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.

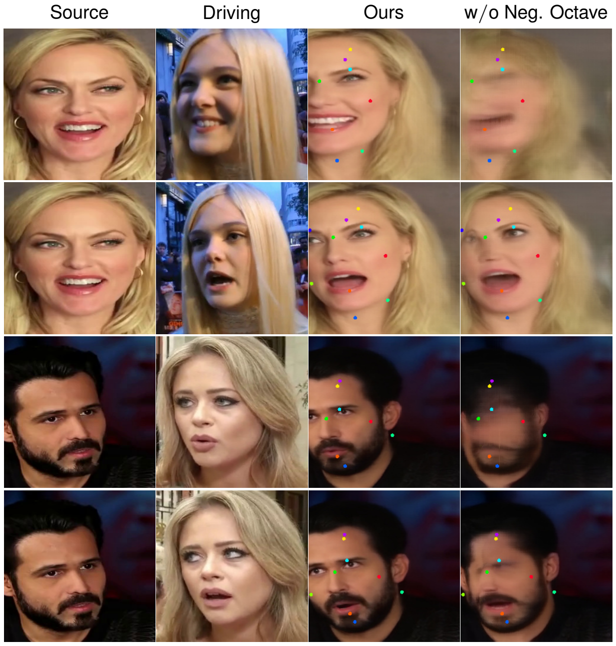

Figure 6. Out-of-frame motion with (Ours) and without explicit addressing keypoints outside the image (w/o Neg. Octave). Out-of-frame motion only occurs when relative motion transfer is used (see Sec. 3.4). The predicted images are visualized with the driving keypoints that were used in the decoder. 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$/2 \to$ 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$/2 \to$ 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$/1 \to$ 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]

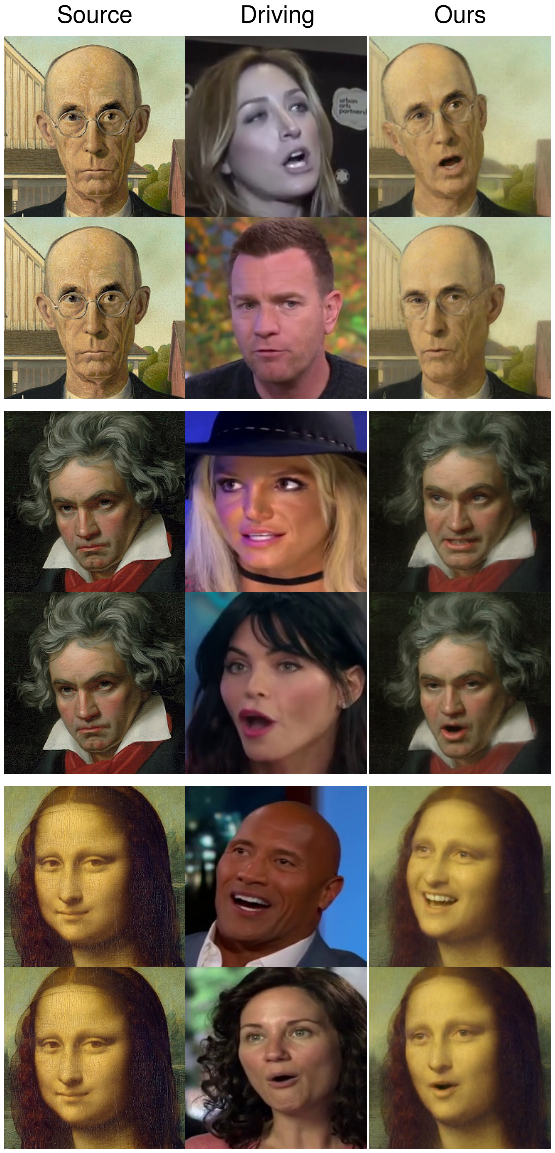

Figure 7. Out-of-distribution results with relative motion transfer generated by our method. The source images are extracted from popular paintings and the driving frames are from the VoxCeleb2 test set [5].

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.

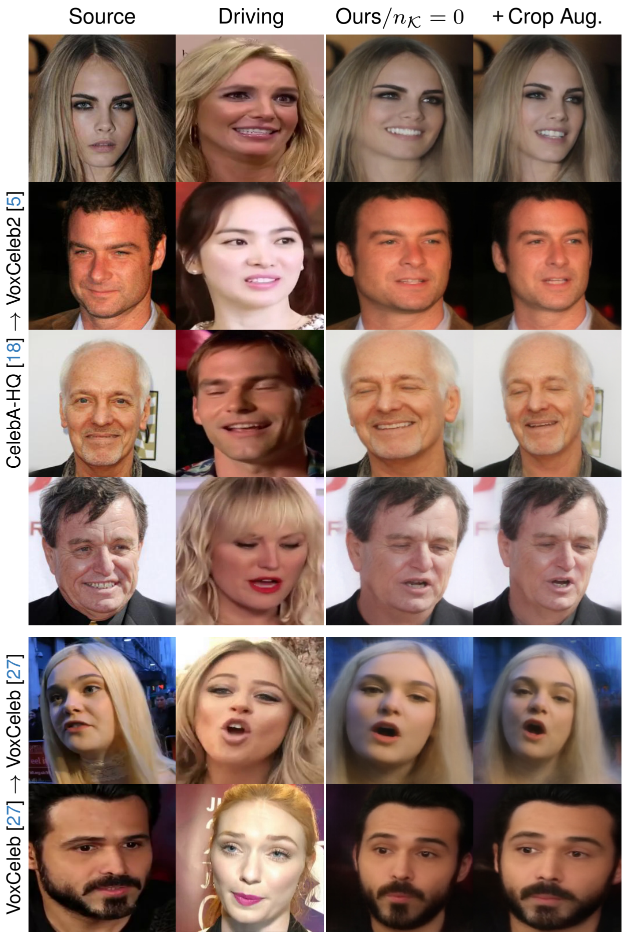
Source
Driving
Ours/$n_{\mathcal{K}} = 0$
+ Crop Aug.
CelebA-HQ [18] → VoxCeleb2 [5]
VoxCeleb [27] → VoxCeleb [27]

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).

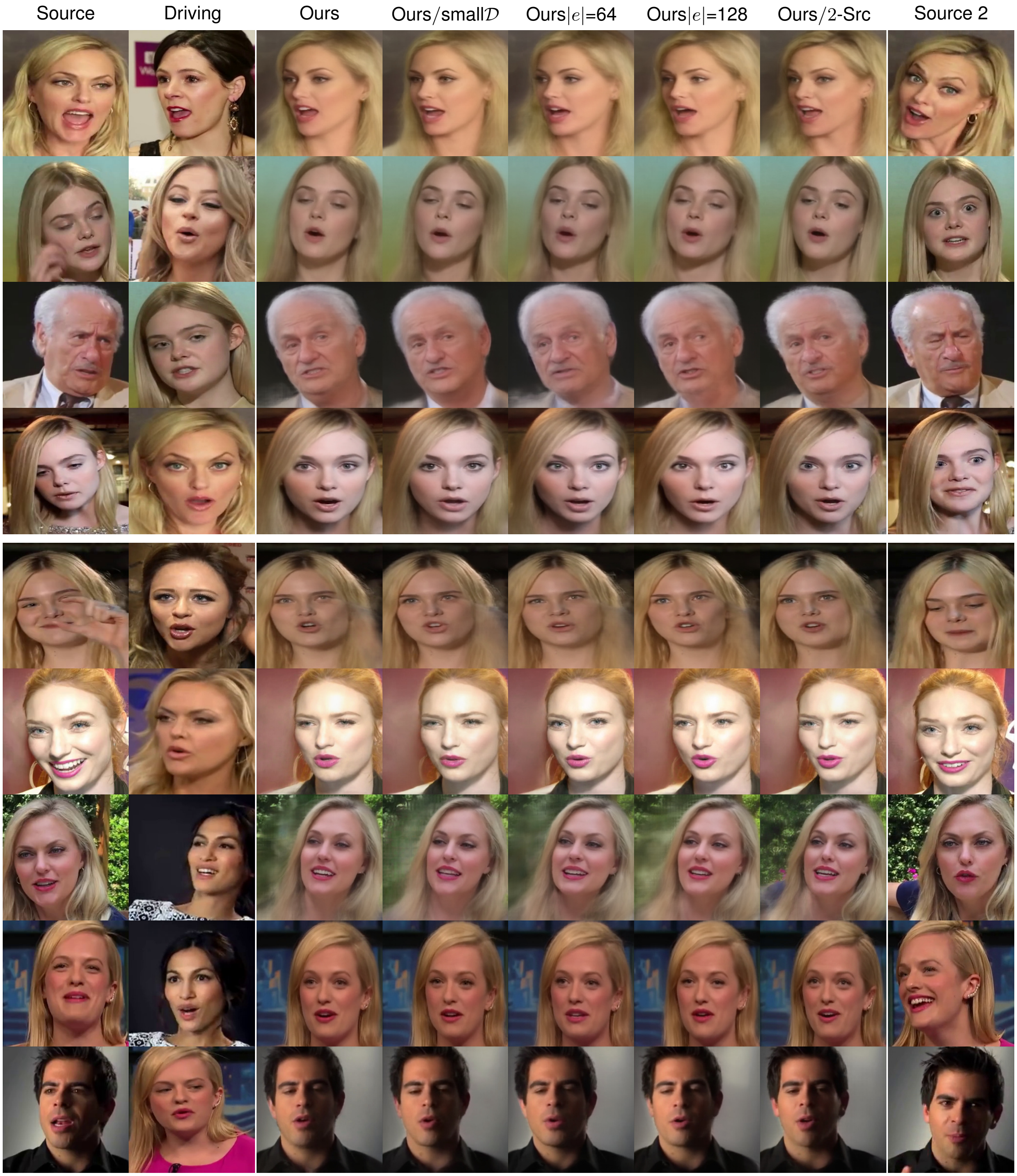

Figure 9. Ablation study in cross-reenactment on the VoxCeleb test set [27] with absolute motion transfer (upper block) and relative motion transfer (lower block). Our ablation Ours/2-Src consistently fuses the information of both source images. It produces more detail in the face, hair, and background,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

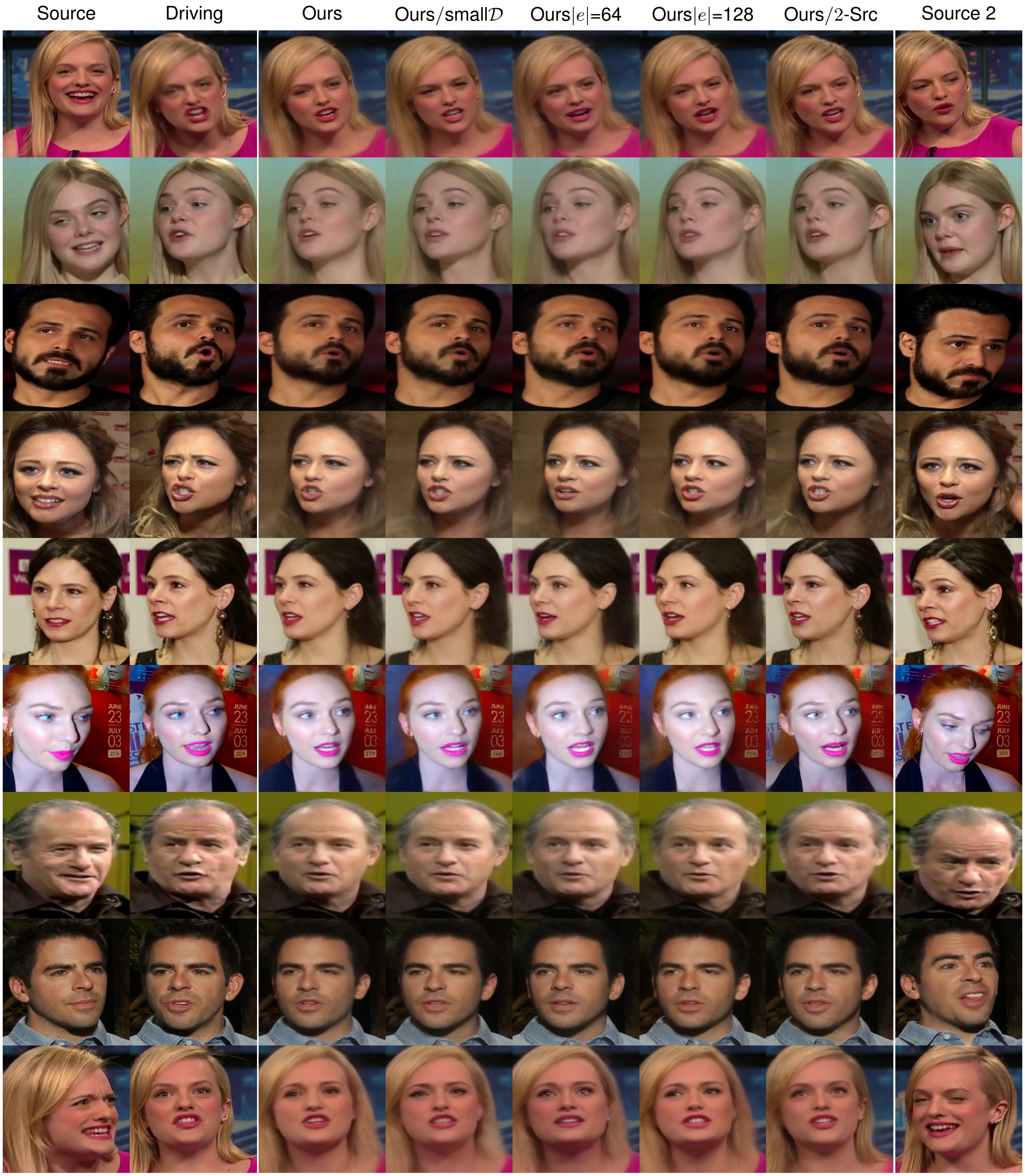

Figure 10. Ablation study in self-reenactment on the VoxCeleb test set [27]. The accuracy of motion transfer (especially mouth and eye motion) decreases slightly when reducing the size of the latent expression vector e. In the first and fourth animation, Ours$|e|$=64 produces inaccurate mouth expressions.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

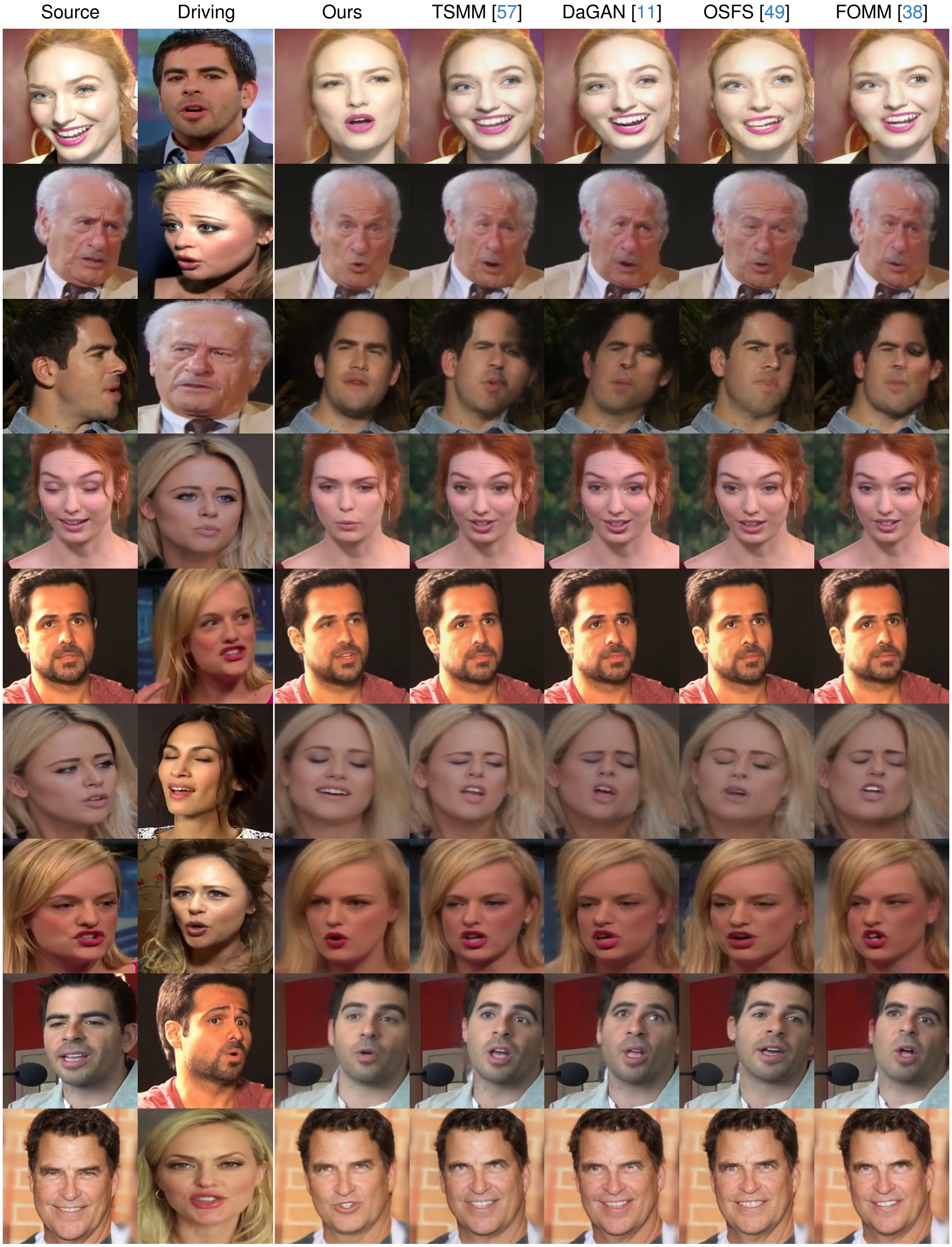

Figure 11. Comparison with SOTA on the VoxCeleb test set [27] in cross-reenactment (relative motion transfer). Our model generates more accurate expressions, is less sensitive to the alignment assumption (Sec. 3.4), and learns to realistically fill missing face parts (third row). Others often produce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

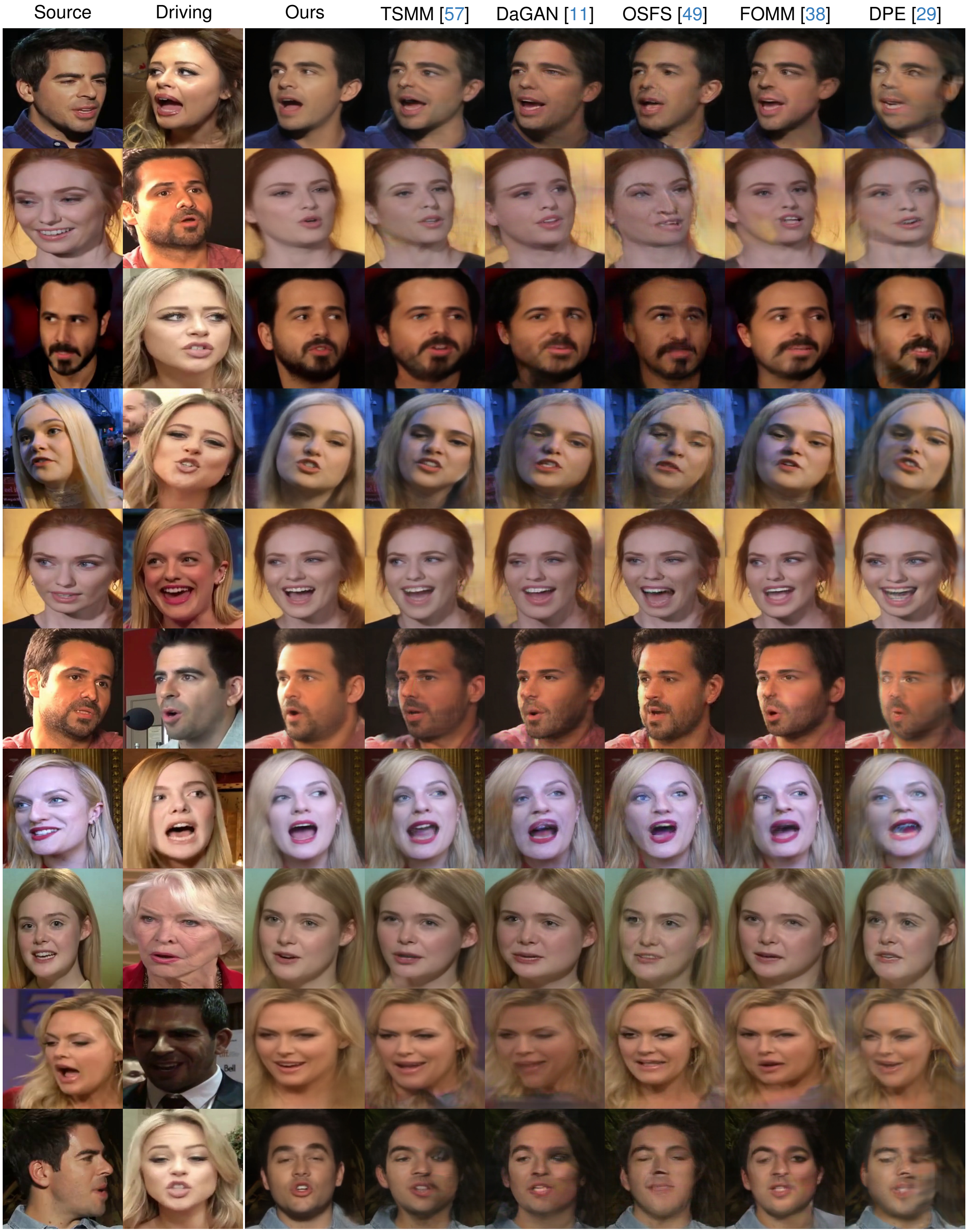

Figure 12. Comparison with SOTA on the VoxCeleb test set [27] in cross-reenactment with absolute motion transfer. We generate more accurate facial expressions with better ID preservation. Related methods often produce strong shape deformations, artifacts and blurry results (especially in the mouth region).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

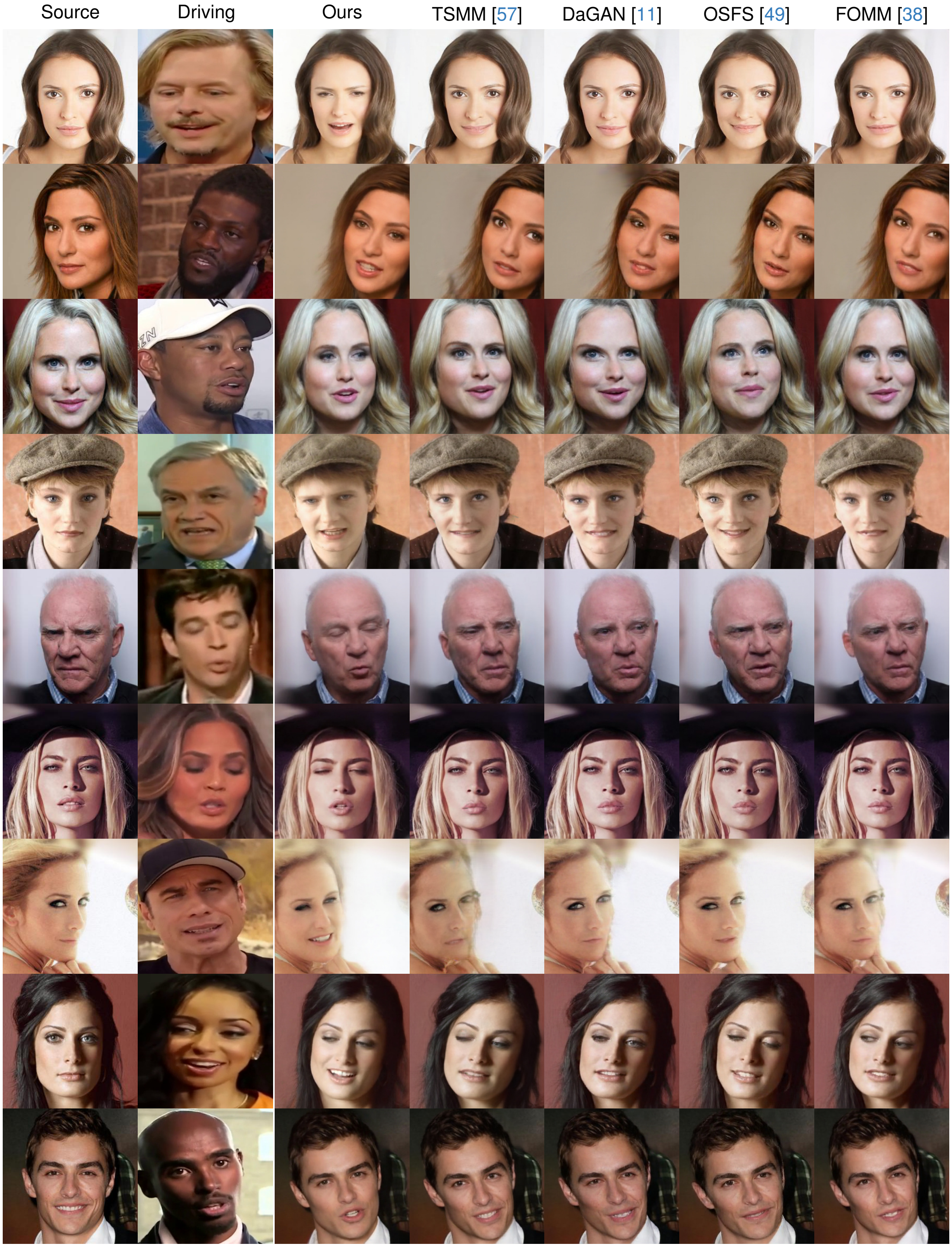

Figure 13. Cross-reenactment generalization to driving videos from the VoxCeleb2 test set [5] and source images from the CelebA-HQ dataset [18] with relative motion transfer.

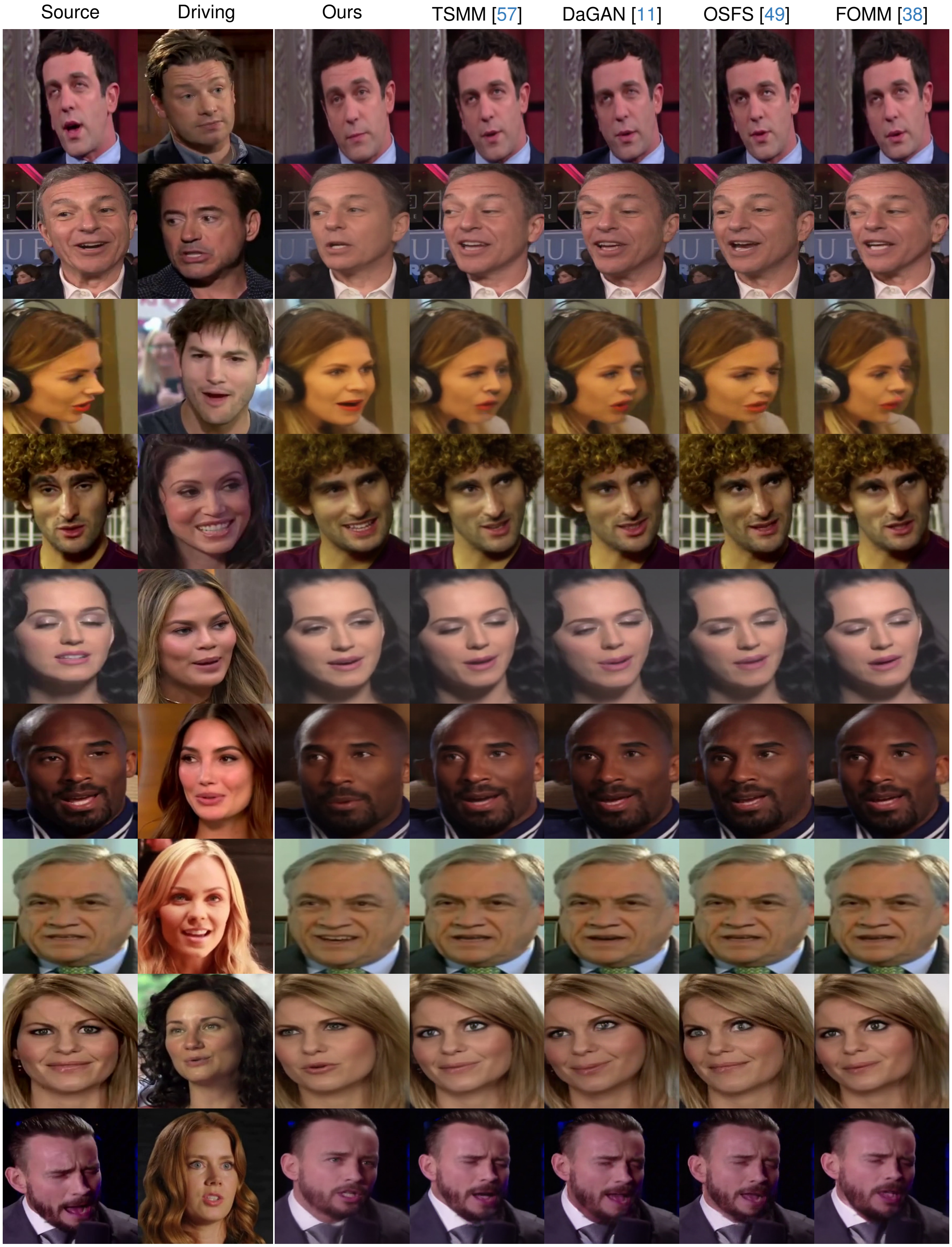

Figure 14. Cross-reenactment generalization to driving videos and source images both from the VoxCeleb2 test set [5] with relative motion transfer.

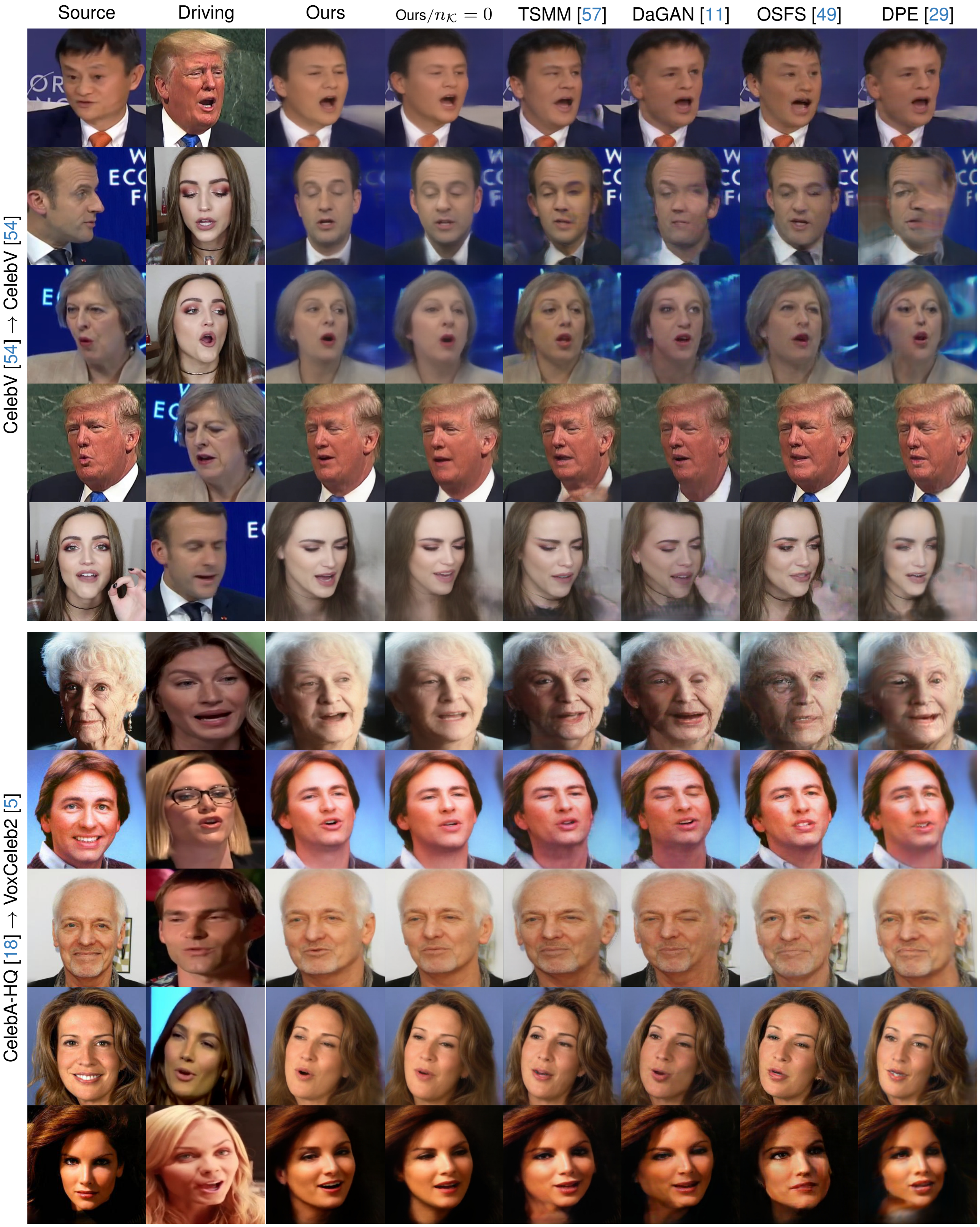
Source
Driving
Ours
Ours/$n_{\mathcal{K}} = 0$
TSMM [57]
DaGAN [11]
OSFS [49]
DPE [29]
CelebV [54] → CelebV [54]
CelebA-HQ [18] → VoxCeleb2 [5]

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].

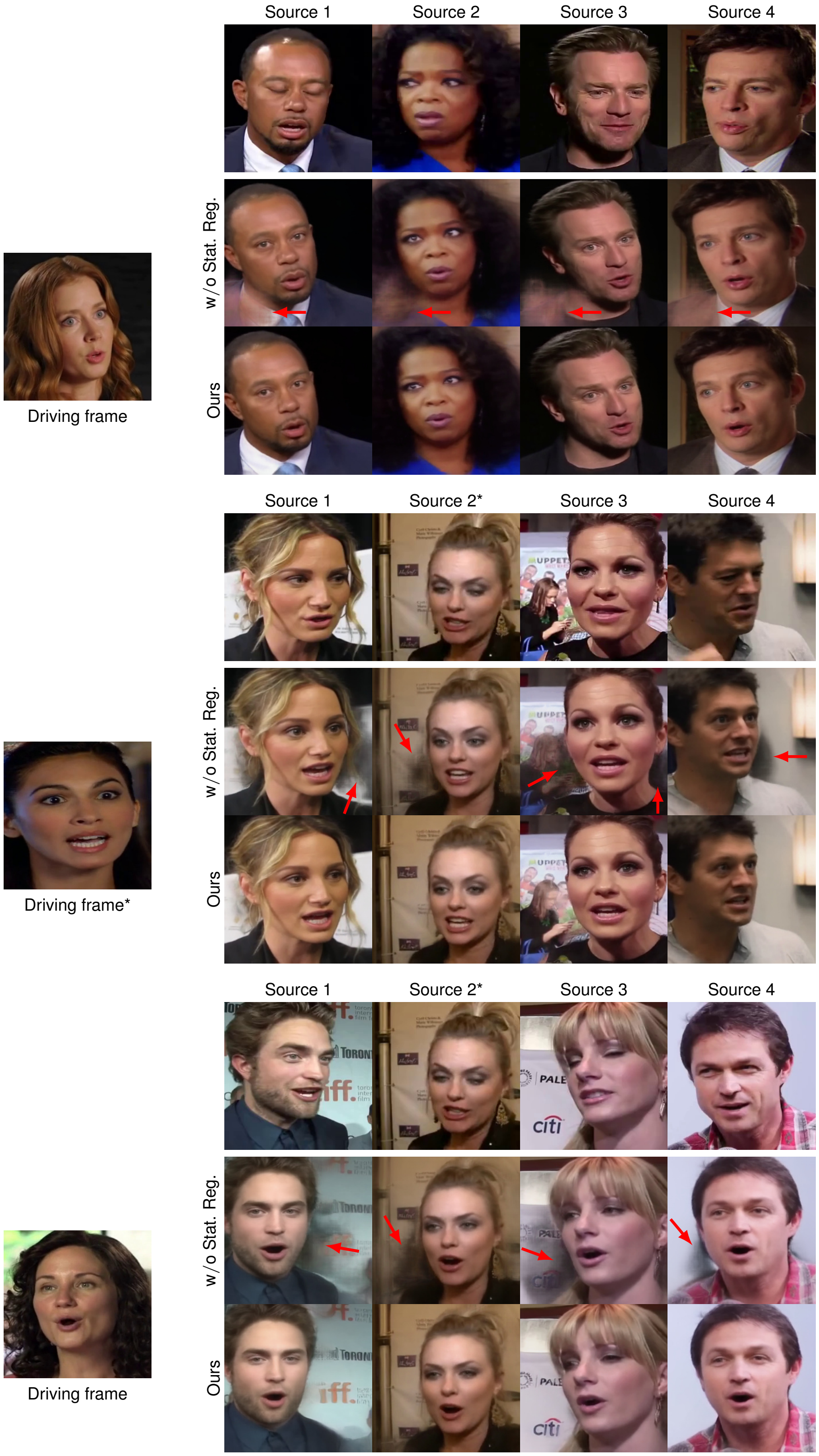

Figure 16. Benefit of statistical regularization (relative motion transfer). Training without $\mathcal{L}_{\text{Cov}}$ and $\mathcal{L}_{\text{Var}}$ leads to visible artifacts around the animated face (see red arrows), indicating that the identity of the driving person is leaking into the expression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