Animating Arbitrary Objects via Deep Motion Transfer Supplementary Material

Aliaksandr Siarohin[1], Stéphane Lathuilière[1], Sergey Tulyakov[2], Elisa Ricci[1,3] and Nicu Sebe[1,4]
[1]DISI, University of Trento, Italy, [2] Snap Inc., [3] Fondazione Bruno Kessler (FBK), Trento, Italy,
[4]Huawei Technologies Ireland, Dublin, Ireland
{aliaksandr.siarohin,stephane.lathuilire,e.ricci,niculae.sebe}@unitn.it, stulyakov@snap.com

In this supplementary material, we provide implementation details (Sec. A), introduce a new dataset (Sec. B) and report additional experimental results (Sec. C). Additionally we provide a video file with further qualitative examples.

A. Implementation details

As described in Sec. 3, each module employs a U-Net architecture. We use the exact same architecture for all the networks. More specifically each block of each of the encoder consists of a 3×3 convolution, batch normalization [4], ReLU and average pooling. The first convolution layers have 32 filters and each subsequent convolution doubles the number of filters. Each encoder is composed of a total of 5 blocks. The decoder blocks have similar structure: 3×3 convolution, batch normalization and ReLU followed by nearest neighbour up-sampling. The first block of the decoder has 512 filters. Each consequent block has the reduced number of filters by a factor of 2.

As described in Sec. 3.2, the keypoint detector Δ produces K heatmaps followed by softmax. In particular, we employ softmax activations with 0.1 temperature. Indeed, thanks to the use of a low temperature for softmax, we obtain sharper heatmaps and avoid uniform heatmaps that would lead to keypoints constantly located in the image center.

For G, we employ 4 additional Residual Blocks [3] in order to remove possible warping artifacts produces by M. The output of G is a 3 channel feature map followed by the sigmoid. We use the discriminator architecture described in [7].

The framework is trained for T epochs where T equals 250, 500 and 10 for *Tai-Chi*, *Nemo* and *Bair* respectively. Epoch involves training the network on 2 randomly sampled frames from each training video. We use the Adam optimizer [5] with learning rate 2e-4 and then with learning rate 2e-5 for another $\frac{T}{2}$ epochs.

As explained in Sec. 4.2, for *Image-to-Video* translation, we employ a single-layer GRU network in order to predict the keypoint sequence used to generate the video. This recurrent network [2] has 1024 hidden units and is trained via $\mathcal{L}_1$ minimization.

B. MGif dataset

We collected an additional dataset of videos containing movements of different cartoon animals. Each video is a moving *gif* file. Therefore, we called this new dataset *MGif*. The dataset consists of 1000 videos, we used 900 videos for training and 100 for evaluation. Each video has size 128×128 and contains from 5 to 100 frames. The dataset is particularly challenging because of the high appearance variation and motion diversity. Note that in the experiments on this dataset, we use absolute keypoint locations from the driving video instead of the relative keypoint motion detailed in Sec. 3.6.

C. Additional experimental results

In this section, we report additional results. In Sec. C.1 we visually motivate our alignment assumption, in Sec. C.2 we complete the ablation study and, in Secs. C.3 and C.4, we report qualitative results for both the image-to-video and image animation problems. Finally, in Sec. C.5, we visualize the keypoint predicted by our self-supervised approach.

C.1. Explanation of alignment assumption

Our approach assumes that the object in the first frame of the driving video and the object in the source image should be in similar poses. This assumption was made to avoid situations of meaningless motion transfer as shown in Fig. 7. In the first row, the driving video shows the action of closing the mouth. Since the mouth of the subject in the source

image is already closed, mouth disappears in the generated video. Similarly, in the second row, the motion in the driving video shows a mouth opening sequence. Since the mouth is already open in the source image, motion transfer leads to unnaturally large teeth. In the third row the man is asked to raise a hand, while it has already been raised.

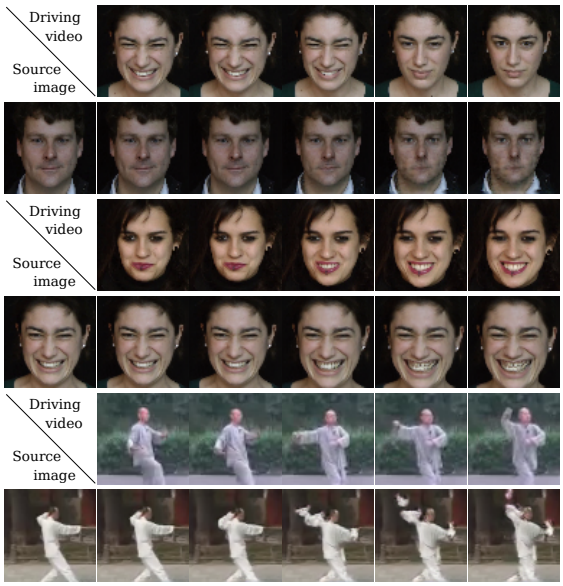

Figure 7: Illustration of the pose misalignment issue on the *Nemo* and *Tai-Chi* datasets.

C.2. Additional ablation study

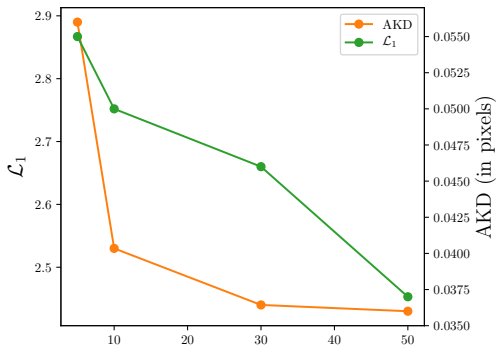

Figure 8: Reconstruction Errors as functions of the number of keypoints. Results obtained on *Tai-Chi*.

We perform experiments to measure the impact of the number of keypoints on video reconstruction quality. We report results on *Tai-Chi* dataset in Fig. 8. We computed $\mathcal{L}_1$ and *AKD* metrics as described in the paper. As expected, increasing the number of keypoints leads to a lower reconstruction error, but additional keypoints introduce memory and computational overhead. We use 10 keypoints in all our experiments, since we consider this to be a good trade-off.

C.3. Image-to-Video translation

As explained in Sec. 4.2 of the main paper, we compare with the three state of the art methods for Image-to-Video translation: MoCoGAN [6] and SV2P [1], and CMM-Net [8]. CMM-Net is evaluated only on *Nemo* and SV2P only on the *Bair* dataset. We report a user study and qualitative results.

User Study. We perform a user study for the image-to-video translation problem. As explained in Sec. 4.3, we perform pairwise comparisons between our method and the competing methods. We employ the following protocol: we randomly select 50 videos and use the first frame of each of video as the reference frames to generate new videos. For each of the 50 videos the initial frame, and two videos generated by our and one of the competing methods are shown to the user. We provide the following instructions: "Select a more realistic animation of the reference image". As in Sec. 4.2 of the main paper, our method is compared with MoCoGAN [6], Sv2p [1], and CMM-Net [8]. The results of the user study are presented in Table 5. On average, users preferred the videos generated by our approach over those generated by other methods. The preference gap is especially evident for the *Tai-Chi* and *Bair* datasets that contain a higher amount of large motion. This supports the ability of our approach to handle driving videos with large motion.

	Tai-Chi	*Nemo*	*Bair*
MoCoGAN [6]	88.2%	68.2%	90.6%
CMM-Net [8]	-	63.6%	-
SV2P [1]	-	-	98.8%

Table 5: User study results on image-to-video translation. Proportion of times our approach is preferred over the competitors methods

Qualitative results. We report additional qualitative results in Figs. 9, 10 and 11. These qualitative results further support the ability of our method to generate realistic videos from source images and driving sequences.

In particular, for the *Nemo* dataset (Fig. 9), MoCoGAN and CMM-Net suffer from more artifacts. In addition, the videos generated by MoCoGAN do not preserve the identity of the person. This issue is particularly visible when comparing the first and the last frames of the generated video. CMM-Net preserves better the identity but fails in generating realistic eyes and teeth. In contrast to these works, our method generates realistic smiles while preserving the person identity.

For *Tai-Chi* (Fig. 11), MoCoGAN [6] produces videos where some parts of the human body are not clearly visible (see rows 3,4 and 6). This is again due to the fact that visual information is embedded in a vector. Conversely, our method generates realistic human body with richer details.

For *Bair* (Fig. 10), [1] completely fails to produce videos where the robotic is sharp. The generated videos are blurred. MoCoGAN [6] generates videos with more details but containing many artifacts. In addition, the backgrounds generated by MoCoGAN are not temporally coherent. Our method generates realistic robotic arm moving in front of detailed and temporally coherent backgrounds.

C.4. Image animation

As explained in the main paper, we compare our method with X2Face [9]. Results are reported in Figs. 13, 14 and 15 on the *Nemo*, *Tai-Chi* and *Bair* datasets respectively.

When tested using the *Nemo* dataset (Fig. 13), our method generates more realistic smiles on most of the randomly selected samples despite the fact that the XFace model is specifically designed for faces. Similarly to the main paper, the benefit of transferring the relative motion over absolute locations can be clearly observed in the bottom right example where the video generated by X2face inherits the large cheeks of the young boy in the driving video.

For *Tai-Chi* (Fig. 14), X2face is not able to handle the motion of the driving video and simply warps the human body in the source image as a single blob.

For *Bair* (Fig. 15), we observe a similar behavior. X2face generates unrealistic videos where the robotic arm is generally not distinguishable. On the contrary, our model is able to generate a realistic robotic arm moving according to the driving video motion.

Finally in Fig 12, we report results on the *MGif* dataset. First, these examples illustrate high diversity of *MGif* dataset. Second, we observe that our model is able to transfer the motion of the driving video even if the appearance of the source frame is very different from the driving video. In particular, in all the generated sequences, we observe that the legs are correctly generated and follow the motion of the driving video. The model preserves the rigid parts of the animals as, for instance, the abdomen. In the last row, we see that the model is also able to animate the fox tail according to the motion of the cheetah tail.

C.5. Keypoint visualization

Finally, we report visual examples of keypoints learned by our model in Figs. 16, 17, 18 and 19. On the *Nemo* dataset, we observe that the obtained keypoints are semantically consistent. For instance, the cyan and light green keypoints constantly correspond to the nose and the chin respectively. For *Tai-Chi*, the keypoints are also semantically consistent: light green for the chin and yellow for the left-side arm (right arm in frontal views and left arm in back views), for instance. For the *Bair* dataset, we observe that two keypoints (light green and dark blue) correspond to the robotic arm. The other keypoints are static and can correspond to the background. Finally, concerning the *MGif* dataset, we observe that each keypoint corresponds to two different animal parts depending if the animal is going towards left or right. In the case of animals going right (last three rows), the keypoints are semantically consistent (red for the tail, dark blue for the head etc.). Similarly, the keypoints are semantically consistent among images of animal going left (red for the head, dark blue for the tail etc.). Importantly, we observe that a keypoint is associated to each highly moving part, as legs and tails.

References

[1] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. In *ICLR*, 2017.

[2] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS*, 2014.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014.

[6] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *CVPR*, 2018.

[7] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2017.

[8] Wei Wang, Xavier Alameda-Pineda, Dan Xu, Pascal Fua, Elisa Ricci, and Nicu Sebe. Every smile is unique: Landmark-guided diverse smile generation. In *CVPR*, 2018.

[9] Olivia Wiles, A Sophia Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *ECCV*, 2018.

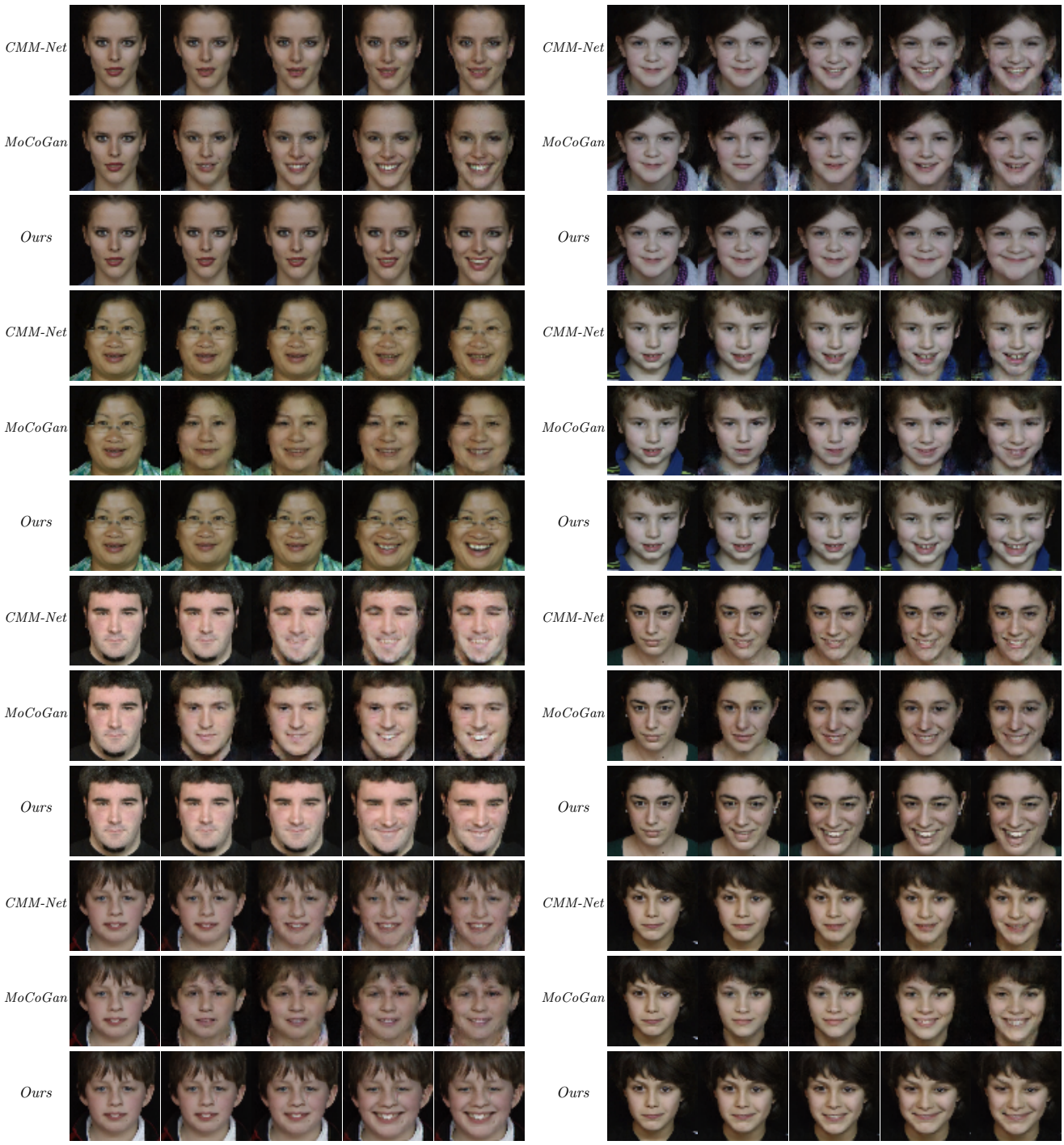

Figure 9: Qualitative results for *Image-to-video* translation on the *Nemo* dataset.

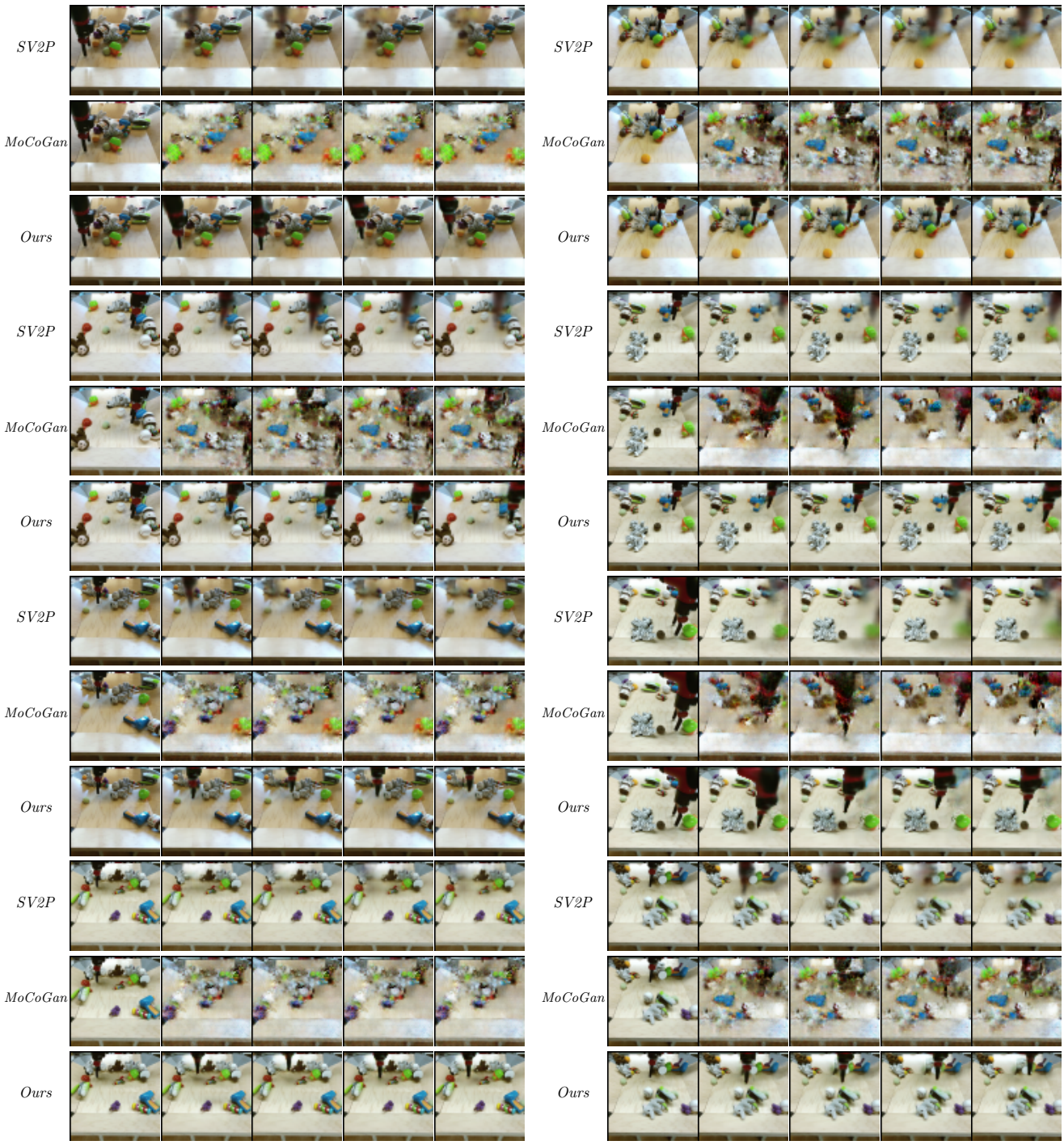

Figure 10: Qualitative results for *Image-to-video* translation on the *Bair* dataset.

Figure 11: Qualitative results for *Image-to-video* translation on the *Tai-Chi* dataset.

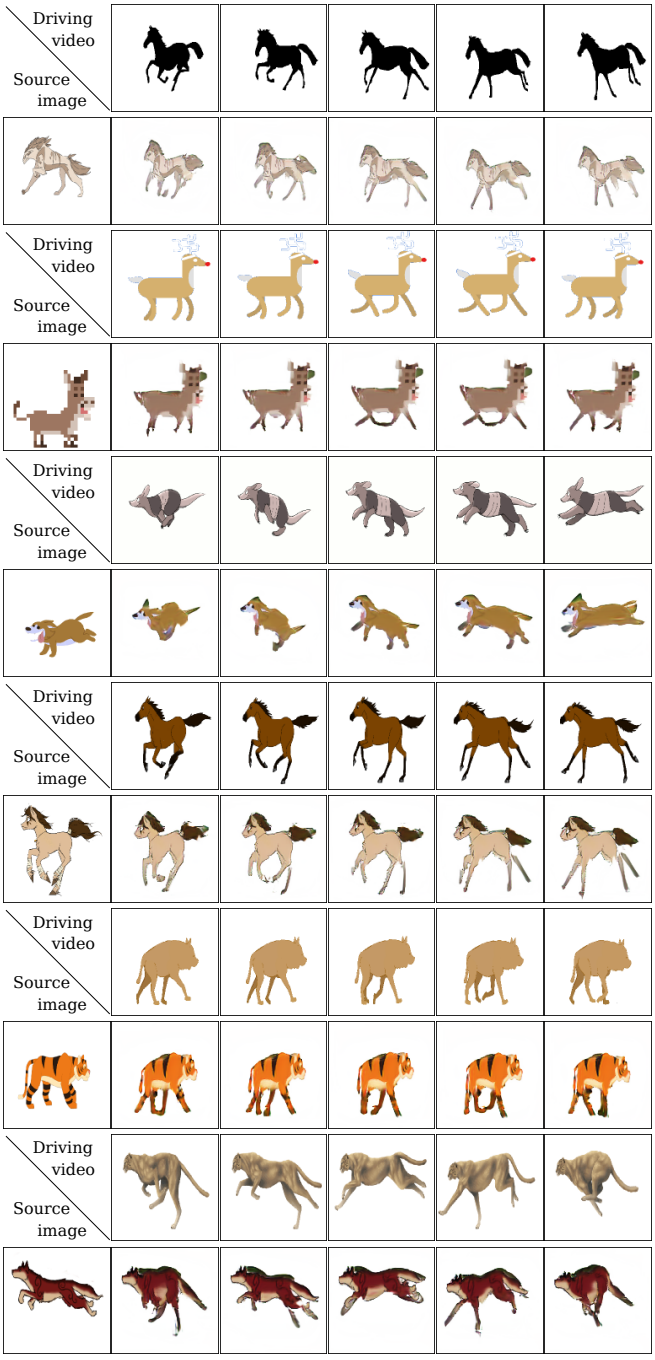

Figure 12: Qualitative results for image animation on the *MGif*.

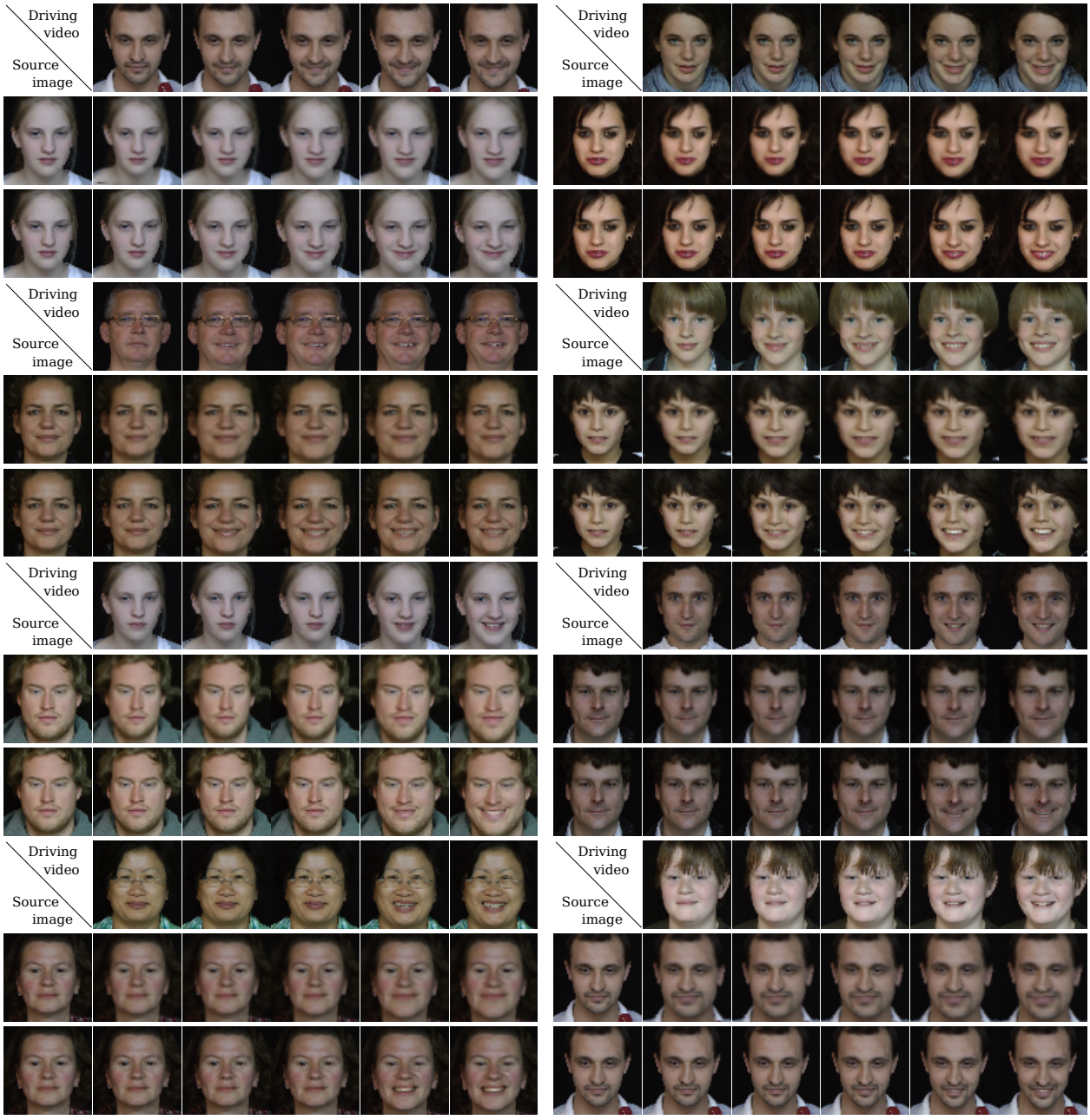

Figure 13: Additional qualitative results for image animation on the *Nemo* dataset: X2face (first) against our method (second).

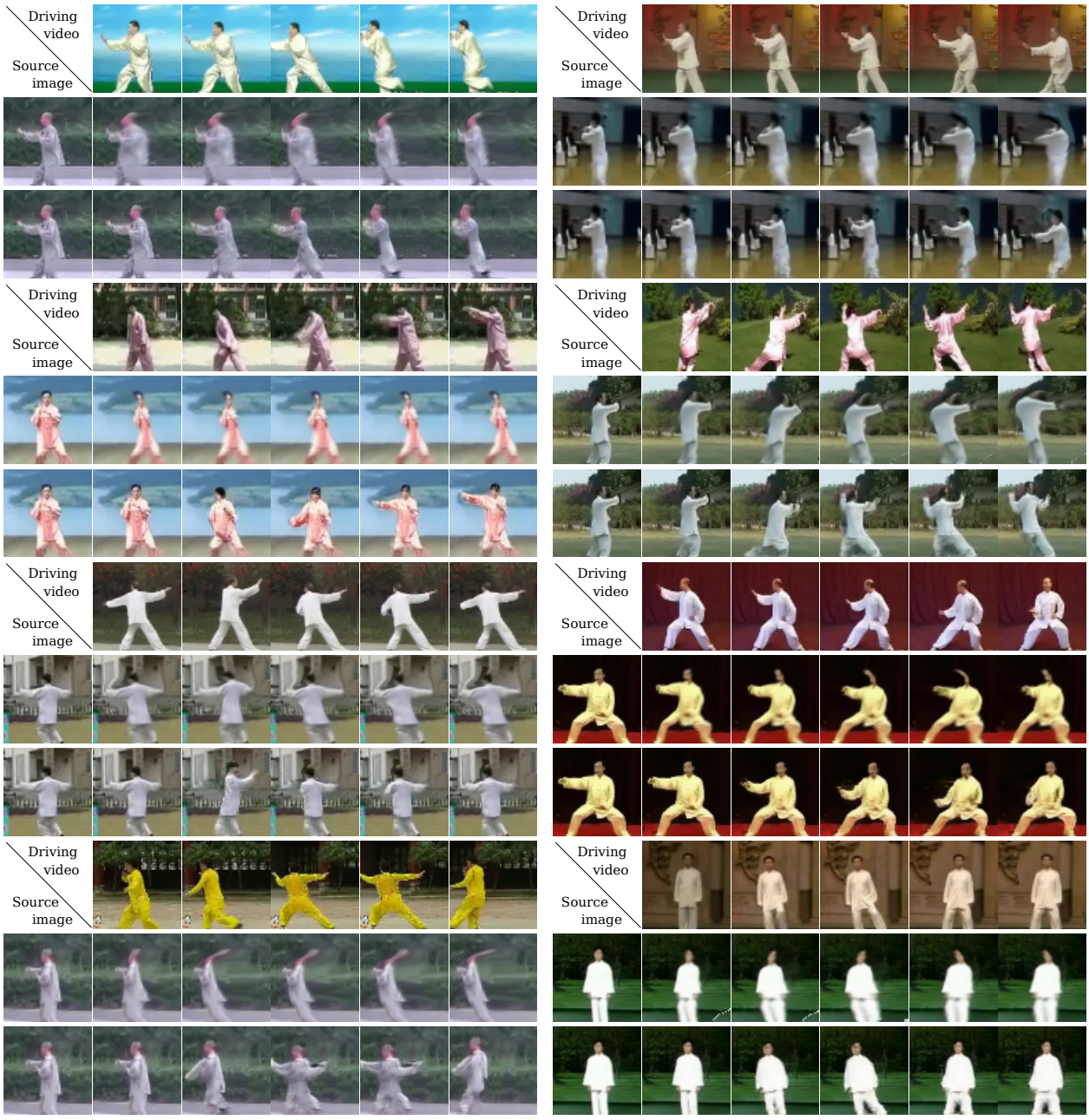

Figure 14: Additional qualitative results for image animation on the *Tai-Chi* dataset: X2face (first) against our method (second).

Figure 15: Additional qualitative results for image animation on the *Bair* dataset: X2face (first) against our method (second).

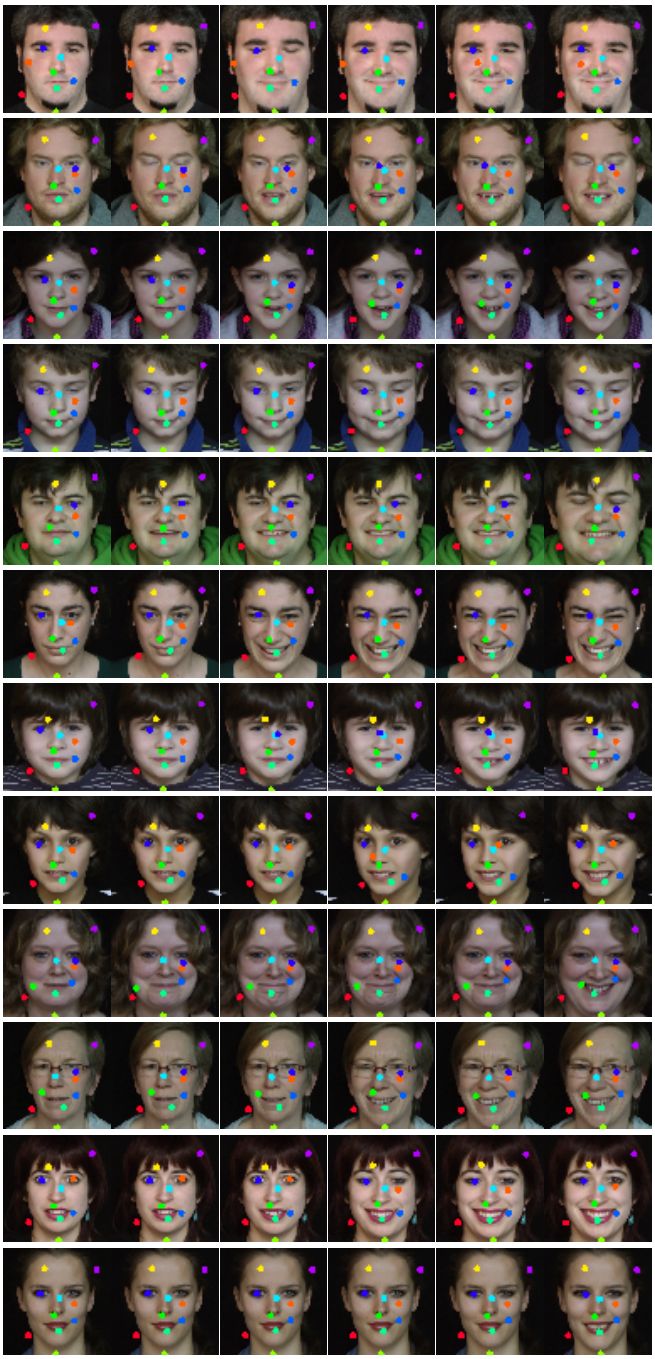

Figure 16: Keypoints predicted on the *Nemo* dataset.

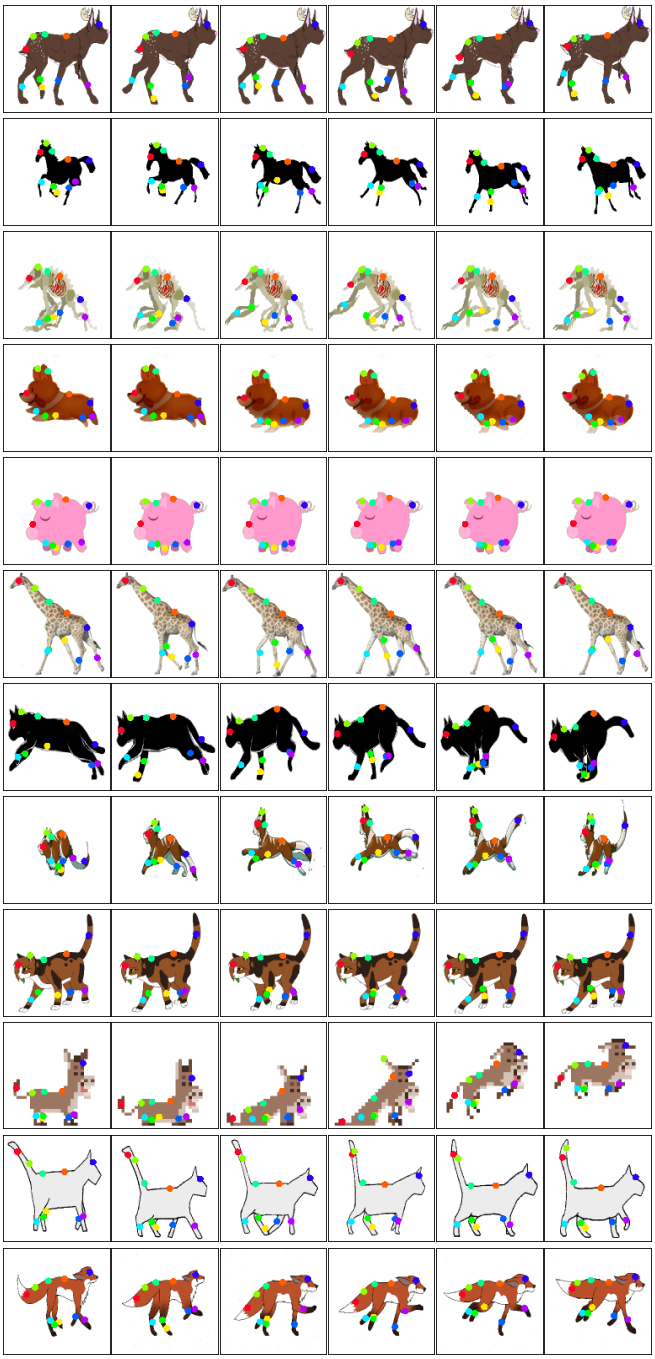

Figure 17: Keypoints predicted on the *MGif* dataset.

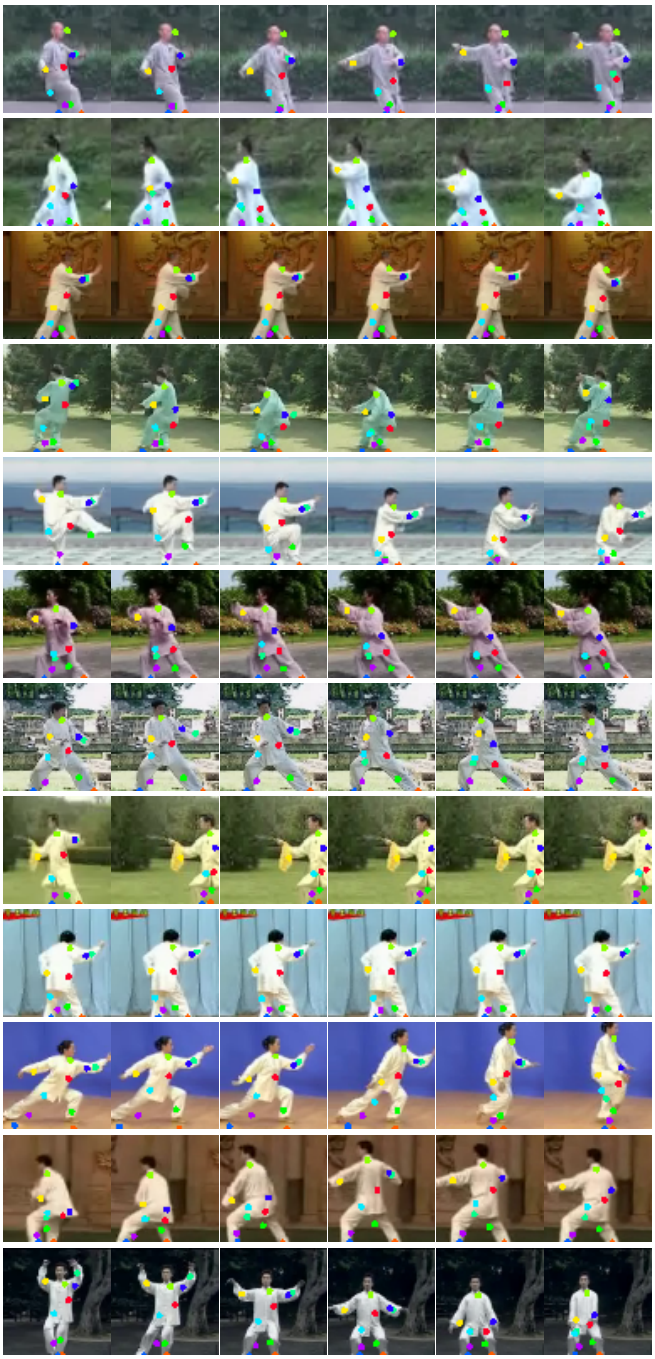

Figure 18: Keypoints predicted on the *Tai-Chi* dataset.

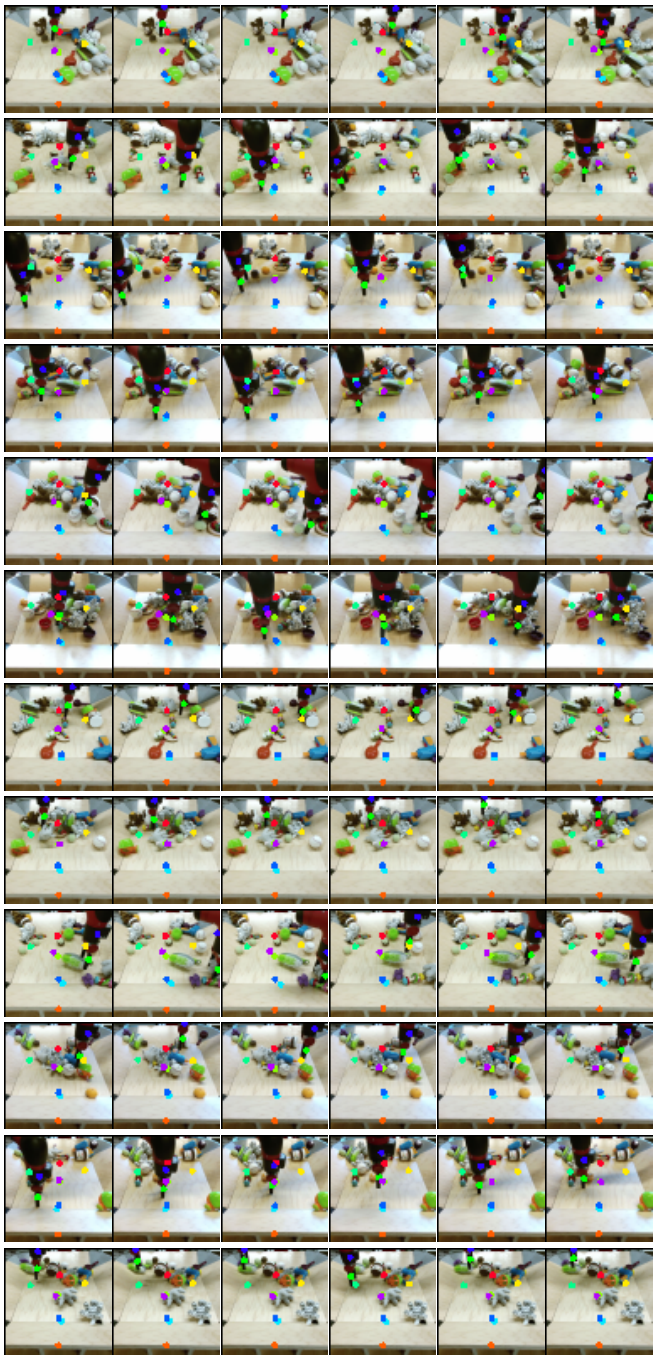

Figure 19: Keypoints predicted on the *Bair* dataset.