
JOKR: Joint Keypoint Representation for Unsupervised Cross-Domain Motion Retargeting

Ron Mokady, Rotem Tzaban, Sagie Benaim, Amit H. Bermano, and Daniel Cohen-Or

The Blavatnik School of Computer Science, Tel Aviv University

Abstract

The task of unsupervised motion retargeting in videos has seen substantial advancements through the use of deep neural networks. While early works concentrated on specific object priors such as a human face or body, recent work considered the unsupervised case. When the source and target videos, however, are of different shapes, current methods fail. To alleviate this problem, we introduce JOKR - a JOint Keypoint Representation that captures the motion common to both the source and target videos, without requiring any object prior or data collection. By employing a domain confusion term, we enforce the unsupervised keypoint representations of both videos to be indistinguishable. This encourages disentanglement between the parts of the motion that are common to the two domains, and their distinctive appearance and motion, enabling the generation of videos that capture the motion of the one while depicting the style of the other. To enable cases where the objects are of different proportions or orientations, we apply a learned affine transformation between the JOKRs. This augments the representation to be affine invariant, and in practice broadens the variety of possible retargeting pairs. This geometry-driven representation enables further intuitive control, such as temporal coherence and manual editing. Through comprehensive experimentation, we demonstrate the applicability of our method to different challenging cross-domain video pairs. We evaluate our method both qualitatively and quantitatively, and demonstrate that our method handles various cross-domain scenarios, such as different animals, different flowers, and humans. We also demonstrate superior temporal coherency and visual quality compared to state-of-the-art alternatives, through statistical metrics and a user study. Source code and videos can be found at: <https://rmokady.github.io/JOKR/>.

1 Introduction

One of the fields that has seen the greatest advancements due to the deep learning revolution is disentangled content creation. Under this paradigm, deep neural networks are leveraged to separate content from style, even when this separation is highly non-trivial. For example, in the image domain, several works have examined disentangling the scene’s geometry (or *content*), from its appearance (or *style*). This enables exciting novel applications in image-to-image translation, such as converting day images to night ones, giving a photo-realistic image the appearance of a painting, and more [10, 22, 14, 21]. It turns out that preserving the geometry and translating the texture of objects in an image is much simpler for networks, compared to performing translations on the geometry itself (e.g., translating a horse to a giraffe) [42, 18]. This task, of transferring shape or pose, is even more difficult when considering videos and motion. To address shape-related translations, many works use a sparse set of 2D locations to describe the geometry. For static images, it has already been shown that these *keypoints* can be learned in an unsupervised manner, enabling content generation that matches in pose (or other geometric features) across domains [42]. For videos, however, success

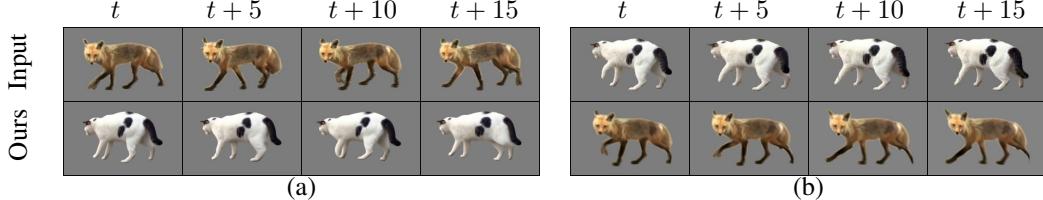


Figure 1: Motion retargeting results from a single video pair. The input videos (top) are used to generate the retargeting (bottom). As can be seen, each generated frame corresponds to the motion portrayed by the source, while keeping the style of the target. For example, note how at frame t (b), both animals simultaneously lift their front leg, however each one does it in its own style, i.e. the generated fox’s leg pose is similar to the original video ($t + 10$ (a)). For all examples, we denote the starting frame as t .

is more limited. To tackle videos, many approaches employ heavy supervision. For example, prior knowledge regarding human body skeletons [7, 32] or human facial expressions [44, 50] is often employed to drive keypoint locations. These approaches are limited to the specific domain of faces or human bodies, where such supervision is abundant, and do not allow transferring or retargeting motion between domains. Unsupervised approaches have also been proposed [4, 34], but they too fall short when it comes to cross-domain inputs (see Section 4).

In this paper, we demonstrate how enforcing shared geometric meaning on the latent representation induces the desired disentanglement between motion and appearance in videos. More specifically, we demonstrate how said representation can be employed to retarget motion across domains in a temporally coherent manner, without the need for any prior knowledge. In other words, we propose to use a joint keypoint representation as a bottleneck to portray pose and motion. Unlike skeletons, which require prior knowledge to drive video generation, we let the network describe the pose and motion by freely choosing keypoint locations, such that the same description is meaningful in both domains. As can be seen in Figure 1, this concept enables cross-domain motion retargeting. By employing a domain confusion term, we enforce the keypoint representations of both domains to be indistinguishable, thus achieving our introduced JOint Keypoint Representation (*JOKR*).

We evaluate the expressiveness of our representation through the setting of cross-domain motion retargeting using a single pair. Using a single video pair alleviates the need for extensive data collection and labelling, which is a costly and exhaustive task, especially for videos. Given a source and target videos, we generate a new video which exhibits the motion of the source video with the style of the target. The design of our joint keypoint representation encourages disentanglement between the motion (content) and the native style of the videos. For example, Figure 1 portrays motion retargeting results between a cat and fox. As can be seen in $t + 10$,(a) both the fox and cat lift their front leg in sync, however the amount that is lifted and the pose of the paws, remain distinctive to each animal to produce a realistic result. Through a novel affine-invariant domain-confusion loss, we prevent the keypoints from capturing information about the image-level texture, shape, or distinctive poses, thus enabling the disentanglement between the parts of the motion that are common to the two domains and their distinctive styles. Furthermore, this geometry-driven representation is also meaningful enough to enable intuitive control, such as imposing temporal coherence or even simple manual editing (see Fig. 6). Note that we use auxiliary input in the form of the object’s silhouette to avoid background related artifacts. Our method operates on both manually annotated silhouettes and silhouettes obtained using an off-the-shelf pretrained saliency segmentation network, thus our method is not limited to specific objects. From the *JOKR* bottleneck, we train two domain specific decoders that are used to generate realistic videos — one for the source video and another for the target. This results in realistic videos which portray one object performing movements of the same meaning and timing as another while keeping the original style.

We evaluate our method both qualitatively and quantitatively on a variety of video pairs from the YouTube-VOS dataset [43] depicting different cross-domain objects. For example, we use animals that exhibit different shapes and styles. Numerically, we demonstrate that our method is superior both in temporal coherency and visual quality. We also demonstrate the capability of our method on other object types such as flowers and dancing persons as well as in the setting of GIF synchronization. We then illustrate how our representation can be leveraged for simple and intuitive manual editing,

demonstrating that our method generates semantic keypoints. Lastly, a comprehensive ablation study is carried, demonstrating the necessity of each component of our method.

2 Related Work

Motion retargeting. Many approaches exist for transferring motion from one video to another. Several works operate in the supervised video-to-video setting [40, 28, 23], which requires supervision in the form of source and corresponding target frames. Other works consider motion specific to the human face or body [3, 45, 26, 19, 27, 38, 2], still requiring a strong prior in the form of extracted landmarks or a 3D model. For instance, Zakharov et al. [45] use facial landmarks to transfer face movement. DeepFake methods [8, 24] were the first to swap faces between two input videos using deep neural networks. These methods either implicitly use 3D facial representations, or use facial landmarks to crop and align the face, and are again limited to facial data. Other works [7, 1, 9] transfer motion from one human body to the other using extracted human silhouettes or 2D skeletons. Unlike these approaches, our method does not assume any specific prior.

Siarohin et al. [33, 34] and Wiles et al. [41] assume no prior and consider the task of image animation given a source video and a target image. As the target video is not provided, the motion is borrowed completely from the source video, meaning that in the case of transferring between different objects, the resulting motion is unrealistic. Motion synchronization [5] can also be used, but it cannot generate novel frames, therefore is limited to only reordering the motion. Most related to ours is the work of RecycleGAN [4], which considers unsupervised motion retargeting using a single video pair. The approach is based on cycle consistency losses in space and time, and adversarial losses. However, their approach struggles to align objects with a substantially different shape. Using our joint keypoint representation that encodes motion common to both videos, our method correctly handles a variety of cross-domain scenarios, where objects are of different shapes.

Shared Geometric Representation. A large body of works considers the unsupervised learning of keypoint representations [36, 47, 42, 35, 15, 33, 34]. Some, by directly training an autoencoder to reconstruct the given images [47], while others by solving a downstream task, such as conditional image generation [15]. Wu et al. [42] translate between objects of different shapes by using keypoints as a shared geometric representation. However, unlike our method, their method does not contain any temporal constraint, intuitive editing, nor does it demonstrate any result over videos. Jakab et al. [16] use unpaired pose prior to train a keypoint extractor, but their work is also limited to humans. Similar to ours, Siarohin et al. [34] learn keypoint representations in an unsupervised fashion, however, as we demonstrate (see Section 4), their method cannot handle cross-domain videos well. The use of a shared representation is also prevalent in other image generation tasks. UNIT [22] and MUNIT [14], for example, use a shared representation for image-to-image translation. Other works [29, 6, 25] use shared representations to disentangle the common content of two domains from the separate part. Unlike these methods, our work disentangles motion from style over videos.

3 Method

We consider two input videos $A = \{a_i\}_{i=0}^{N_A-1}$ and $B = \{b_i\}_{i=0}^{N_B-1}$, where a_i (resp. b_i) is the i th frame of A (resp. B). WLOG, we wish to generate a new video $AB = \{ab_i\}_{i=0}^{N_B-1}$ portraying the analogues motion of B , while maintaining the appearance and style of A . We assume a single object is present in each video. To alleviate cases where the background rapidly changes, we mask out the background of each frame. The segmentation maps $\{s_{a,i}\}_{i=0}^{N_A-1}, \{s_{b,i}\}_{i=0}^{N_B-1}$ are given as part of the data, or are acquired using off-the-shelf image segmentation networks (see Sec. 4). We now describe the different components of our method. An illustration is provided in Figure 2, and a visual example of the intermediate results is shown in Figure 3. For brevity, we describe the retargeting from A (source) to B (target), keeping in mind that the opposite direction is symmetrical.

Shape-Invariant Representation. To translate between objects of different shapes and texture, we use JOKR as a bottleneck. As manual keypoint annotation is not always available (see Section 1), we use an unsupervised keypoint extractor E , similar to previous work [35, 33, 34], to extract K keypoints, denoted $k_{a,i} = \{k_{a,i}^\ell\}_{\ell=0}^{K-1}$. To leverage the convolutional network’s ability to utilize spatial information, we project the extracted keypoints to spatial maps by fitting a Gaussian for each keypoint, obtaining K confidence maps $h_{a,i} = \{h_{a,i}^\ell\}_{\ell=0}^{K-1}$ (see Appendix B for more details).

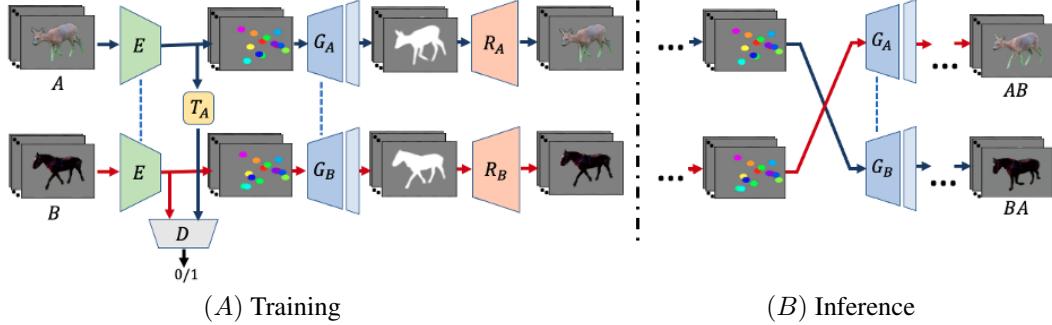


Figure 2: Method illustration. (A) Training: Given Videos A and B , we extract our Joint Keypoint Representation using network E . Discriminator D is used to encourage the representation to be of the same distribution, up to the learned affine transformation T_A . G_A and G_B , which share weights for all but the last layer, translate the given keypoint to a segmentation map that is then refined by R_A and R_B . (B) Inference: We pass the keypoints of A (resp. B) to G_B (resp. G_A). The result BA depicts the motion of A , as represented by the keypoints, and the appearance and style of B , due to the usage of G_B and R_B . The result AB is its counterpart.

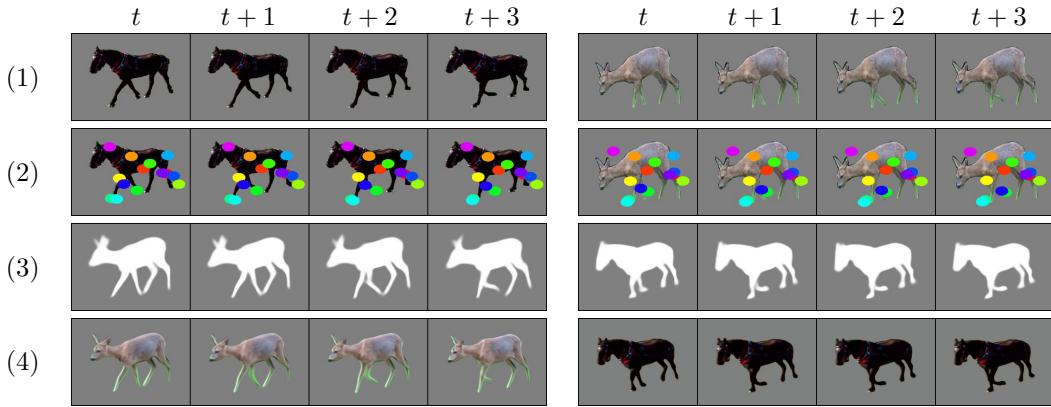


Figure 3: Intermediate results. Given input images (1) from video A (resp. B), we extract corresponding keypoints (2), which are then translated to analogue shape of B (resp. A) (3), then the final texture is added (4). Zoom-in is recommended.

To encourage the disentanglement between geometry and appearance, the generation process is divided into two steps. First, given $h_{a,i}$, generator G_A is trained to output a silhouette that corresponds to the extracted keypoints and at the same time to the shape of the object in A . To reduce the number of parameters, G_A and G_B share the same weights, except for the last layer. Similarly, the same keypoint extractor E is used for both videos. Formally, given frames a_i, b_j , we generate the silhouettes by minimizing the following MSE loss:

$$\mathcal{L}_{\text{seg}} = \sum_{i=0}^{N_A-1} \|G_A(E(a_i)) - s_{a,i}\|_2 + \sum_{j=0}^{N_B-1} \|G_B(E(b_j)) - s_{b,j}\|_2 \quad (1)$$

We now train generators R_A and R_B to translate the obtained segmentation map to the original image, thereby adding texture. Specifically we consider the following reconstruction and perceptual losses:

$$\mathcal{L}_{L1} = \sum_{i=0}^{N_A-1} \|R_A(G_A(E(a_i))) - a_i\|_1 + \sum_{j=0}^{N_B-1} \|R_A(G_B(E(b_j))) - b_j\|_1 \quad (2)$$

$$\mathcal{L}_{\text{LPIPS}} = \sum_{i=0}^{N_A-1} \|\mathcal{F}(R_A(G_A(E(a_i)))) - \mathcal{F}(a_i)\|_2 + \sum_{j=0}^{N_B-1} \|\mathcal{F}(R_A(G_B(E(b_j)))) - \mathcal{F}(b_j)\|_2 \quad (3)$$

\mathcal{L}_{L1} uses the $L1$ norm and $\mathcal{L}_{\text{LPIPS}}$ uses the LPIPS perceptual loss proposed by Zhang et al. [46], where the feature extractor is denoted by \mathcal{F} .

Shared representation. Videos A and B may depict objects from a different domain, and so the extracted keypoints for each video may have a different semantic meaning. For example, the same keypoint might represent a leg in one video and tail in the other. Therefore, we enforce that the encoded keypoints for both videos are from a shared distribution, thus encouraging the keypoints to capture motion that is common to both videos. The specific style of A or B is then encoded in the generator’s weights. To enforce the shared distribution between the keypoints, we use a domain confusion loss [37]. Specifically, a discriminator D is used to distinguish between keypoints of domains A and those of B , while the encoder is trained adversarially to fool the discriminator, thus forcing the keypoints of the two domains to statistically match:

$$\mathcal{L}_{\text{DC}} = \sum_{i=0}^{N_A-1} \ell_{\text{bce}}(D(k_{a,i}), 1) + \sum_{j=0}^{N_B-1} \ell_{\text{bce}}(D(k_{b,j}), 1) \quad (4)$$

where we use the binary cross entropy loss function $\ell_{\text{bce}}(p, q) = -(q \log(p) + (1 - q) \log(1 - p))$. While the keypoint extractor E attempts to make the keypoints distributions indistinguishable, the discriminator is trained adversarially using the objective function:

$$\mathcal{L}_D = \sum_{i=0}^{N_A-1} \ell_{\text{bce}}(D(k_{a,i}), 0) + \sum_{j=0}^{N_B-1} \ell_{\text{bce}}(D(k_{b,j}), 1) \quad (5)$$

In some cases, the object appearing in video A can be of different proportions or location in the images, which makes it already distinguished from B , just based on a change in rotation, scale or translation. We therefore augment Eq. 4 and Eq. 5 so that our domain confusion loss is invariant to affine transformations, thus enabling a broader variety of possible retargeting pairs. To do so, a learned affine transformation T_A is applied to B ’s keypoints before passing them to discriminator D , where T_A is optimized with the keypoint extractor E .

Temporal Coherence. We would like to ensure that the generated videos are temporally coherent. That is, a smooth and non-jittery motion is generated. To this end, we apply a temporal regularization on the generated keypoints and minimize the distance between keypoints in adjacent frames:

$$\mathcal{L}_{\text{tmp}} = \sum_{i=0}^{N_A-1} \|k_{a,i} - k_{a,i+1}\|_2 + \sum_{j=0}^{N_B-1} \|k_{b,j} - k_{b,j+1}\|_2 \quad (6)$$

Since JOKR is encoded for every frame only from its respective image, sometimes flickering is introduced because a keypoint shifts in meaning between frames (e.g., a keypoint describing a back leg suddenly describes the tail). We observe this usually happens when the figures undergo large motion. Hence, similarly to Siarohin et al., [34], we ensure that the generated keypoints are equivariant under an arbitrary affine transformation; We apply a random affine transformation on the keypoints and the original frame, and compare the transformed keypoints with the keypoints extracted from the transformed image. This ensures the semantic meaning of each keypoint is consistent, and significantly improves coherency, since decoding temporally coherent keypoints results in temporally coherent frames. For an affine transformation T , transformation equivariance loss is defined as:

$$\mathcal{L}_{\text{eq}} = \sum_{i=0}^{N_A-1} \|T(E(a_i)) - E(T(a_i))\|_1 + \sum_{j=0}^{N_B-1} \|T(E(b_j)) - E(T(b_j))\|_1 \quad (7)$$

Keypoints Regularization. Ensuring a shared representation and temporal coherence is important, but not sufficient to ensure the encoded keypoints capture meaningful information about motion. Specifically, the keypoints might collapse to a single point without any relation to the object itself. Therefore, we suggest additional two loss terms based on the terms used by Suwajanakorn et al. [35]. First, we use a separation loss which prevents the keypoints sharing the same location, by penalizes two keypoints if they are closer than some hyperparameter threshold δ :

$$\mathcal{L}_{\text{sep}} = \frac{1}{K^2} \sum_{\ell=0}^{K-1} \sum_{\ell \neq r} \left(\sum_{i=0}^{N_A-1} \max(0, \delta - \|k_{a,i}^\ell - k_{a,i}^r\|^2) + \sum_{j=0}^{N_B-1} \max(0, \delta - \|k_{b,j}^\ell - k_{b,j}^r\|^2) \right)$$

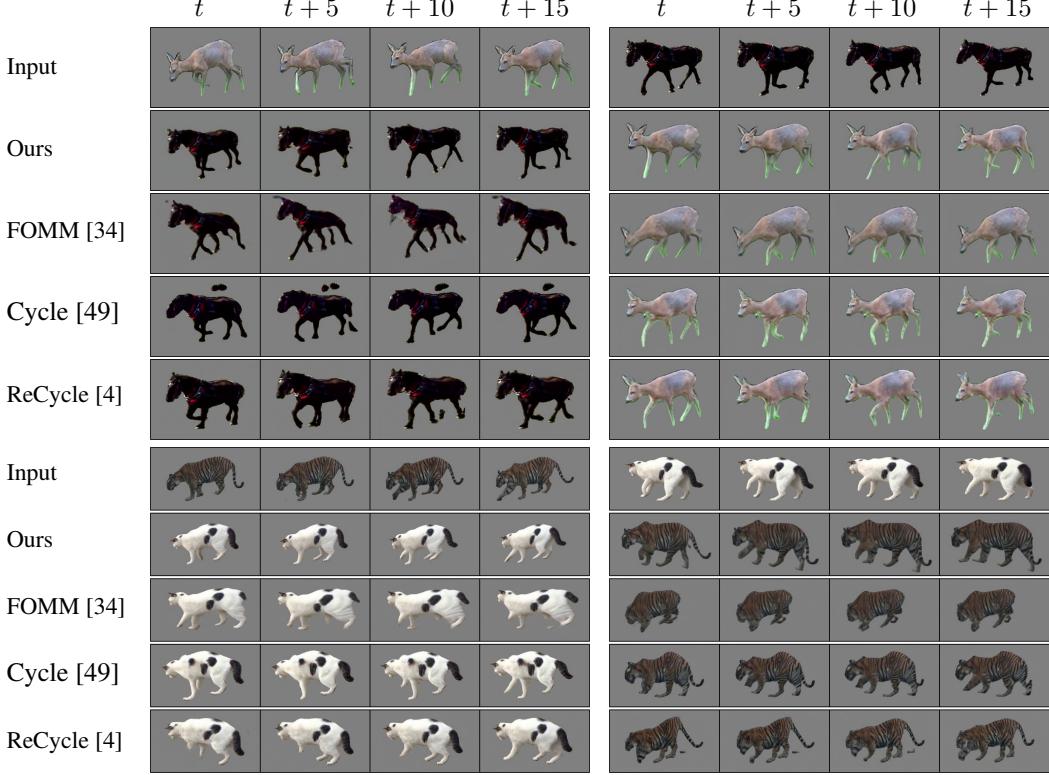


Figure 4: A comparison of our method to baselines. As can be seen, ours is the only one to transfer the correct pose while faithfully generating the target style. Zoom-in recommended.

where $k_{a,i} = \{k_{a,i}^\ell\}_{\ell=0}^{K-1}$ are the extracted keypoints from frame a_i . Second, we use silhouette loss to encourage the keypoints to lie on the object itself:

$$\mathcal{L}_{\text{sill}} = \frac{1}{K} \sum_{\ell=0}^{K-1} \left(\sum_{i=0}^{N_A-1} -\log \sum_{u,v} s_{a,i}(u,v) H_{a,i}^\ell(u,v) + \sum_{j=0}^{N_B-1} -\log \sum_{u,v} s_{b,j}(u,v) H_{b,j}^\ell(u,v) \right)$$

where the sum $\sum_{u,v}$ is over all image pixels (u,v) and $H_{a,i}^\ell$ is the heatmap generated by the keypoint extractor for the ℓ th keypoint from frame a_i (see implementation details in Appendix B). Without this loss, the representation might focus on meaningless regions of the image, rendering some of the points irrelevant.

Two-Steps Optimization. Since most loss terms are related to shape and not to texture, we can optimize our objective function in two steps. First, we train the discriminator using \mathcal{L}_D , while also training the networks E , G_A , and G_B . In the second step, we train the refinement networks R_A , R_B to add the texture, using the aforementioned \mathcal{L}_{L1} and $\mathcal{L}_{\text{LPIPS}}$ losses. For challenging textures, such as in Fig. 5 (bottom), we employ additional adversarial loss over the generated frames at the second stage training (texture), using the discriminator and adversarial loss proposed by Wang et al. [39].

Augmentations. Using adversarial loss with very limited data might cause mode collapse. Thus, we use augmentations in the form of random affine transformations during training. However, when preserving the background is necessary as in Fig. 5 (Bottom), these augmentations might leak to the generated frames resulting with artifacts. Hence, similar to [48, 17], we perform the augmentations directly over the keypoints just before passing them to the discriminator, resulting with less artifacts and stable training.

Inference. Since the distribution of keypoints match, we can use G_A , R_A to translate the keypoints of B to the appearance and style of A , while also adhering to the motion of B : $ab_j = R_A(G_A(T(E(b_j))))$. Note that the learned affine transform T may be omitted. Using T , the generated object will more faithfully preserve the target object's rotation, scale and translation. Without it, the source object's rotation, scale and translation will be better preserved (See Fig. 9).

Measure	Ours	FOMM [34]	CycleGAN [49]	RecycleGAN [4]
FID ↓	39.01	74.35	86.41	55.0
SVFID ↓	294.89	345.1	349.43	317.84
User Study - Appearance ↑	3.91	2.07	2.64	2.27
User Study - Motion ↑	3.27	1.57	3.61	2.75

Table 1: Numerical evaluation for our method and baselines. FID and SVFID are used to measure the temporal realism of generated videos (lower is better). A user study is used to analyze the appearance and motion consistency, measured by mean opinion score (higher is better).

4 Results

We evaluate our method both qualitatively and quantitatively on video pairs of walking animals from the YouTube-VOS dataset [43]. All pairs are challenging as they contain different shapes such as a cat/fox or a deer/horse. To demonstrate versatility, we also present results for humans and flowers, and for synchronization of short GIFs. For segmentation, we used ground-truth if available or extract it using a pretrained network, see more details in Appendix B. In addition, we show that our learned keypoints have semantic meaning by performing simple editing. Lastly, an ablation analysis is performed to illustrate the effectiveness of the different components. For all experiments, videos can be found at our webpage: <https://rmokady.github.io/JOKR/>.

Qualitative evaluation. We consider several baselines most related to ours. First, we consider FOMM [34], which, similarly to ours, learn a keypoint representation in an unsupervised fashion. A second baseline is that of CycleGAN [49], where a cycle loss is used to perform unsupervised image-to-image translation for every frame. Since CycleGAN is normally trained on many images, we perform augmentations to avoid overfitting. Lastly, we compare to ReCycleGAN [4], which extends CycleGAN by employing additional temporal constraints. Fig. 4 gives a visual comparison of our method (additional results are in Fig. 11 to Fig. 16, videos can be found at the webpage). As can be seen, for both examples in Fig. 4, our method correctly transfers the leg movement present in the source video, while preserving the style of the target. For instance, the horse legs appear realistic and are not of the same shape as the input deer. As can be seen, FOMM is unable to produce realistic target shapes, as the warping mechanism matches the keypoint locations themselves, without considering the different semantic meanings of the two domains. CycleGAN struggles in changing the deer’s shape to a realistic horse shape. ReCycleGAN manages to alter the shape correctly, but suffers from significant artifacts, such as a missing leg.

Quantitative evaluation. To evaluate the realism of generated frames, we use the FID metric [13] over each one. For temporal consistency, we adopt the recently proposed SVFID score introduced by Gur et al. [12]. SVFID is an extension of FID for a single video, evaluating how the generated samples capture the temporal statistics of a single video, by using features from a pretrained action recognition network. We compare video *AB* (resp. *BA*) and video *A* (resp. *B*), and report a superior result (Tab. 1). To evaluate the quality of the motion transfer, we performed a user study, as we do not have ground truth keypoint supervision. For each video pair, users were asked to rank from 1 to 5: (1) How much the appearance of *AB* looks realistic compared to *B*? (User Study - Appearance) and (2) How similar is the motion of *A* to that of *AB*? (User Study - Motion). As can be seen, our generated videos are much more realistic than all baselines. While CycleGAN results are better for motion transfer, it scores significantly worse both in appearance and on FID/SVFID scores. The reason is that CycleGAN struggles with geometric changes and focuses on the texture. For example, the horse in Fig 4 is well aligned but bears very different appearance from the original horse.

GIF synchronization. We demonstrate our ability to handle very modest data, through the GIF synchronization setting, where the number of different frames is approximately 40. We use video pairs, depicting different objects from the MGif dataset [33]. Synchronization of an elephant video and a giraffe video is shown in Fig. 5 where our method successfully generates an analogue motion. In addition, we successfully keep the target object’s appearance, even when the shapes and texture of source and target videos are different. For example, the same leg moves forward at the same time for both input and respective generated sequence, while the way the leg moves (in terms of length, bendiness, and stride size) is preserved well. As motion range is limited, our generated frames are similar to the input ones, hence we define this result as synchronization. Additional result in Fig. 10.

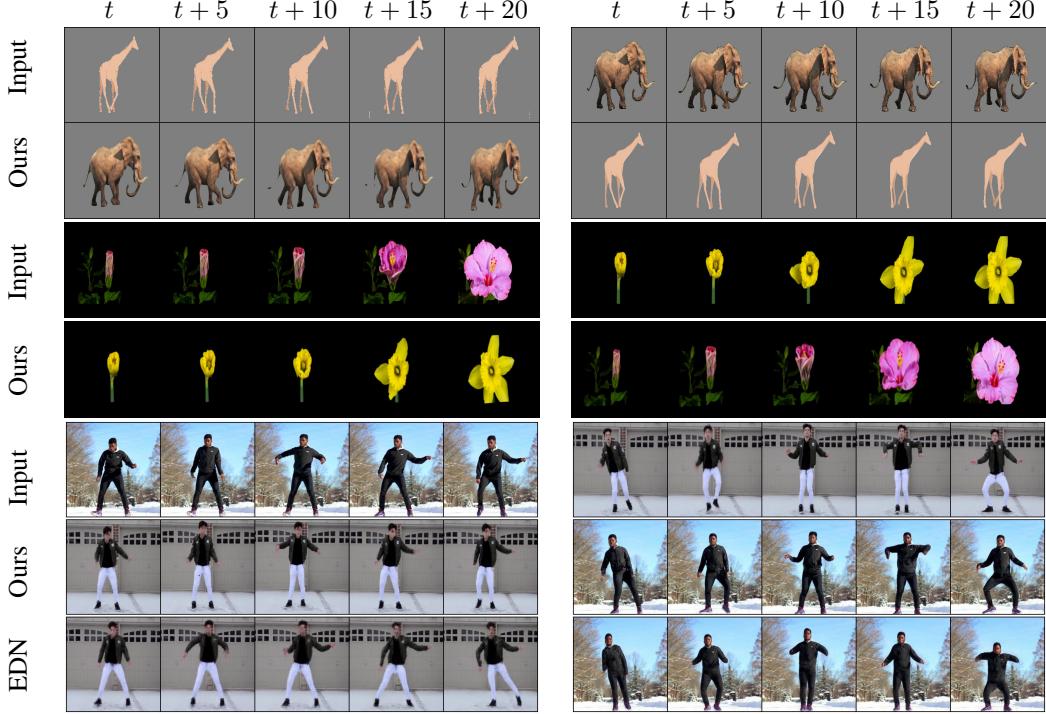


Figure 5: Top: GIF Synchronization for a video pair of a zebra and a giraffe. Middle: Two videos of blooming flowers. Bottom: Pairs of dancing persons videos with comparison to EDN [7].

Other domains Fig. 5 (Middle) presents blooming flowers as used in Bansal et al. [4], and Fig. 5 (Bottom) presents dancers. For the dancers, we compare against EDN [7]. The latter assumes the supervision of a semantic skeleton, and so our setting is more challenging. Further, we use very short (less than a minute) videos that consist of a wide range of different motions. As can be seen, our method is comparable to EDN. Additional results are in Fig .27, Fig .25, and Fig .26

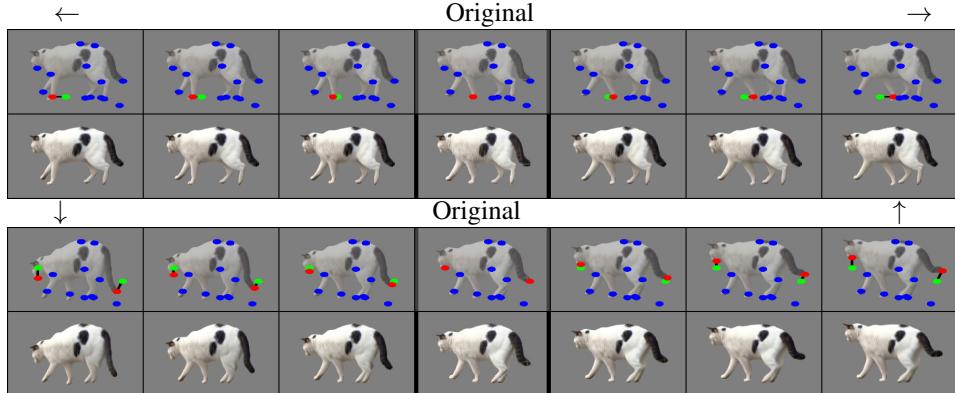


Figure 6: Moving the keypoints linearly induces semantic editing. All keypoints are colored in blue, while the original location of the edited keypoints in green and their new location in red. Top: We manipulate the front leg by moving the corresponding keypoint right and left. Bottom: We now move both the head and the tail up and down, where additional small movement right and left (resp.) is applied. As the red points move, we get a corresponding movement in the generated frame.

Keypoint Interpretability. To demonstrate the semantic meaning of our generated keypoints, we perform a simple, yet effective, editing procedure of manual manipulation of the obtained keypoints. For a given real frame a_i , we select one or two of the keypoints generated using $E(a_i)$ (see Sec. 4). We then move these keypoints linearly and generate the corresponding frame by passing the new set of keypoints to the generators. Results presented in Fig. 6 demonstrate that this type of editing yields semantic manipulation of the given object (additional results in Fig. 24).

Ablation Study. An ablation analysis is presented in Fig. 7 for a horse and deer: (a) Original horse frames. (b) Results obtained by the motion of the horse with the style of the deer. (c) The equivariance loss is omitted, resulting in inconsistent motion, such as a separated leg. (d) Without augmentations overfitting occur, resulting in significant artifacts. (e) The domain-confusion loss is omitted, causing the keypoints to contain vital information about the shape, resulting in the horse shape and the deer’s texture. (f) Omitting the temporal regularization decreases temporal coherency and sometimes causes additional artifacts such as a missing head. (g) Avoiding the two-step approach, that is the result is generated directly from the keypoint, without generating the shape first. As can be seen, in this case the appearance is inferior.

We show additional ablations for temporal regularization in Fig. 8, focusing on the keypoint locations with and without regularization. Without regularization (b),(c) the keypoints movement from one frame to another is not proportional to the object’s movement, resulting in inferior temporal consistency compared to using the regularization (d),(e). Video can be found at the webpage. We also measure the normalized distance between the same keypoints across adjacent frames. As expected, the distance without temporal regularization (0.0055) is substantially higher than using the regularization (0.0023).

Lastly, we demonstrate the effect of the learned affine transformation in Fig 9. Given a video of a horizontal cat (a), and rotated tiger (b), the learned affine transformation results in rotating and scaling the object. Therefore, applying the transformation at inference (e) resulting in a horizontal cat as the transformation fixes the rotation. Without applying it at inference (d), the result is perfectly aligned to the source video depicting the exact rotation and scale. Omitting the learned affine transformation at training (c) reduces the stability of the domain confusion loss, which sometimes results in artifacts.

Limitations. Looking forward, there is much room for further investigation. Currently, JOKR is agnostic to affine disproportions between the source and target videos. Beyond that, the retargeted videos should still bear some similarities, such as topology. Second, while multiple objects can be handled separately, our method is unable to handle complicated scenes with multiple objects which may occlude each other. We also note that we ignore the background at the current scope of the paper. Requiring these segmentation maps implies supervision, however as we demonstrate, using off-the-shelf tools suffices in our case. Lastly, we note that training time is around 12 hours for a single NVIDIA GTX1080 GPU, and reducing training time could be helpful as future work.

5 Conclusion

We presented a method for unsupervised motion retargeting. Our key idea is to use a joint keypoint representation to capture motion that is common to both the source and target videos. JOKR demonstrates how imposing a bottleneck of geometric meaning, the aforementioned semantics are encouraged to adhere to geometric reasoning. Hence, poses that bear geometric similarities across the domains are represented by the same keypoints. Even more so, our editing experiments suggest that these semantics are intuitively interpretable. We demonstrate that such a representation can be used to retarget motion successfully across different domains, where the videos depict different shapes and styles, such as four-legged animals, flowers, and dancers. Through our GIFs example, we demonstrate that this representation can be jointly distilled even for short clips, where due to the low range of motion, the network reverts to simple synchronization successfully. Moving forward, challenges remain with videos depicting moving backgrounds, multiple objects, or different topology.

References

- [1] Kfir Aberman, Mingyi Shi, Jing Liao, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or. Deep video-based performance cloning. In *Computer Graphics Forum*, volume 38, pages 219–233. Wiley Online Library, 2019.
- [2] Kfir Aberman, Rundi Wu, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or. Learning character-agnostic motion for motion retargeting in 2d. *arXiv preprint arXiv:1905.01680*, 2019.
- [3] Hadar Averbuch-Elor, Daniel Cohen-Or, Johannes Kopf, and Michael F Cohen. Bringing portraits to life. *ACM Transactions on Graphics (TOG)*, 36(6):1–13, 2017.
- [4] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. Recycle-gan: Unsupervised video retargeting. In *Proceedings of the European conference on computer vision (ECCV)*, pages 119–135, 2018.
- [5] Jean-Charles Bazin and Alexander Sorkine-Hornung. Actionsnapping: Motion-based video synchronization. In *European Conference on Computer Vision*, pages 155–169. Springer, 2016.
- [6] S. Benaim, M. Khaitov, T. Galanti, and L. Wolf. Domain intersection and domain difference. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, South Korea*, pages 3444–3452, 2019.
- [7] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5933–5942, 2019.
- [8] deepfakes (alias). github faceswap project. 2018.
- [9] Oran Gafni, Oron Ashual, and Lior Wolf. Single-shot freestyle dance reenactment. *arXiv preprint arXiv:2012.01158*, 2020.
- [10] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6 2016.
- [11] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7297–7306, 2018.
- [12] Shir Gur, Sagie Benaim, and Lior Wolf. Hierarchical patch vae-gan: Generating diverse videos from a single sample. *arXiv preprint arXiv:2006.12226*, 2020.
- [13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc.
- [14] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 179–196, Cham, 2018. Springer International Publishing.
- [15] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. *arXiv preprint arXiv:1806.07823*, 2018.
- [16] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Self-supervised learning of interpretable keypoints from unlabelled videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8787–8797, 2020.
- [17] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *arXiv preprint arXiv:2006.06676*, 2020.

- [18] Oren Katzir, Dani Lischinski, and Daniel Cohen-Or. {CROSS}-{}domain {}cascaded} {}deep} {}translation}, 2020.
- [19] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018.
- [20] D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations (ICLR)*, 2016.
- [21] Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang. Drit++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision*, 128(10):2402–2417, 11 2020.
- [22] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 700–708, Red Hook, NY, USA, 2017. Curran Associates Inc.
- [23] Arun Mallya, Ting-Chun Wang, Karan Sapra, and Ming-Yu Liu. World-consistent video-to-video synthesis. In *Proceedings of the European Conference on Computer Vision*, 2020.
- [24] Momina Masood, Marriam Nawaz, Khalid Mahmood Malik, Ali Javed, and Aun Irtaza. Deep-fakes generation and detection: State-of-the-art, open challenges, countermeasures, and way forward. *arXiv preprint arXiv:2103.00484*, 2021.
- [25] Ron Mokady, Sagie Benaim, Lior Wolf, and Amit Bermano. Mask based unsupervised content transfer. *arXiv preprint arXiv:1906.06558*, 2019.
- [26] Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, and Hao Li. pagan: real-time avatars using dynamic textures. *ACM Transactions on Graphics (TOG)*, 37(6):1–12, 2018.
- [27] Yuval Nirkin, Yosi Keller, and Tal Hassner. Fsgan: Subject agnostic face swapping and reenactment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7184–7193, 2019.
- [28] Junting Pan, Chengyu Wang, Xu Jia, Jing Shao, Lu Sheng, Junjie Yan, and Xiaogang Wang. Video generation from single semantic label map. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2019.
- [29] Ori Press, Tomer Galanti, Sagie Benaim, and Lior Wolf. Emerging disentanglement in auto-encoder based unsupervised image content transfer. In *International Conference on Learning Representations*, 2019.
- [30] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, 106:107404, 2020.
- [31] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]).
- [32] Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, Igor Pasechnik, Dmitry Ulyanov, et al. Textured neural avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2387–2397, 2019.
- [33] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2377–2386, 2019.
- [34] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *Advances in Neural Information Processing Systems*, 32:7137–7147, 2019.

- [35] Supasorn Suwajanakorn, Noah Snavely, Jonathan Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. *arXiv preprint arXiv:1807.03146*, 2018.
- [36] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *Proceedings of the IEEE international conference on computer vision*, pages 5916–5925, 2017.
- [37] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.
- [38] Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. Neural kinematic networks for unsupervised motion retargetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8639–8648, 2018.
- [39] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA*, pages 8798–8807, 2018.
- [40] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018.
- [41] Olivia Wiles, A Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European conference on computer vision (ECCV)*, pages 670–686, 2018.
- [42] W. Wu, K. Cao, C. Li, C. Qian, and C. C. Loy. Transgaga: Geometry-aware unsupervised image-to-image translation, long beach, usa. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8004–8013, 2019.
- [43] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 585–601, 2018.
- [44] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9459–9468, 2019.
- [45] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9458–9467, 2019.
- [46] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018.
- [47] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2694–2703, 2018.
- [48] Zhengli Zhao, Zizhao Zhang, Ting Chen, Sameer Singh, and Han Zhang. Image augmentations for gan training. *arXiv preprint arXiv:2006.02595*, 2020.
- [49] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy*, pages 2242–2251, 2017.
- [50] Michael Zollhöfer, Justus Thies, Pablo Garrido, Derek Bradley, Thabo Beeler, Patrick Pérez, Marc Stamminger, Matthias Nießner, and Christian Theobalt. State of the art on monocular 3d face reconstruction, tracking, and applications. In *Computer Graphics Forum*, volume 37, pages 523–550. Wiley Online Library, 2018.



Figure 7: Ablation Study. (a) original horse frames. (b) our results. (c) without equivariance loss. (d) without augmentations. (e) without domain-confusion loss. (f) without temporal regularization. (g) without two-step approach, i.e. generating the result directly from the keypoints.

Appendix

A Ablation Study

An ablation study is provided in the main text, we present the visual results in Fig. 7, Fig. 8, and Fig 9.

B Implementation details

B.1 Keypoint extraction

For the keypoint extractor E , similarly to previous work [35, 33, 34], we employ a U-Net [31] architecture which estimates K heatmaps from the input image and uses a spatial softmax layer to

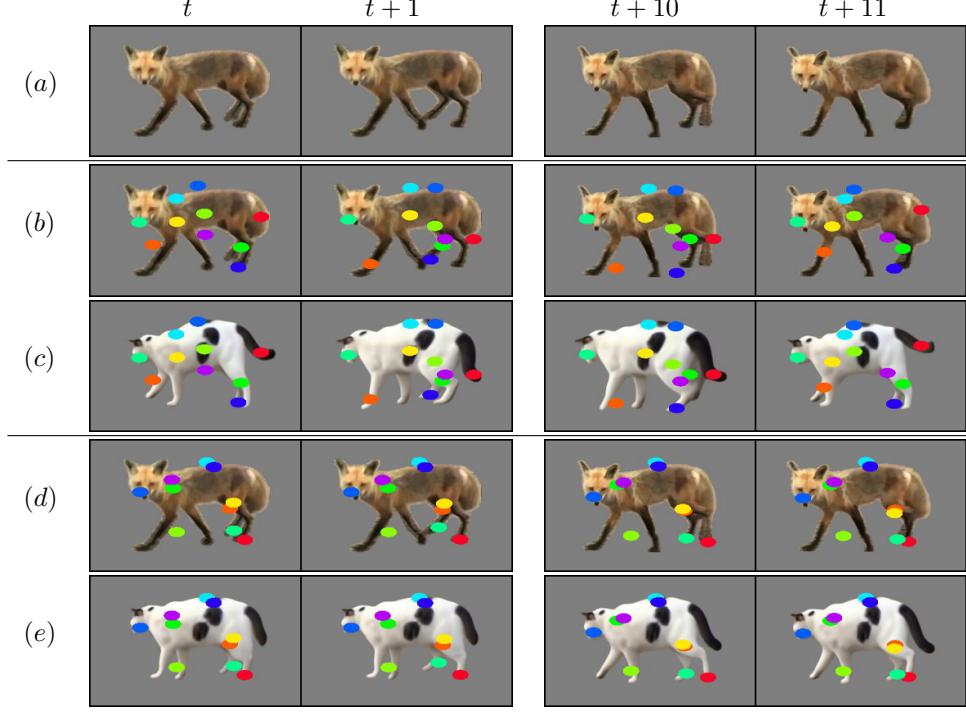


Figure 8: Ablation study for the temporal regularization. (a) Input. (b) Extracted keypoints without temporal regularization, presented on the top of the input frame. (c) Result using the extracted keypoints without temporal regularization, presented with the keypoints. (d) Extracted keypoints using temporal regularization, presented on the top of the input frame. (e) Result using temporal regularization, presented with the extracted keypoints. As can be seen, without temporal consistency the keypoints move much further within adjacent frames, resulting in temporal inconsistency.

produce a distribution map over the image pixels, denoted $\{H^\ell\}_{\ell=0}^{K-1}$. We then compute the expected values of these distributions to recover a pixel coordinate for each keypoint:

$$k^\ell = [u^\ell, v^\ell]^T = \sum_{u,v} [u \cdot H^\ell(u, v), v \cdot H^\ell(u, v)]^T \quad (8)$$

where we sum over all image pixels $\{u, v\}$. The keypoints are then projected to spatial confidence maps h^ℓ such that for any pixel coordinates in the image $p = (u, v)$, we have:

$$h^\ell(p) = \frac{1}{\alpha} \exp \frac{-|p - k^\ell|}{\sigma^2} \quad (9)$$

where α, σ are constant across all the experiments and k^ℓ are the extracted keypoints. We have used the constant values $\alpha = 1$ and $\sigma = 0.1$ for all experiments.

B.2 Network Architecture

For E we use the architecture based on U-Net [31], as proposed by Siarohin et al. [34]. For G_A, G_B, R_A, R_B we use the generator architecture proposed by Zhu et al. [49], which utilizes skip connections for better visual quality. The generator consists of 9 residual blocks, each contains convolution, ReLU, and Instance Normalization layers. The discriminator consists of 3 fully connected and Leaky ReLU layers, followed by a final sigmoid activation, similarly to Mokady et al. [25].

B.3 Training details

We use the Adam [20] optimizer with a learning rate of $1e^{-4}$ for both generators and discriminators. Training time is approximately 12 hours over a single NVIDIA GTX1080 GPU.

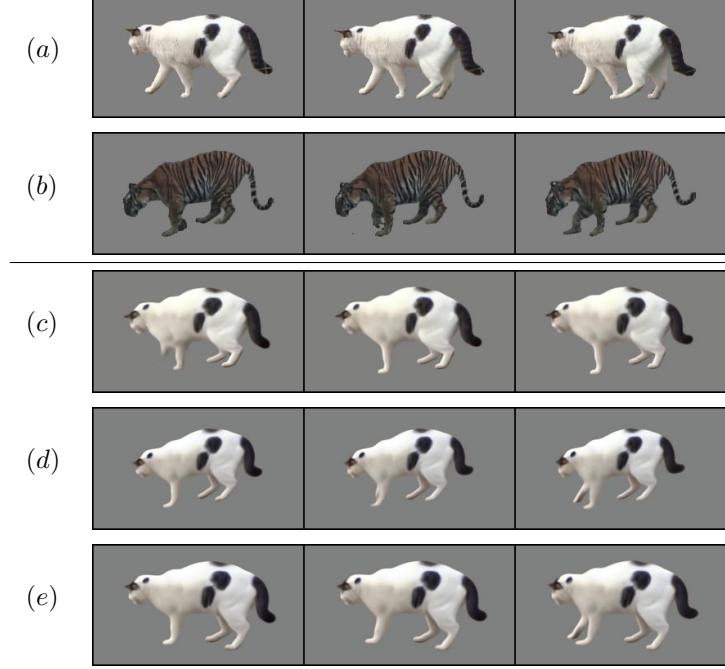


Figure 9: Ablation study for the affine invariant domain confusion component. (a), (b) depict the original cat and tiger. The (c) row presents the artifacts caused after omitting the learned affine transformation. Bottom rows depict the results for avoiding the learned transformation at inference (d) and applying the transformation at inference (e), after using the learned affine transformation at training.

For the first step optimization:

$$\mathcal{L}_G = \lambda_{\text{seg}} \mathcal{L}_{\text{seg}} + \lambda_{\text{DC}} \mathcal{L}_{\text{DC}}^A + \lambda_{\text{tmp}} \mathcal{L}_{\text{tmp}} + \lambda_{\text{eq}} \mathcal{L}_{\text{eq}} + \lambda_{\text{sep}} \mathcal{L}_{\text{sep}} + \lambda_{\text{sill}} \mathcal{L}_{\text{sill}}, \quad (10)$$

we use the following hyperparameters: $\lambda_{\text{seg}} = 50$, $\lambda_{\text{DC}} = 0.5$, $\lambda_{\text{tmp}} = 1.0$, $\lambda_{\text{eq}} = 1.0$, $\lambda_{\text{sep}} = 1.0$, $\lambda_{\text{sill}} = 0.5$, $\delta = 0.1$.

And for the second step:

$$\mathcal{L}_G = \mathcal{L}_{L1} + \lambda_{\text{LPIPS}} \mathcal{L}_{\text{LPIPS}}, \quad (11)$$

We use $\lambda_{\text{LPIPS}} = 2.0$. For all experiments we use between 10 to 14 keypoints, and train each step for approximately 45,000 iterations.

B.4 Segmentation

Even though our method requires a binary segmentation of the object (silhouette), various methods can be used to acquire it, allowing a wide variety of object types to be considered. For the Chihuahua in Fig. 14 and Fig. 23, we used an off-the-shelf pretrained saliency segmentation network of Qin et al. [30]. For YouTube-VOS dataset [43], we used manually annotated segmentations. GIFs and flowers have no background, and so the silhouette can be extracted using a simple threshold. For dancing videos, we used a pretrained network of Guler et al. [11] for human segmentation.

C Additional visual results

Additional GIF synchronization result is in Fig. 10. Comparisons to baselines over four-legged animals is given in Fig. 11 to Fig. 16. Additional results for our method is given in Fig. 17 to Fig. 23. Editing results are in Fig. 24. Both Fig. 25 and Fig. 26 present additional dancing results, while Fig. 27 presents additional flower results. Fig. 28 demonstrates that our method generates similar results for different random initializations.

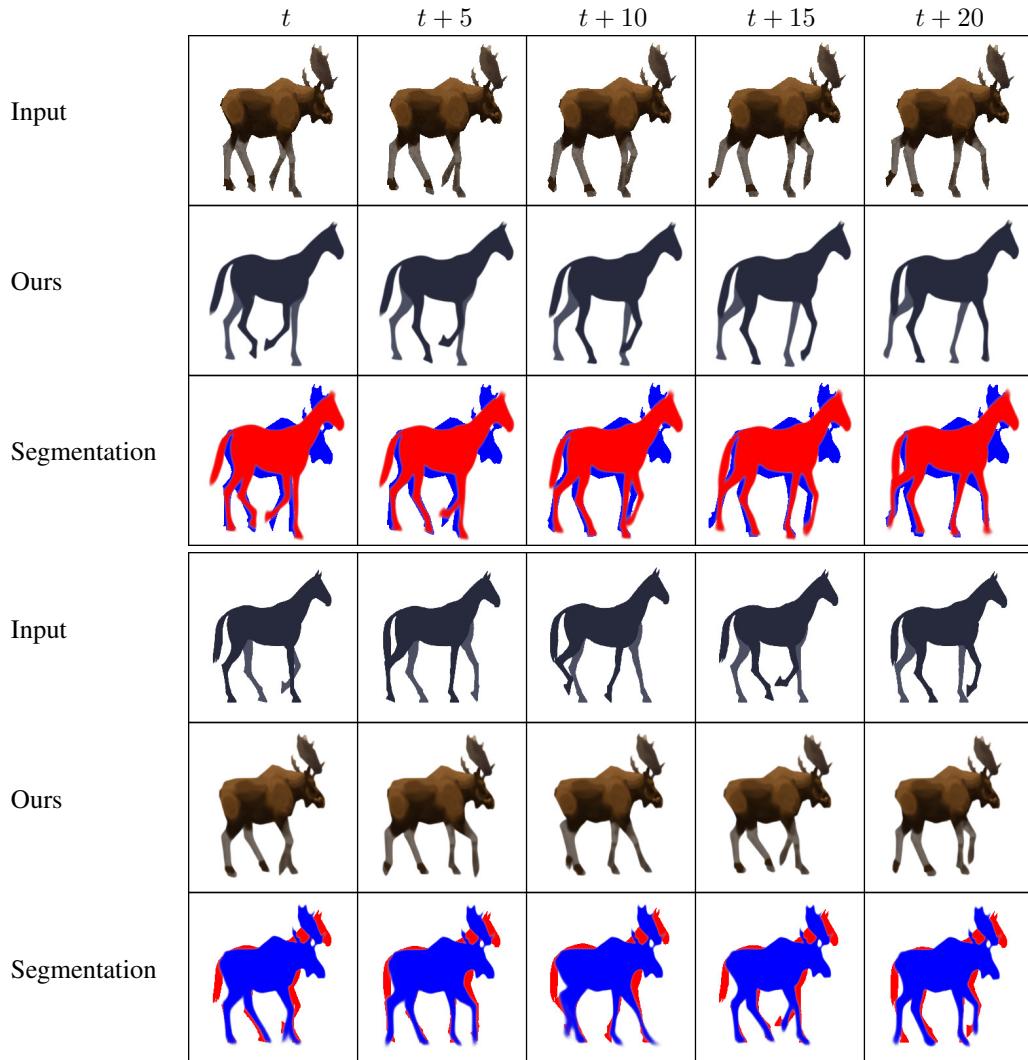


Figure 10: GIF Synchronization. To demonstrate alignment we show the segmentation of the elk (blue) and the horse (red) on top of each other.

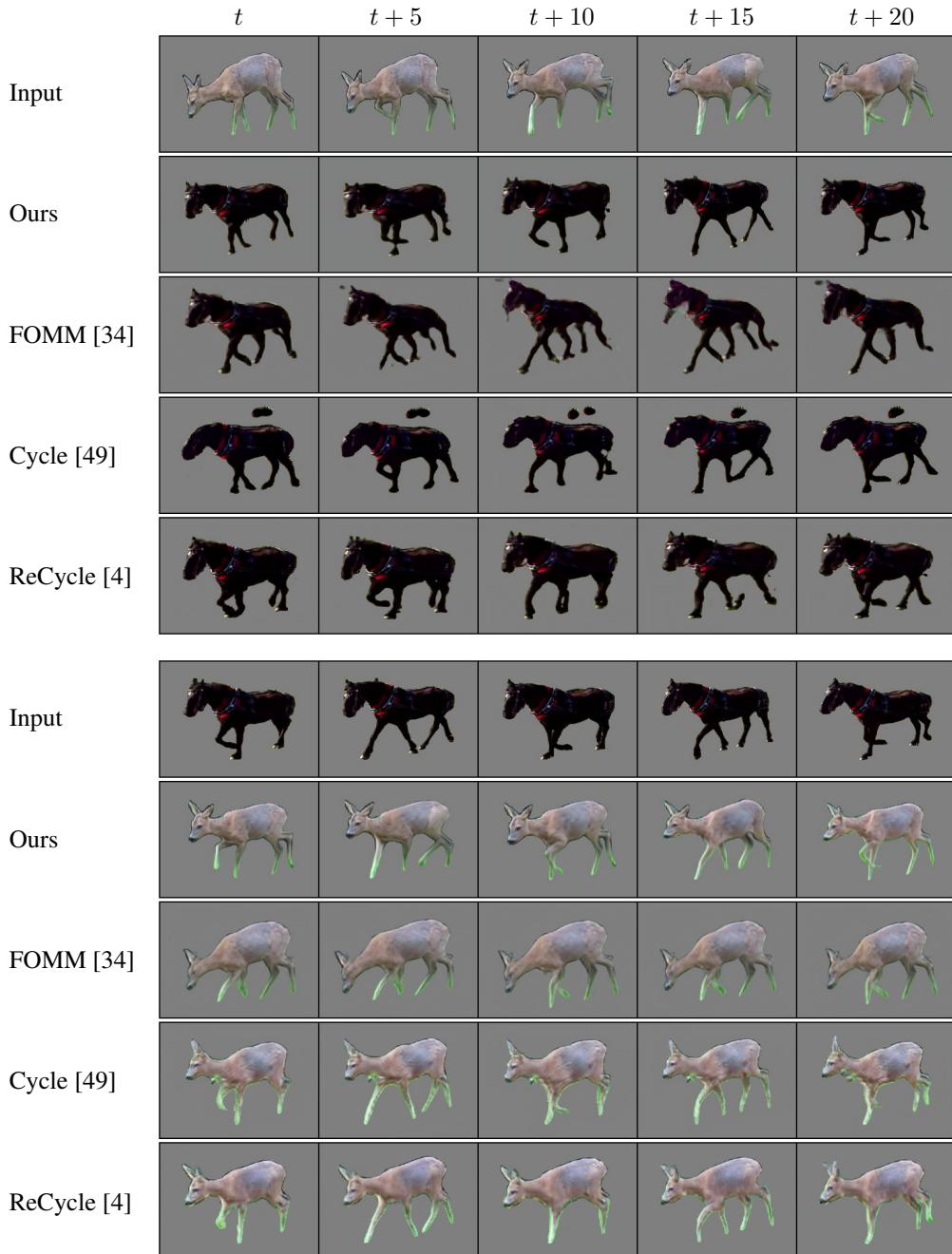


Figure 11: Comparison for horse/deer pair. As can be seen, our method successfully transfers motion while preserving the original style and appearance.

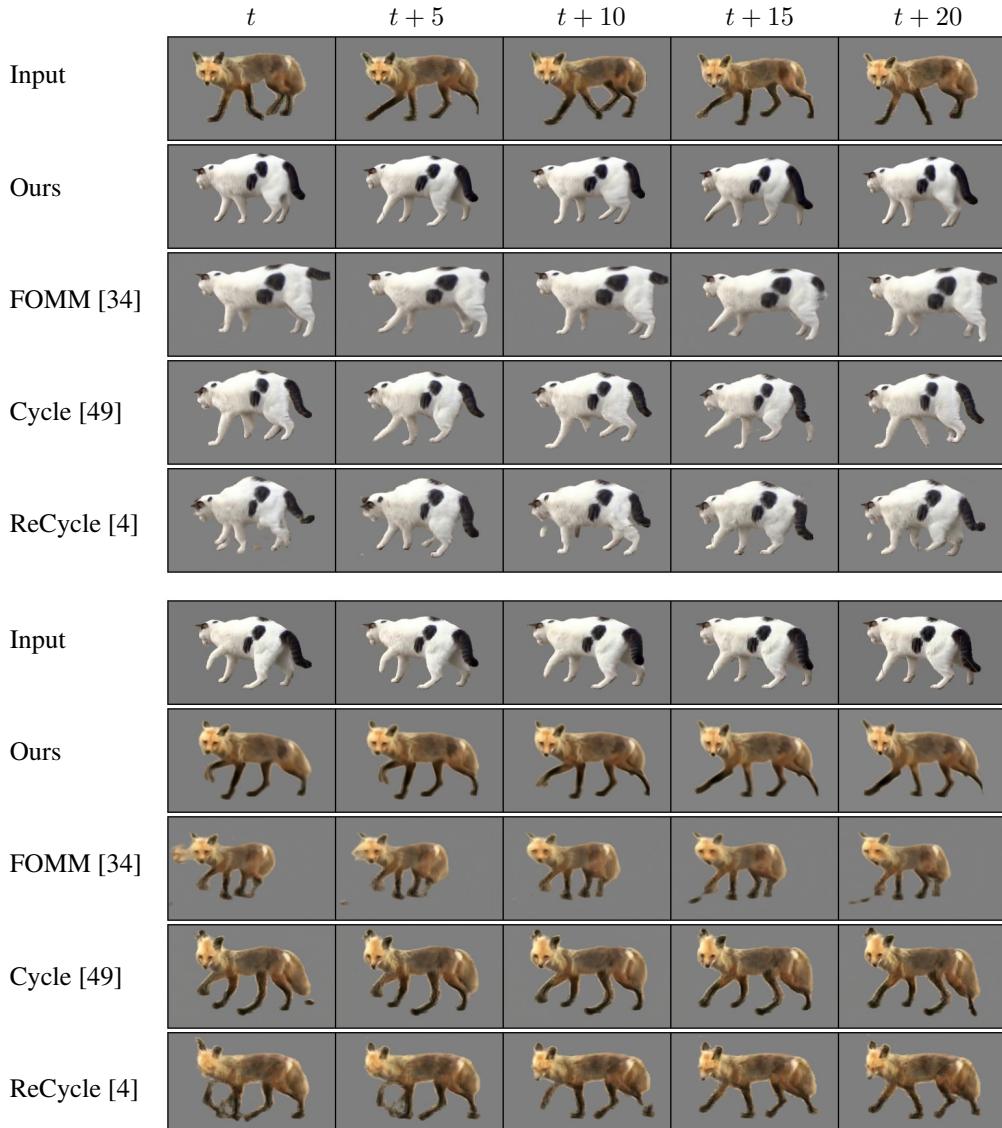


Figure 12: Comparison for cat/fox. As can be seen, our method successfully transfers motion while preserving the original style and appearance.

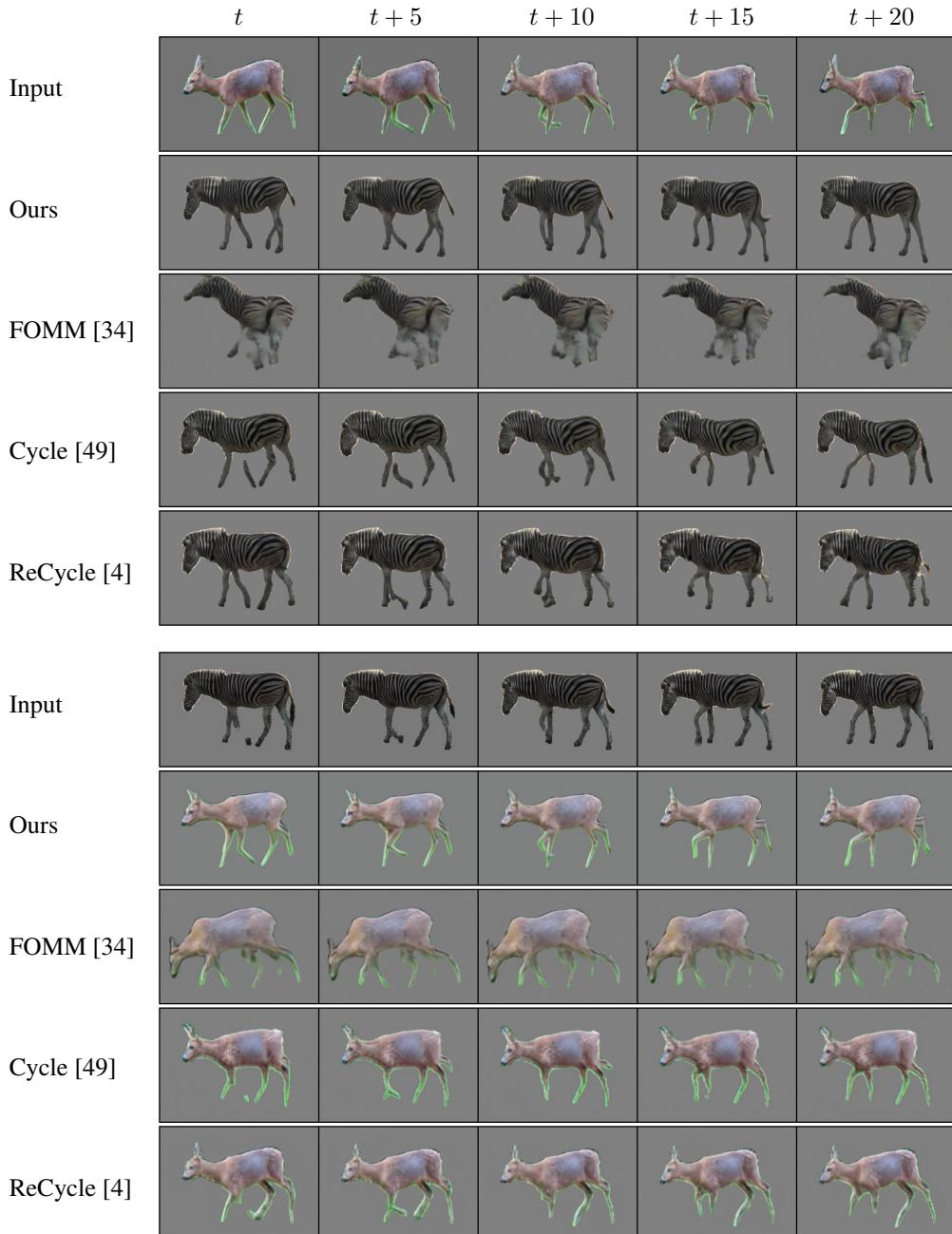


Figure 13: Comparison for zebra/deer. As can be seen, our method successfully transfers motion while preserving the original style and appearance.

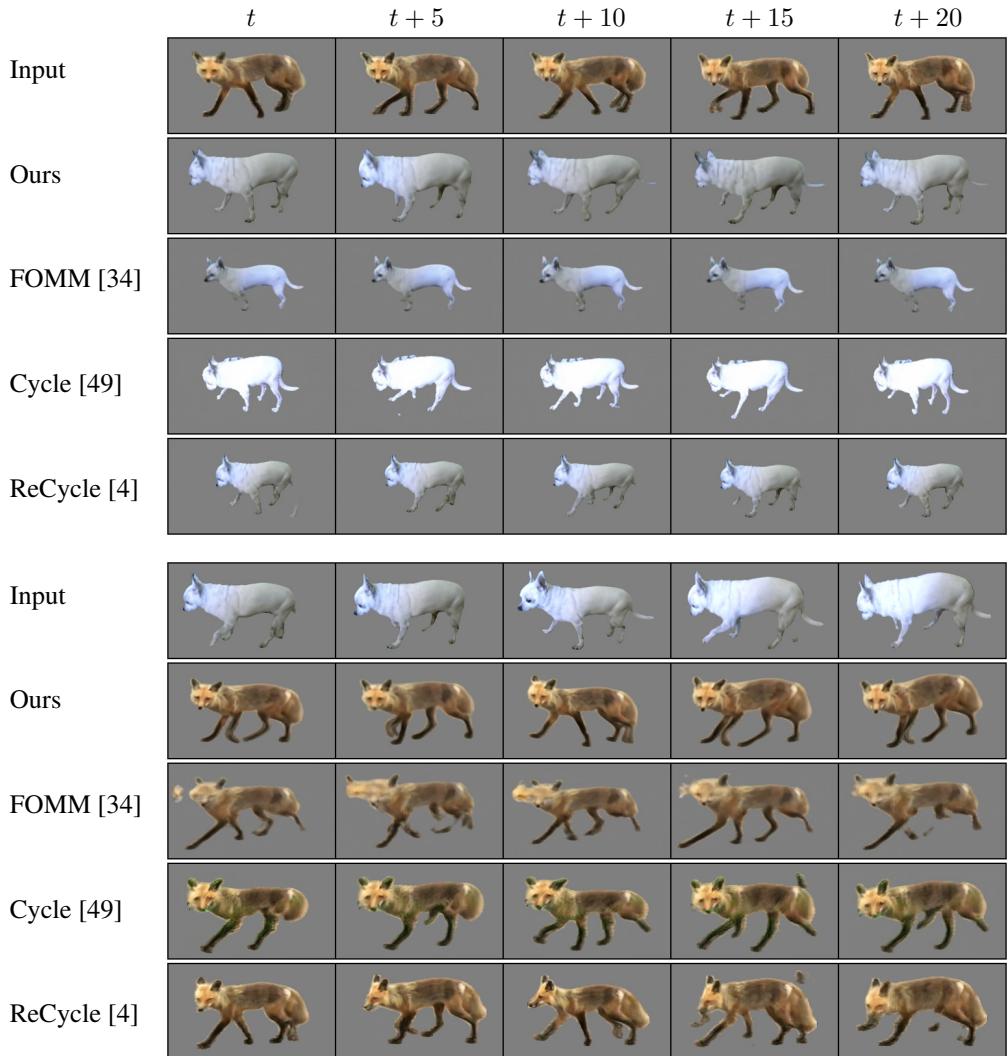


Figure 14: Comparison for chihuahua/fox. As can be seen, our method successfully transfers motion while preserving the original style and appearance.

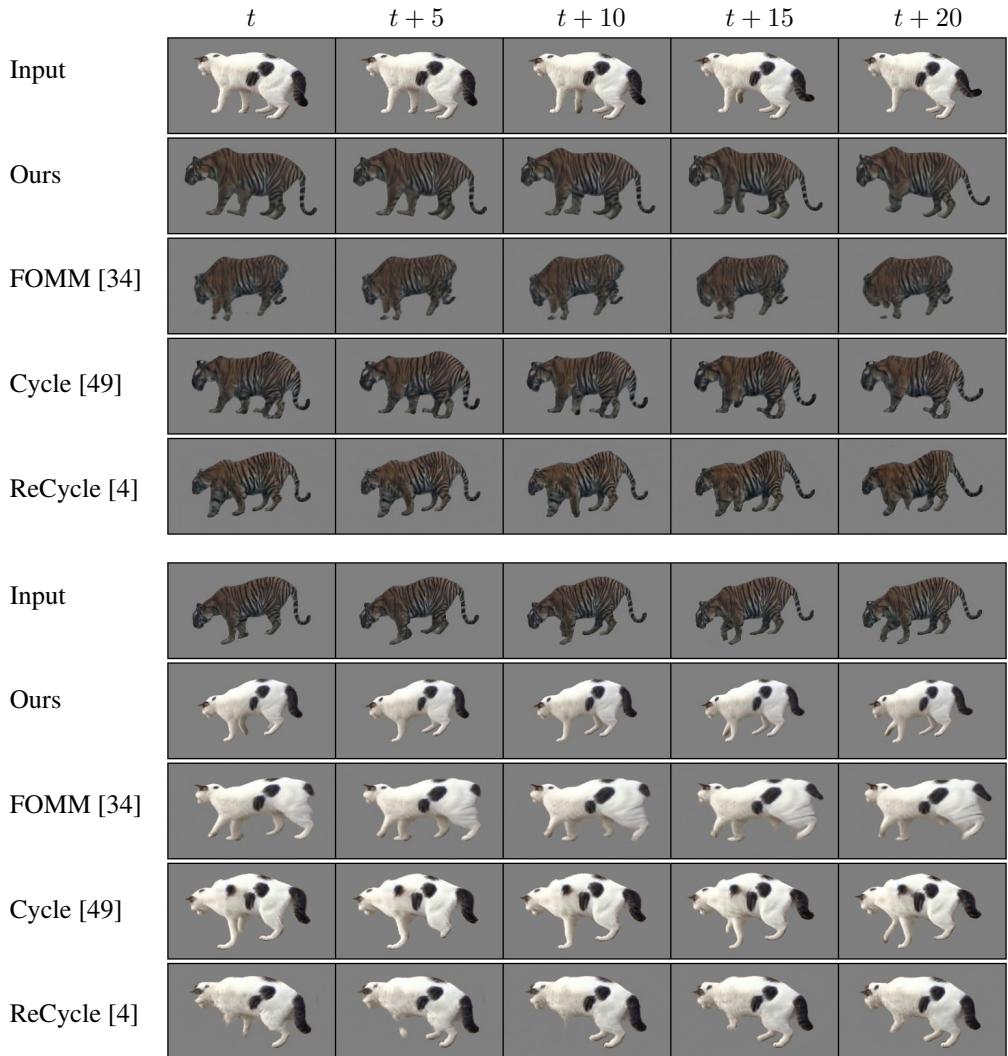


Figure 15: Comparison for cat/tiger. As can be seen, our method successfully transfers motion while preserving the original style and appearance.



Figure 16: Comparison for cow/fox. As can be seen, our method successfully transfers motion while preserving the original style and appearance.

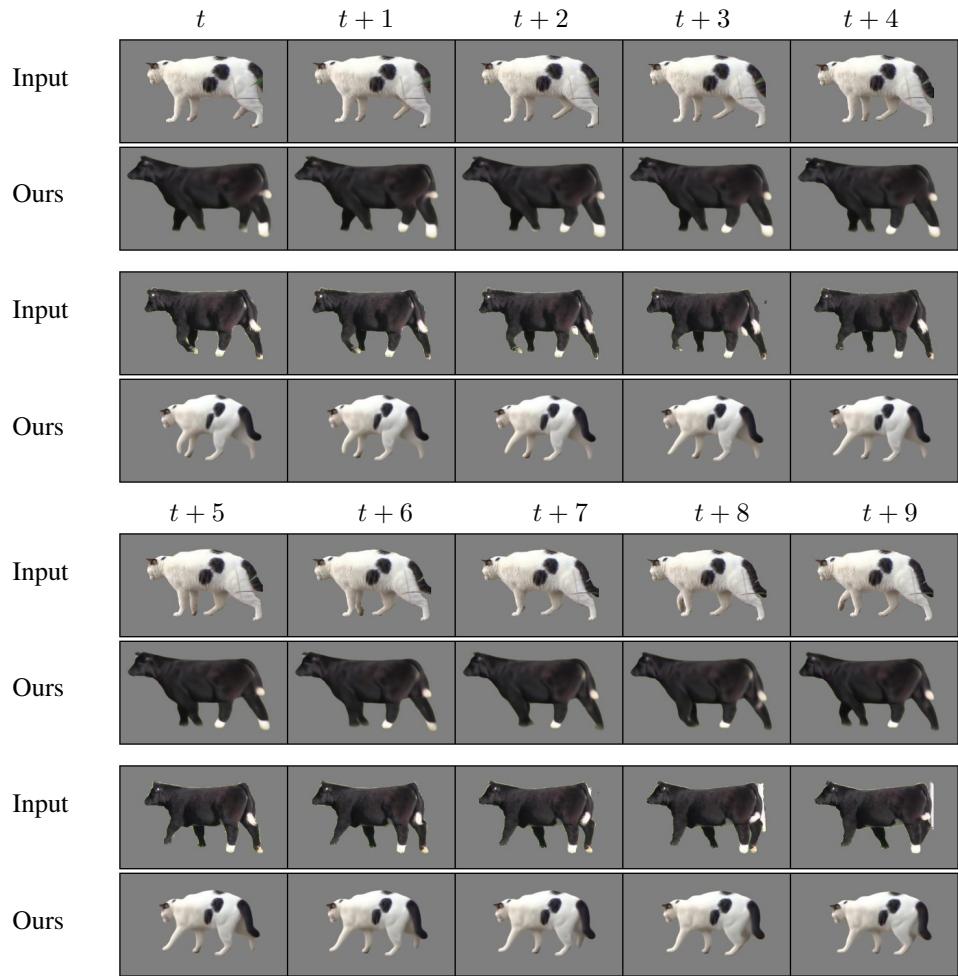


Figure 17: Additional results for our method.

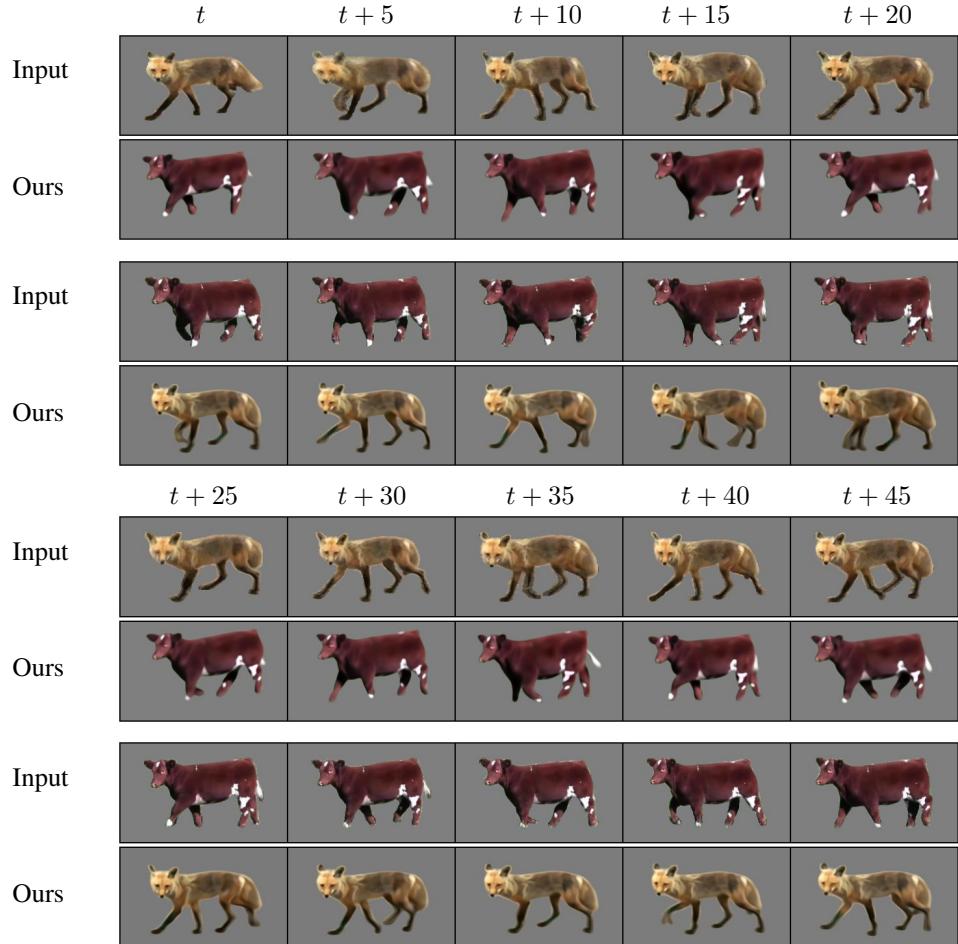


Figure 18: Additional results for our method.

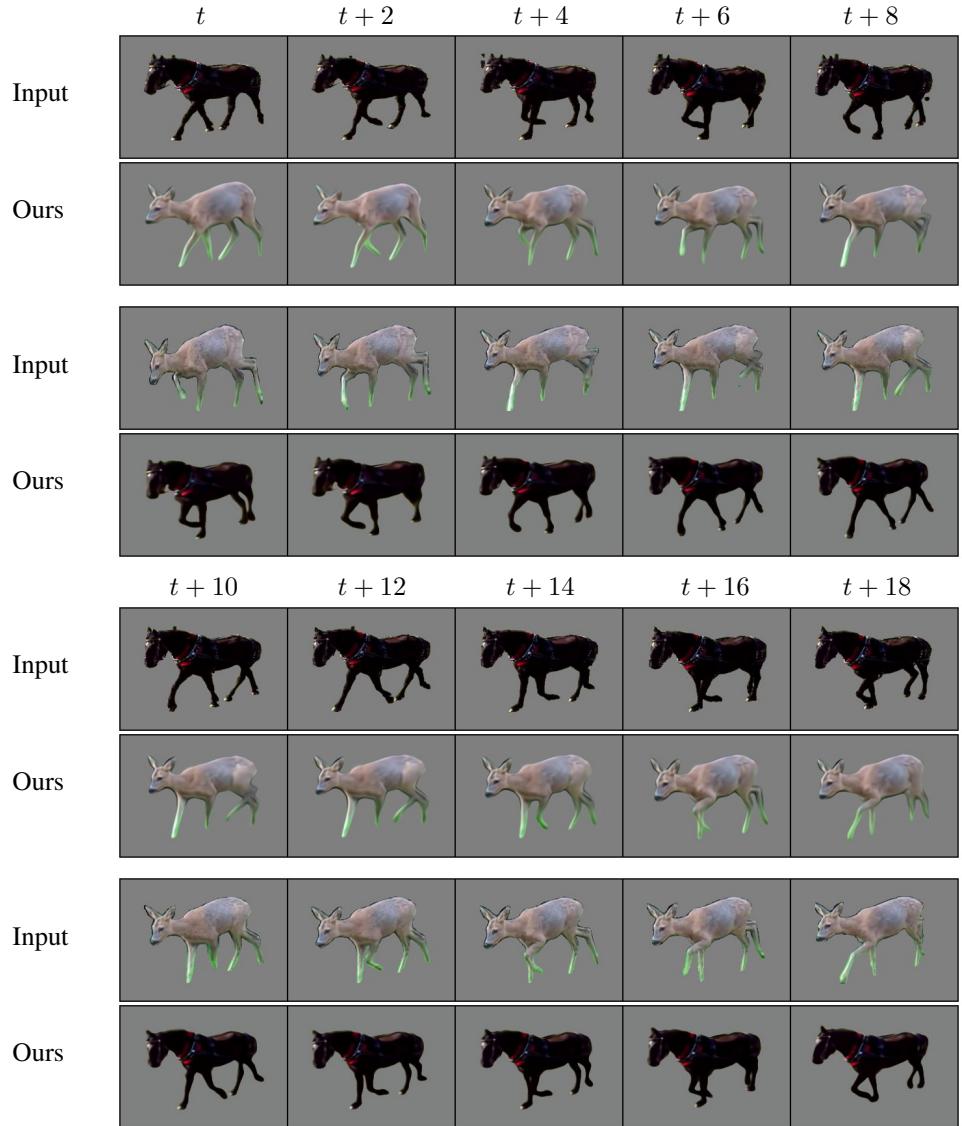


Figure 19: Additional results for our method.

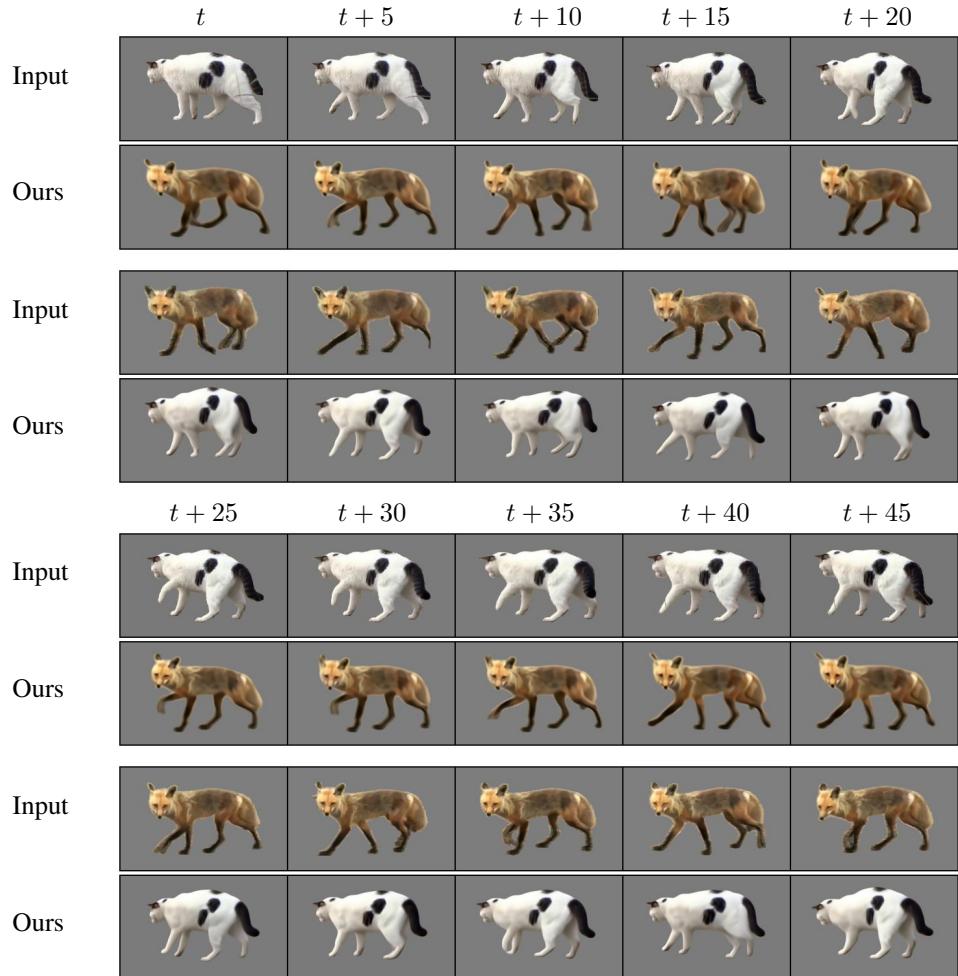


Figure 20: Additional results for our method.

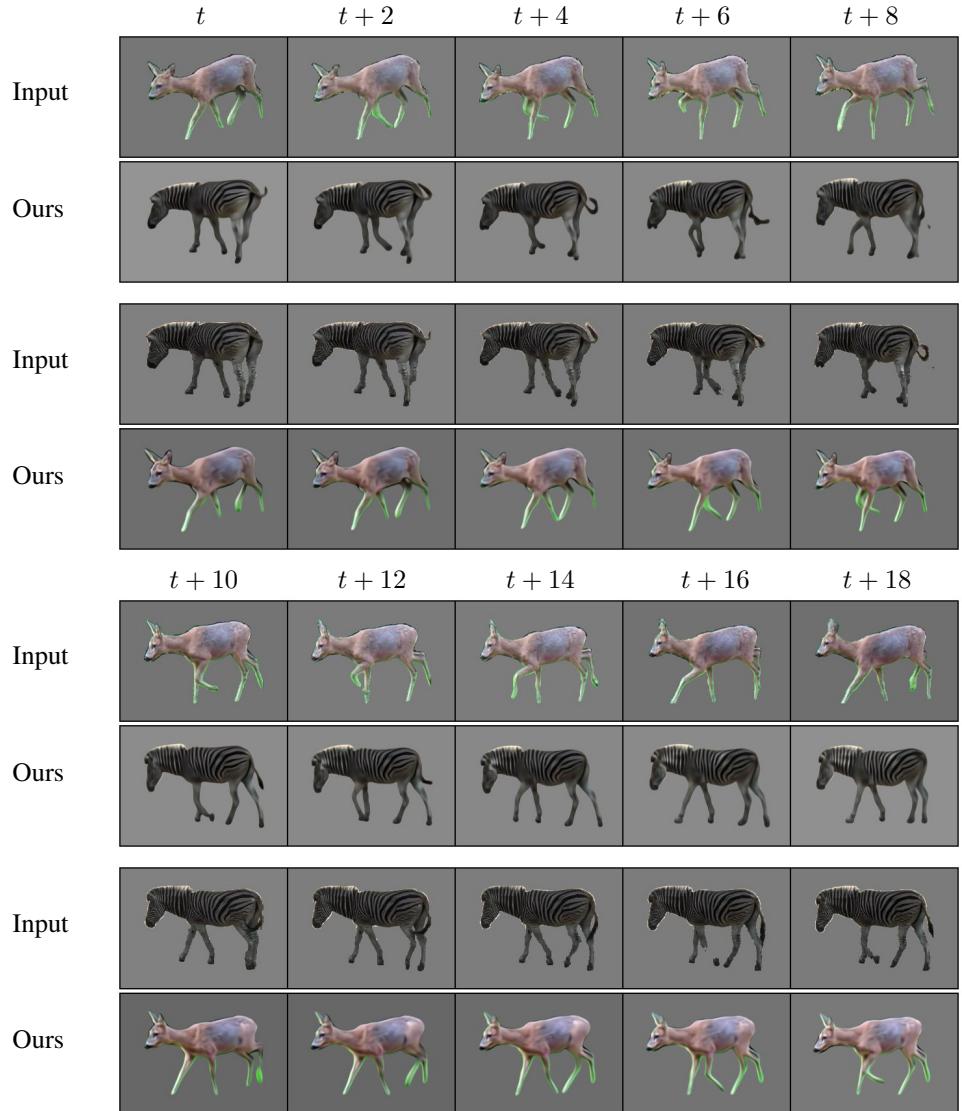


Figure 21: Additional results for our method.



Figure 22: Additional results for our method.

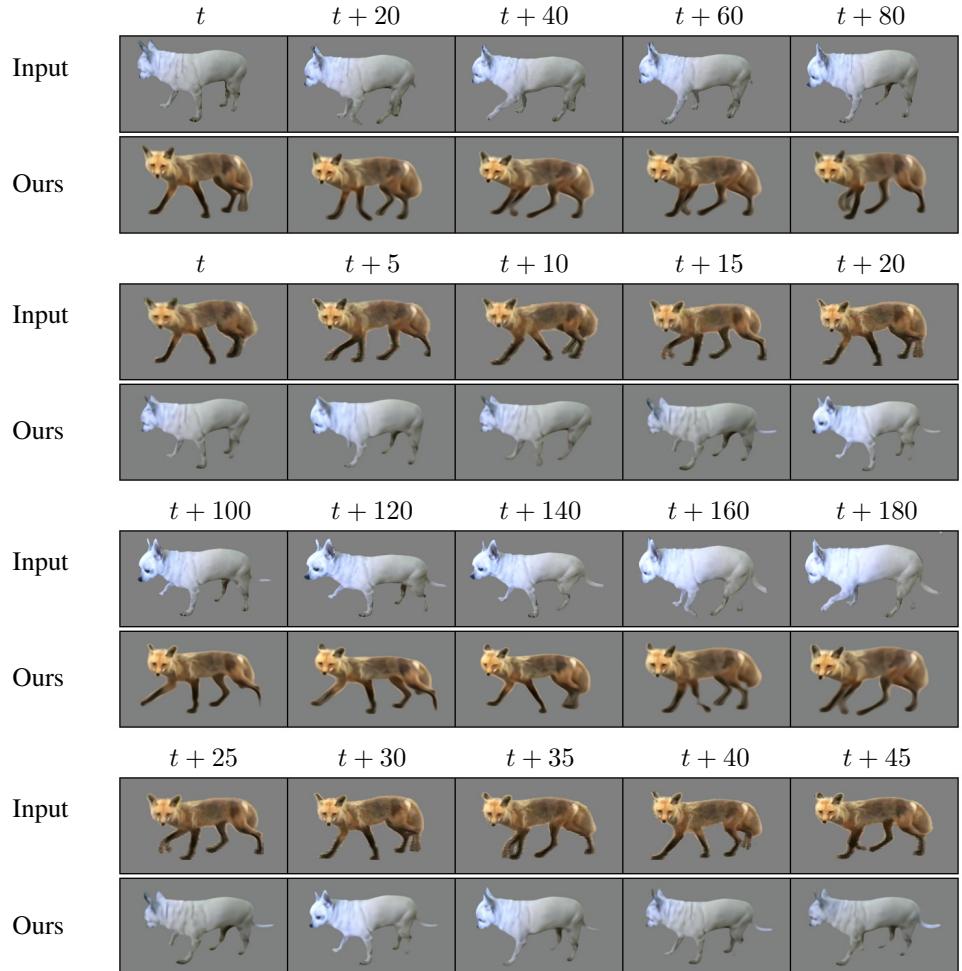


Figure 23: Additional results for our method. We show different time offsets, as the input videos has significantly different frame rates.

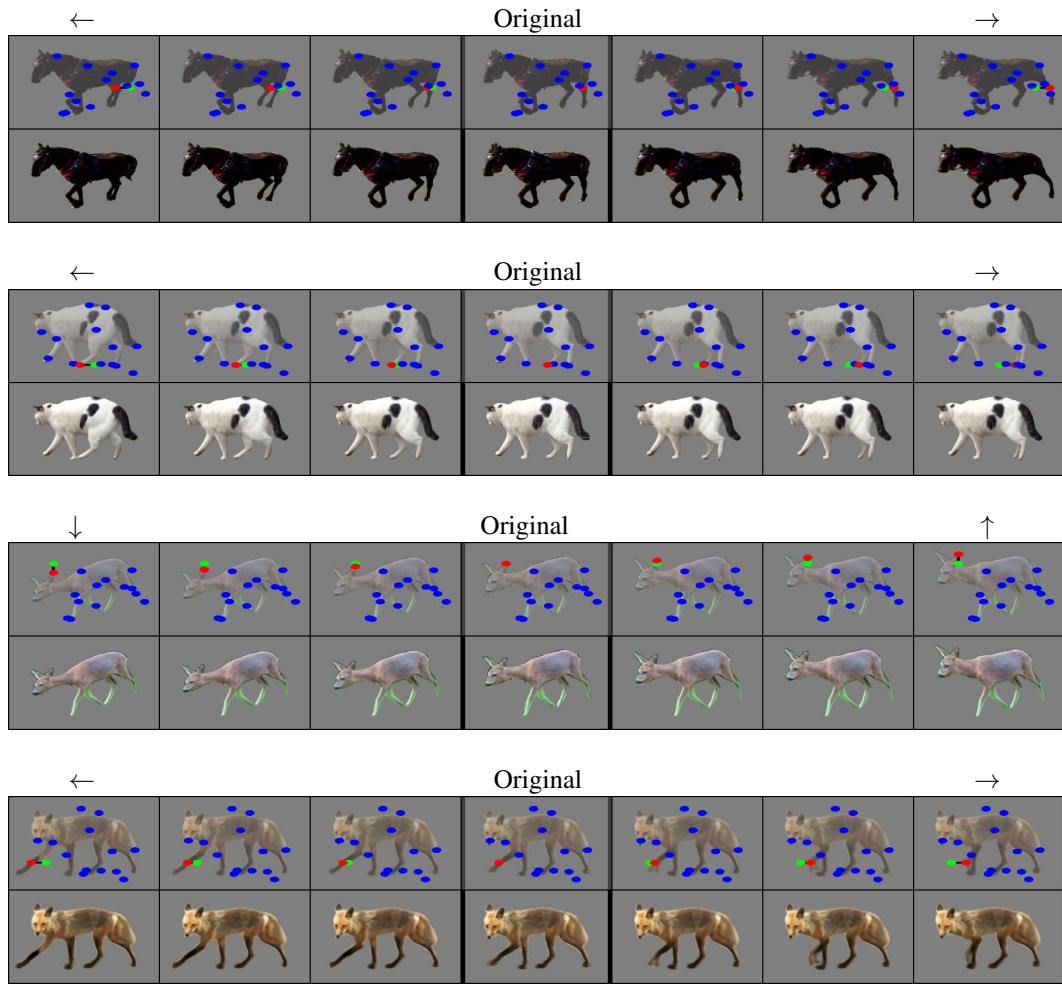


Figure 24: Additional editing results. Top to bottom: Moving the horse leg left/right, Moving the cat leg left/right, Moving the deer head up/down, Moving the fox leg left/right. As can be seen, moving the red keypoint induces a meaningful and realistic editing operation.

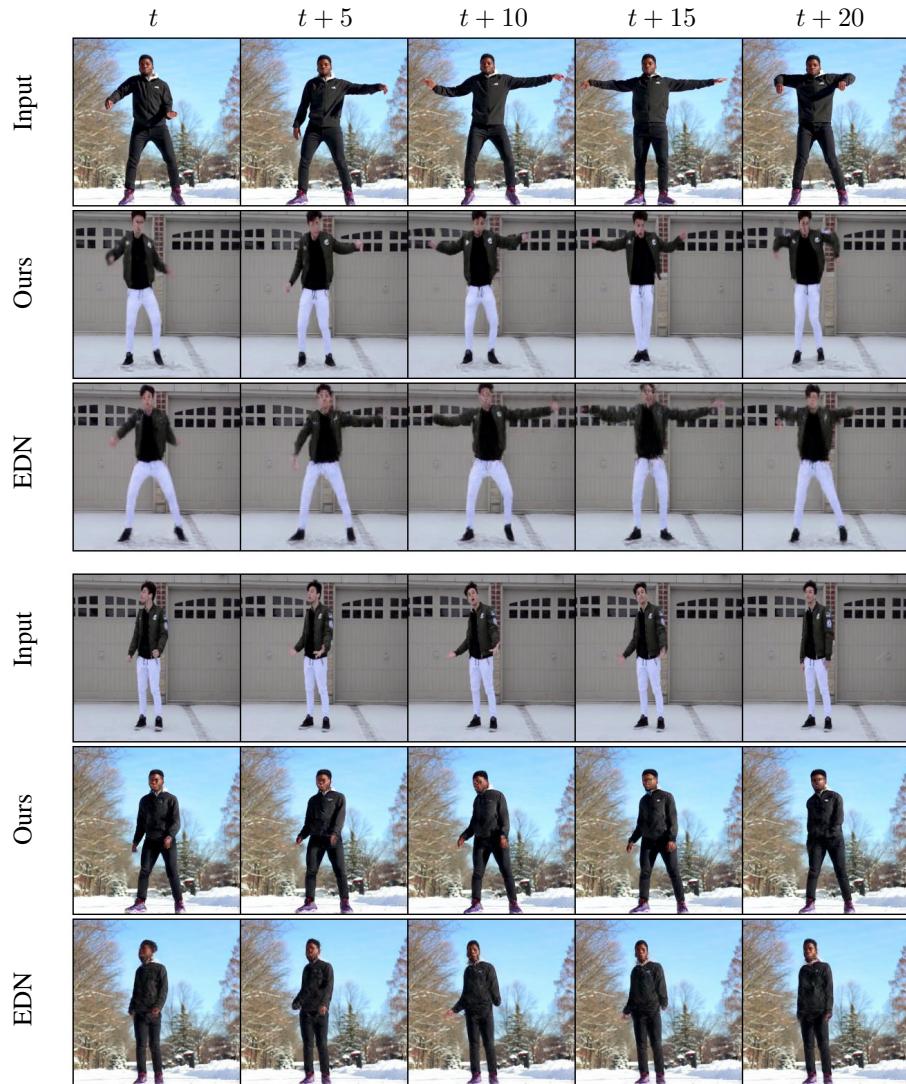


Figure 25: Dancing persons videos with comparison to EDN [7]. As can be seen, our results are comparable to EDN [7] which assumes a stronger supervision.

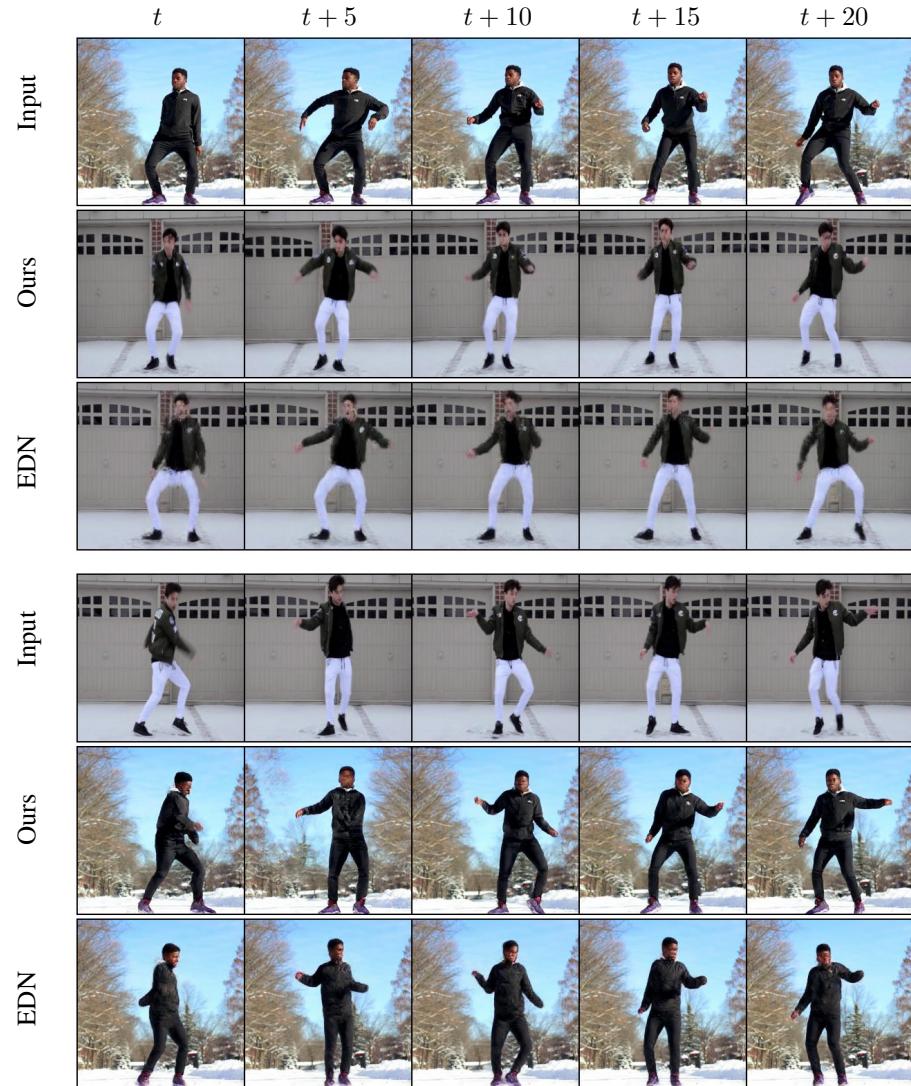


Figure 26: Dancing person videos with comparison to EDN [7]. As can be seen, our results are comparable to EDN [7] which assumes stronger supervision.

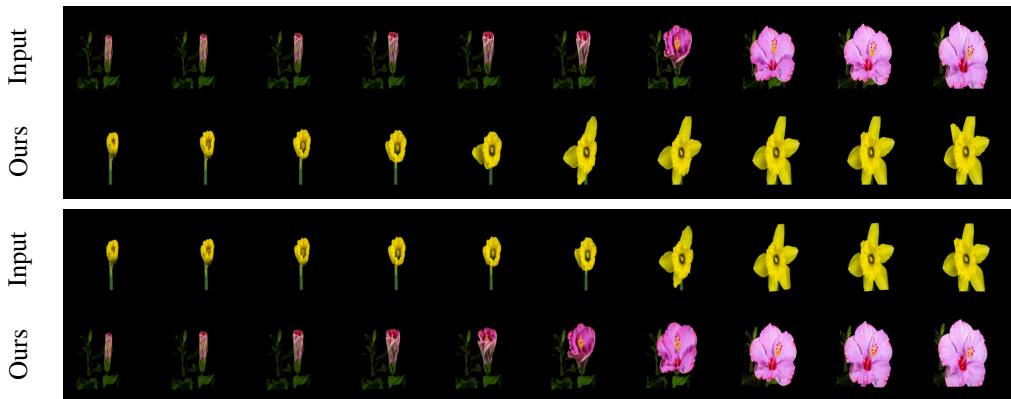


Figure 27: Blooming flowers results.

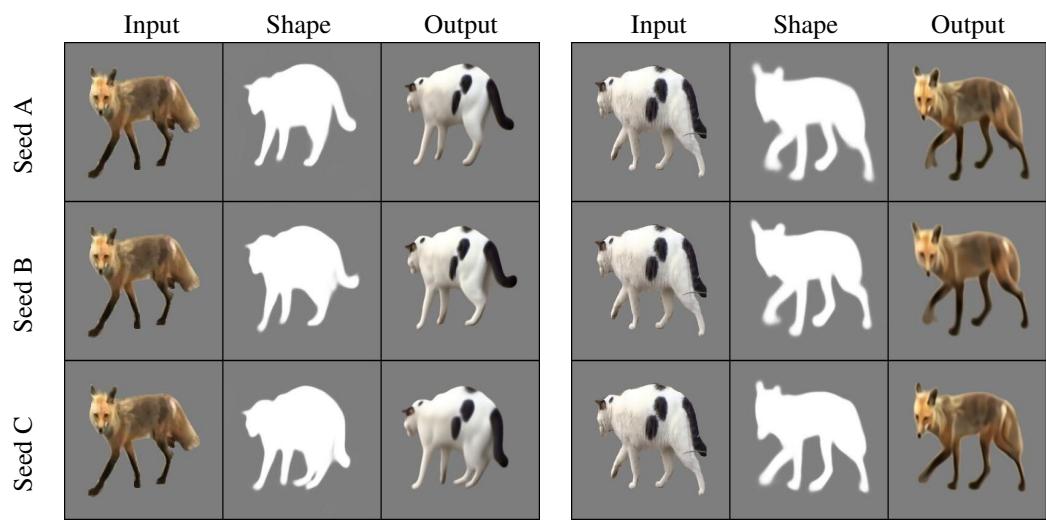


Figure 28: Results for different random seeds, as can be seen, our method generates similar results for the random initializations. Left to right: Input image, intermediate shape, and final result.