discovered that performing a random center crop with variable aspect ratio on the driving frame (while requiring the network to reconstruct the full driving frame) reduces shape deformations, since the network becomes invariant against aspect ratio changes and scale (see Ours/$n_{\mathcal{K}} = 0$ + Crop Aug. in Fig. 8). While this might be useful in cross-reenactment applications where relative motion transfer is not required, it reduces self-reenactment scores (see Tab. 4)—where this invariance is not helpful but actually harmful. A particular reason for this might be that this variant cannot transfer zooming or dolly shots due to scale invariance.

8.7. Statistical Regularization

In Fig. 16, we visualize the effect of training without our proposed statistical regularization. As the results show, training without $\mathcal{L}_{\text{Cov}}$ and $\mathcal{L}_{\text{Var}}$ leads to significant artifacts around the animated face region, indicating that ID information leaks from the driving frame through the expression vector e_D. Our proposed factorization is therefore not achieved.

Method	#KP	SSIM↑	PSNR↑	L1↓	AKD↓
Ours	10	.7576	23.67	.0421	2.13
$n_{\mathcal{K}} = 0$	0	.7445	23.56	.0436	2.64
+Crop Aug.	0	.7240	22.98	.0469	2.99

Table 4. Self-reenactment results on the official VoxCeleb test set [27]. We compare our model ablation without keypoints (Ours/$n_{\mathcal{K}} = 0$) with an ablation that is additionally trained with random center cropping (Ours/$n_{\mathcal{K}} = 0$ + Crop Aug.). The scores of our reference model (Ours) are shown in the first row.

Figure 8. Ablations without keypoints. This comparison is using absolute motion transfer. When combining a keypoint-less model with random center cropping during training (right column), shape deformations and scale changes are prevented. The images are from the VoxCeleb test set [27], the VoxCeleb2 test set [5], and the CelebA-HQ dataset [18] (as indicated by the source → driving notation).

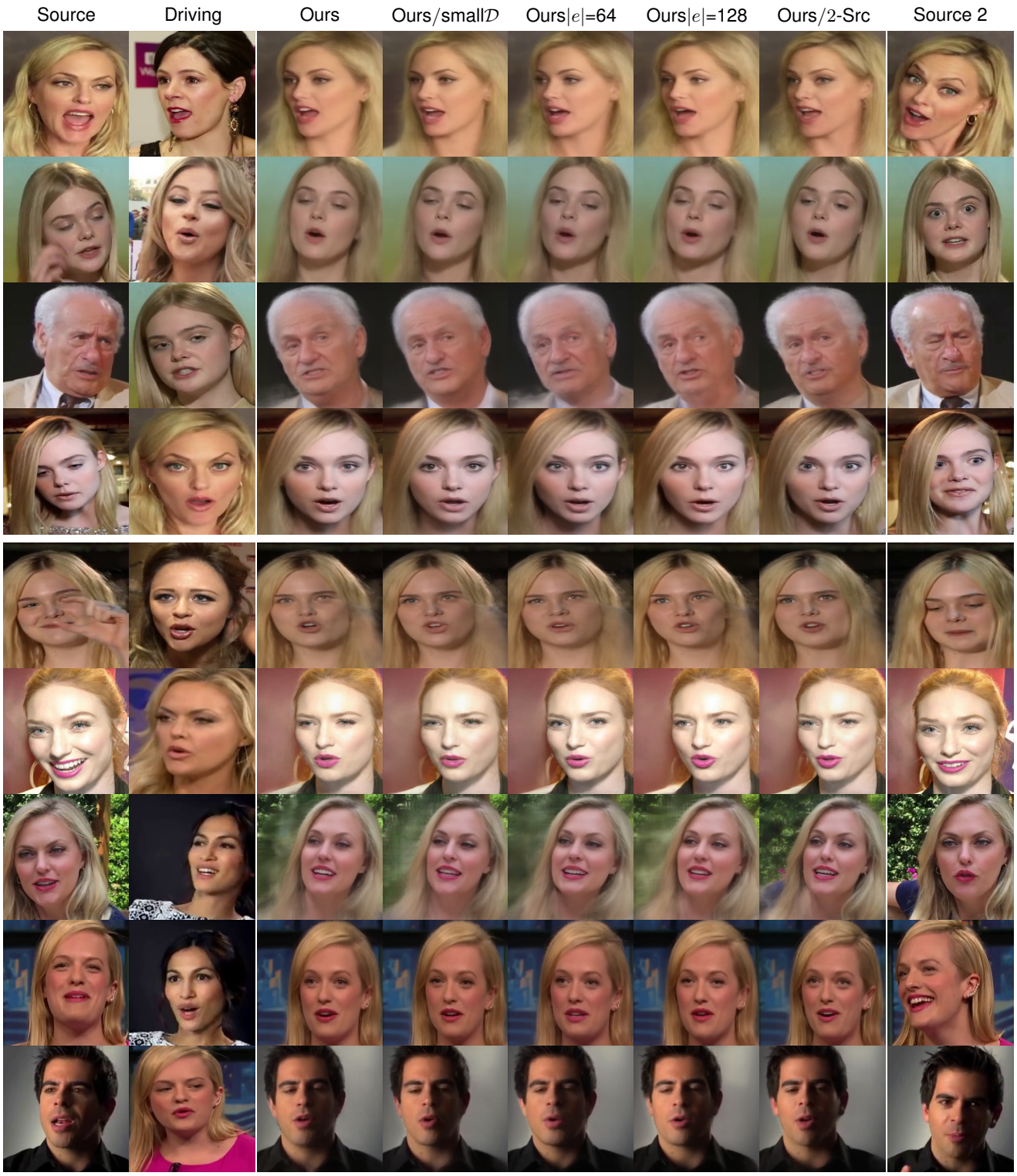

Figure 9. Ablation study in cross-reenactment on the VoxCeleb test set [27] with absolute motion transfer (upper block) and relative motion transfer (lower block). Our ablation Ours/2-Src consistently fuses the information of both source images. It produces more detail in the face, hair, and background, especially when the second source image reveals information missing in the first source image.

Figure 10. Ablation study in self-reenactment on the VoxCeleb test set [27]. The accuracy of motion transfer (especially mouth and eye motion) decreases slightly when reducing the size of the latent expression vector e. In the first and fourth animation, Ours$|e|$=64 produces inaccurate mouth expressions. Ours/2-Src generates more detail by integrating the information from both source images.

Figure 11. Comparison with SOTA on the VoxCeleb test set [27] in cross-reenactment (relative motion transfer). Our model generates more accurate expressions, is less sensitive to the alignment assumption (Sec. 3.4), and learns to realistically fill missing face parts (third row). Others often produce mismatched expressions and fail for large pose offsets. The last row shows a source image from CelebA-HQ [18].

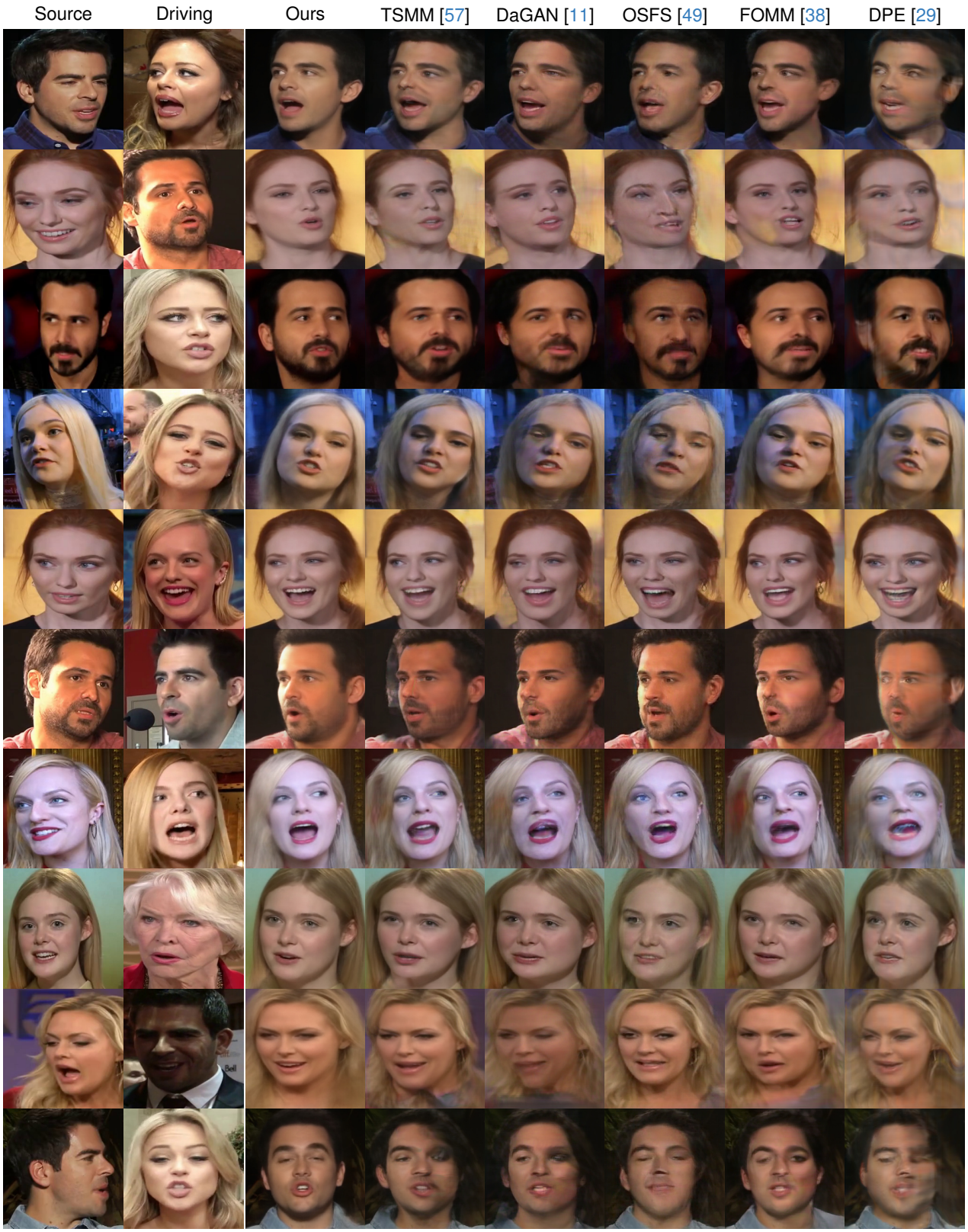

Figure 12. Comparison with SOTA on the VoxCeleb test set [27] in cross-reenactment with absolute motion transfer. We generate more accurate facial expressions with better ID preservation. Related methods often produce strong shape deformations, artifacts and blurry results (especially in the mouth region). The sixth animation shows that our method even animates the sunlight on the side of the face.

Figure 13. Cross-reenactment generalization to driving videos from the VoxCeleb2 test set [5] and source images from the CelebA-HQ dataset [18] with relative motion transfer.

Figure 14. Cross-reenactment generalization to driving videos and source images both from the VoxCeleb2 test set [5] with relative motion transfer.

Figure 15. Comparison of our model with and without keypoints and state-of-the-art methods in cross-reenactment with absolute motion transfer. The top block shows generalization to source and driving frames extracted from the CelebV dataset [54]. The bottom block shows generalization to driving frames extracted from the VoxCeleb2 test set [5] and source images from the CelebA-HQ dataset [18].

Figure 16. Benefit of statistical regularization (relative motion transfer). Training without $\mathcal{L}_{\text{Cov}}$ and $\mathcal{L}_{\text{Var}}$ leads to visible artifacts around the animated face (see red arrows), indicating that the identity of the driving person is leaking into the expression vector e_D. The images are from the VoxCeleb test set [27] (indicated with *) and the VoxCeleb2 test set [5] (remaining).