

Adversarial Framework for Unsupervised Learning of Motion Dynamics in Videos

C. Spampinato · S. Palazzo · P. D'Oro · D. Giordano · M. Shah

Abstract Human behavior understanding in videos is a complex, still unsolved problem and requires to accurately model motion at both the local (pixel-wise dense prediction) and global (aggregation of motion cues) levels. Current approaches based on supervised learning require large amounts of annotated data, whose scarce availability is one of the main limiting factors to the development of general solutions. Unsupervised learning can instead leverage the vast amount of videos available on the web and it is a promising solution for overcoming the existing limitations. In this paper, we propose an adversarial GAN-based framework that learns video representations and dynamics through a self-supervision mechanism in order to perform dense and global prediction in videos. Our approach synthesizes videos by 1) factorizing the process into the generation of static visual content and motion, 2) learning a suitable representation of a motion latent space in order to enforce spatio-temporal coherency of object trajectories, and 3) incorporating motion estimation and pixel-wise dense prediction into the training procedure. Self-supervision is enforced by using motion masks produced by the generator, as a co-product of its generation process, to supervise the discriminator network in performing dense prediction. Performance evaluation, carried out on standard benchmarks, shows that our approach is able to learn, in an unsupervised way, both local and global video dynamics. The learned representations, then, support the training of video object segmentation methods with sensibly less (about 50%) annotations, giving performance comparable to the state of the art. Further-

more, the proposed method achieves promising performance in generating realistic videos, outperforming state-of-the-art approaches especially on motion-related metrics.

1 Introduction

Learning motion dynamics plays an important role in video understanding, which fosters many applications, such as object tracking, video object segmentation, event detection and human behavior understanding. The latter is a particularly complex task, due to the variability of possible scenarios, conditions, actions/behaviors of interest, appearance of agents and to a generic ambiguity in how behaviors should be defined, categorized and represented. Behavior understanding is also a key component in the direction toward visual intelligence: the identification of what happens in a given environment necessarily requires the capability to decode actions and intentions from visual evidence.

If behavior understanding in computer vision is a fundamental component for scene understanding, pixel-wise dense prediction for video object segmentation is one of the founding stones for the whole process, as it isolates relevant regions in a scene from unnecessary background elements, thus serving both as a way to focus analysis on a subset of the input data, and to compute a preliminary representation suitable for further processing. Unfortunately, although video object segmentation has been studied for decades, it is far from solved, as current approaches are not yet able to generalize to the variety of unforeseeable conditions that are found in real-world applications. Additionally, learning long-term spatio-temporal features directly for dense prediction greatly depends on the availability of large annotated video object segmentation benchmarks (e.g., the popular DAVIS 2017 benchmark dataset contains only 150 short video clips, barely enough for training end-to-end deep models from scratch). The alternative approach is to

C. Spampinato, S. Palazzo, P. D'Oro, D. Giordano
PeRCeiVe Lab - University of Catania - Italy
Tel.: +39-0957387902
E-mail: {cspampin, palazzosim, dgiordan@dieei.unict.it}

M. Shah and C. Spampinato
Center for Research in Computer Vision - University of Central Florida
Tel.: +1 (407) 823-1119
E-mail: shah@crev.ucf.edu

avoid resorting to manual annotations by explicitly defining a “cost” or “energy” function based on *a priori* considerations on the motion patterns that characterize objects of interest in a video [34, 10]. However, these approaches do not seem to be on par with deep learning methods.

Recently, generative adversarial networks (GANs) [11] have become a successful trend in computer vision and machine learning, thanks to their results in advancing the state of the art on image generation to unprecedented levels of accuracy and realism [5, 38, 61, 1, 30, 40, 17]. The key idea of GANs is to have two models, a *generator* and a *discriminator*, compete with each other in a scenario where the discriminator learns to distinguish between real and fake samples, while the generator learns to produce more and more realistic images. As the models improve in parallel, they learn hierarchies of general feature representations that can be used for multiple tasks, e.g., image classification [38] and semantic segmentation [45]. These characteristics have demonstrated GANs’ usefulness in training or supporting the training of models from unlabeled data [42], rising as one of the most promising paradigms for unsupervised learning¹.

Given the success of GANs with images, the natural direction of research has been to attempt and extend their applicability to videos, both as a generative approach and as a way to disentangle video dynamics by learning features that leverage the vast amount of unlabeled data available on the web. For example, in a classification scenario such as *action recognition*, a simple way to employ unlabeled videos is to add an additional class for fake videos (i.e., produced by the generator) and have the discriminator both predict the realism of the input video and identify its class [31, 42]. However, naively extending image generation methods to videos by adding the temporal dimension to convolutional layers may be too simplistic, as it jointly attempts to handle both the spatial component of the video, which describes object and background appearance, and the temporal one, representing object motion and consistency across frames. Building on these considerations, recent generative efforts [53, 41] have attempted to factor the latent representation of each video frame into two components that model a time-independent background of the scene and the time-varying foreground elements. We argue that the main limitation of these methods is that both factors are learned by mapping a single point of a latent space (sampled as random noise) to a whole video. This, indeed, over-complicates the generation task as two videos depicting the same scene with different object trajectories or the same trajectory on differ-

ent scenes are represented as different points in the latent space, although they share a common factor (in the first case the background, in the second one object motion).

In this paper, we tackle both the problem of unsupervised learning for video object segmentation and that of video generation with disentangled background and foreground dynamics, combining both of them into an adversarial framework that guides the discriminator in performing video object segmentation through a self-supervision mechanism, by using ground-truth masks internally synthesized by the generator. In particular, our video generation approach employs two latent spaces (as shown in Fig. 1) to improve the video generation process: 1) a traditional random latent space to model the static visual content of the scene (background), and 2) a trajectory latent space suitable designed to ensure spatio-temporal consistency of generated foreground content. In particular, object motion dynamics are modeled as point trajectories in the second latent space, with each point representing the foreground content in a scene and each latent trajectory ensuring regularity and realism of the generated motion across frames. On top of the traditional adversarial framework, we extend the discriminator architecture in order to perform adversarial dense pixel-wise prediction in videos. In particular, besides the adversarial loss driving the generator/discriminator game, we add loss terms related to the discriminator’s estimation of optical flow (supervised by the output of a state-of-the-art algorithm) and segmentation masks (supervised by the foreground masks computed by the generator) from the generated videos. The three losses encourage the generator to produce realistic videos, while improving representation learning in the discriminator and unlocking the possibility to perform dense video predictions with no manual annotations. Experimentally, we verify that our video generation approach is able to effectively synthesize videos, outperforming existing solutions, especially in motion coherency metrics, thus suggesting that it indeed learns, in an unsupervised way, motion features. We further demonstrate that the features learned by the model’s discriminator can be used for effective unsupervised video object segmentation in different domains and allow for reducing significantly (about 50% less) the number of annotated frames required to achieve the same performance as through traditional supervision. Additionally, we find that the features learned through unsupervised learning encode general appearance and motion cues and can be also employed for global prediction tasks such as video action recognition.

To summarize, the main contributions of this paper are:

- We introduce a GAN-based video generation framework able to explicitly model object motion through learning a latent trajectory representation space that enforces spatio-temporal regularity of object motion in the gen-

¹ Note the distinction between *unsupervised learning* (the class of approaches that train machine learning models without annotations) and what is known in the video segmentation literature as *unsupervised segmentation* (the class of methods that perform segmentation in inference without additional input on object location other than the video frames).



Fig. 1: **Our adversarial video generation model:** we employ a scene latent space to generate background and a foreground latent space to generate object appearance and motion.

erated videos as well estimating motion cues from the generated content;

- We demonstrate that our framework provides a useful means for video object segmentation — known as being an annotation-hungry task — by both employing a trained generator to create synthetic foreground masks and directly integrating dense prediction into the adversarial training procedure;
- We verify that our approach is able to successfully learn video features that can be used in a variety of computer vision tasks.

2 Related Work

This paper mainly tackles the problem of unsupervised learning of motion dynamics for pixel-wise dense prediction in a video object segmentation scenario through an adversarial video generation framework. Thus, we first review the recent literature on video object segmentation methods and then focus on video generation approaches.

Video object segmentation is the task of predicting each video pixel either as foreground or background and consequently to segment objects preserving their boundaries. Recent video object segmentation methods can be classified as *unsupervised*, i.e., methods that perform segmentation in inference without additional input on object location other than the video frames, *semi-supervised*, i.e., methods that, instead, employ annotations in the first frame for inference, and *supervised*, i.e. methods that require annotations for every frame or user interaction. In this paper we propose a framework that enables to learn motion features through *unsupervised learning*, i.e., without using annotations at train-

ing time. The learned representations are then employed to train a method for unsupervised video object segmentation; thus, in this section, we will focus on this last class of approaches.

Many of these methods formulate the problem as a spatio-temporal tube (representing motion trajectories) classification task [34, 3, 9, 26, 23, 58]. The core idea is to track points or regions over consecutive frames in order to create coherent moving objects, by learning motion/appearance cues. Brox and Malik [3] propose a pairwise metric on trajectories across frames for clustering trajectories. Analogously, Fragkiadaki et al. [9] analyze embedding density variations between spatially contiguous trajectories to perform segmentation. Papazoglou and Ferrari [34], instead, model motion, appearance and spatial feature density forward and backward in time across frames to extract moving objects. Keuper et al. [23] also track points in consecutive frames and employ a multicut-based approach for trajectory clustering. Wang et al. [58] define spatio-temporal saliency, computed as geodesic distance of spatial edges and temporal motion boundaries, and employ such saliency as a prior for segmentation.

Recently, CNN-based methods for unsupervised video object segmentation have been proposed [35, 20, 4, 48], based on the concept of making the network implicitly learn motion/appearance cues for object trajectory modeling. Most of CNN-based methods use pre-trained segmentation models [35, 20, 4] or optical-flow models [48] to perform either semi-supervised or unsupervised segmentation. Such methods are, however, profoundly different from the more challenging case of learning to segment moving objects without utilizing, or reducing significantly the need of,

labeled data at training time. Thus, although our approach shares the strategy of these methods, i.e., making a CNN learn motion and appearance cues for segmentation, it significantly differs from them in that we learn motion cues in a completely unsupervised way during training.

To carry out unsupervised learning, we leverage the recent success of generative adversarial networks (GANs) [11] for high-quality image [5, 38, 61, 5, 1, 30, 40], video synthesis [53, 41, 50, 32] and prediction [21, 55], by devising a video generation model that performs, at the same time, pixel-wise dense prediction for video object segmentation. Existing adversarial deep image generation methods have attempted to increase the level of realism [5, 38, 61, 5] of the generated images, while providing solutions to cope with training stabilization issues [1, 30, 40]. Most of these methods use a single input latent space to describe all possible visual appearance variations, with some exceptions, such as Stacked GAN [17] that, instead, employs multiple latent spaces for different image scales. Works reported in [53, 41, 50] extend the GAN image generation framework to the video generation problem. In particular, the work in [53] replaces the GAN image generator with a spatio-temporal CNN-based network able to generate, from an input latent space, background and foreground separately, which are then merged into a video clip for the subsequent discriminator. Similarly, the work in [41] shares the same philosophy of [53] with the difference that the input latent space is mapped into a set of subspaces, where each one is used for the generation of a single video frame. However, simply extending the traditional GAN framework to videos fails, because of the time-varying pixel relations that are found in videos due to structured object motion. Although the above generation methods exploit video factorization into a stationary and a temporally varying part, deriving the two components from a single input latent space greatly complicates the task, and needs more training data given that each point in the input latent space corresponds to a complete scene with specific object motion. Furthermore, object motion is only loosely linked to the scene, e.g., the motion of a person walking on two different environments is more or less independent from the specific environment. Thus, the assumption that the two video components are highly inter-correlated (so as to derive both from a single point in the latent space) is too strong, and recently [50] performed video generation by disentangling the two video components into different latent spaces. While our approach is in the same spirit of [50], there are several crucial differences both in the problem formulation and in the motivation. In terms of motivation, our approach performs video generation with the main goal to learn motion dynamics in an unsupervised manner, and it is designed in order to implement a self-supervision mechanism for guiding the pixel-wise dense prediction process. In terms of problem formulation, we use two latent spaces as in

[50], but our foreground latent space is a multidimensional space from which we sample trajectories, and not a sequence of uncorrelated and isolated points, as done in [50]. These trajectories are fed to a recurrent layer that learns suitable temporal embeddings. Thus, we learn a latent representation for motion trajectories, which, as shown in the results, leads to a better spatio-temporal consistency than [50]. Along the line of using two separate latent spaces for background and foreground modeling, [33] propose a video generation approach that first generates optical flow and then converts it to foreground content for integration with a background model. Our approach differs from this work in the following ways:

- Although [33] employs two separate latent spaces for motion and content, single samples are drawn from the two for generating a video; instead, we learn a motion latent space, which more naturally maps to the spatio-temporal nature of motion, and encodes it as input for the generation process in the foreground stream.
- It employs optical flow provided by a state-of-the-art algorithm as a condition, as done in standard conditional GANs, in the hierarchical structure of the generator. We, instead, estimate optical flow through the discriminator and use it to supervise, in the form of “self-supervision”, video generation in order to directly encourage a better understanding of motion dynamics.
- Lastly, this work does not address the video object segmentation problem, which remains the main objective of our work.

Video GANs have been also adopted for the video prediction task [56, 52, 22], i.e., for conditioning the generation process of future frames given a set of input frames. While this is a different task than what we here propose, i.e., video generation for supporting spatio-temporal dense segmentation, the way we encode motion can resemble theirs with the key difference that they learn motion dynamics from real data and then draw samples from the learned data distribution; we, instead, learn a latent representation for motion trajectories by enforcing spatio-temporal consistency of generated content. Additionally, appearance and motion are also captured differently; for example, in [22], they are included by using explicit values as conditions for the generation of future frames, namely, the first frame of the video and some motion attributes; we, instead, model motion and appearance simply as samples drawn from latent spaces and provided as inputs to the generator, and not as quantities estimated by the discriminator.

As mentioned above, we adopt GANs mainly for unsupervised learning of object motion. GANs have been already employed for unsupervised domain adaption [2, 51], image-to-image translation [62, 60], for semi-supervised semantic image segmentation [45], as well as for unsupervised feature learning in the image domain [6]. In the video domain, GANs have been particularly useful for semi-supervised and

unsupervised video action recognition [53,41,50] or representation learning [28], given their innate ability to learn video dynamics while discriminating between real and fake videos. Unlike existing approaches, our video generation framework supports unsupervised pixel-wise dense prediction for video object segmentation, which is a more complex task that requires learning contextual relations between time-varying pixels. To the best of our knowledge, this is the first attempt to perform adversarial unsupervised video object segmentation, although some GAN-based approaches [25] perform dense prediction for the optical flow estimation problem. In particular, [25] proposes a conditional GAN taking an image pair as input and predicting the optical flow. Flow-warped error images both on predicted flow and on ground-truth flow are then computed and a discriminator is trained to distinguish between the real and the fake ones. The network is trained both on labeled and unlabeled data and the adversarial GAN loss is extended with the supervised end-point-error loss, computed on the labeled data. Differently from this work, our dense-prediction network uses only unlabeled data and extends the traditional adversarial loss by including the error made by the discriminator in estimating motion as well as in predicting segmentation maps.

3 Adversarial Framework for Video Generation and Unsupervised Motion Learning

Our adversarial framework for video generation and dense prediction — *VOS-GAN* — is based on a GAN framework and consists of the following two modules:

- a *generator*, implemented as a hybrid deep CNN-RNN, that receives two inputs: 1) a noise vector from a latent space that models scene background; 2) a sequence of vectors that model foreground motion as a trajectory in another latent space. The output of the generator is a video with its corresponding foreground mask.
- a *discriminator*, implemented as a deep CNN, that receives an input video and 1) predicts whether it is real or not; 2) performs pixel-wise dense prediction to generate an object segmentation map; 3) performs pixel-wise dense prediction to estimate the optical flow between video frames.

The traditional adversarial loss is extended by having the discriminator learn to compute motion-related dense predictions for the input video, thus forcing the generator to produce more realistic motion trajectories. Additionally, this formulation makes the discriminator suitable as a stand-alone model for object segmentation and optical flow estimation.

3.1 Generator Architecture

The architecture of the generator, inspired by the two-stream approach in [53], is shown in Fig. 2. Specifically, our generation approach factorizes the process into separate background and foreground generation, on the assumption that a scene is generally stationary and the presence of informative motion can be constrained only to a set of objects of interest in a semi-static environment. However, unlike [53] and similar to [50], we separate the latent spaces for scene and foreground generation, and explicitly represent the latter as a temporal quantity, thus enforcing a more natural correspondence between the latent input and the frame-by-frame motion output.

Hence, the generator receives two inputs: $z_C \in \mathcal{Z}_C = \mathbb{R}^d$ and $z_M = \{z_{M,i}\}_{i=1}^t$, with each $z_{M,i} \in \mathcal{Z}_M = \mathbb{R}^d$. A point z_C in the latent space \mathcal{Z}_C encodes the general scene to be applied to the output video, and is mainly responsible for driving the *background stream* of the model. This stream consists of a cascade of transposed convolutions, which gradually increase the spatial dimension of the input in order to obtain a full-scale background image $b(z_C)$, that is used for all frames in the generated video.

The set of $z_{M,i}$ points from the latent space, \mathcal{Z}_M , defines the objects motion to be applied in the video. The latent sequence is obtained by sampling the initial and final points and performing a spherical linear interpolation (SLERP [43]) to compute all intermediate vectors, such that the length of the sequence is equal to the length (in frames) of the generated video. Using an interpolation rather than sampling multiple random points enforces temporal coherency in the foreground latent space. The list of latent points is then encoded through a recurrent neural network (LSTM) in order to provide a single vector (i.e., the LSTM’s final state) summarizing a representation of the whole motion. After a cascade of spatio-temporal convolutions (i.e., with 3D kernels that also span the time dimension), motion features are conditioned on content by concatenation of intermediate features from the background that are replicated along the time dimension. Then, these activations are processed by a $1 \times 1 \times 1$ convolution and three 3D residual layers [15], with three residual blocks in each layer. After another convolutional layer, this *foreground stream* outputs a set of frames $f(z_C, z_M)$ with foreground content and masks defining motion pixel location $m(z_C, z_M)$.

The two streams are finally combined as

$$G(z_C, z_M) = m(z_C, z_M) \odot f(z_C, z_M) + (1 - m(z_C, z_M)) \odot b(z_C) \quad (1)$$

Foreground generation can be directly controlled acting on z_M . Indeed, varying z_M for a fixed value of z_C results in videos with the same background and different foreground appearance and motion.

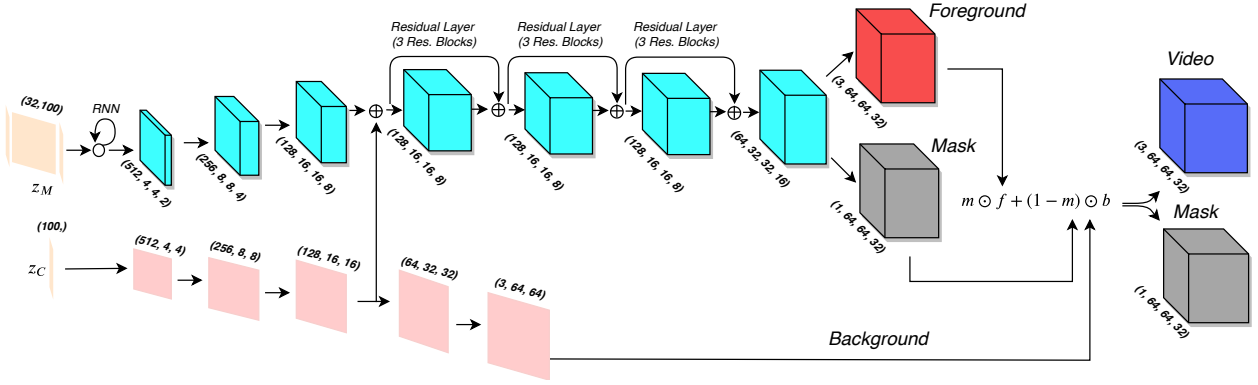


Fig. 2: **Generator architecture:** the *background stream* (bottom) uses a latent vector defining the general scene of the video, and produces a background image; the *foreground stream* (top) processes a sequence of latent vectors, obtained by spherically interpolating the start and end points, to generate frame-by-frame foreground appearance and motion masks. The foreground stream is conditioned on video content by concatenation (denoted with \oplus in the figure) of intermediate features produced by the background stream. Information about dimensions of intermediate outputs is given in the figure in the format (*channels, height, width, duration*) tuples.

3.2 Discriminator Architecture

The primary goal of the discriminator network is to distinguish between generated and real videos, in order to push the generator towards more realistic outputs. At the same time, we train the discriminator to perform dense pixel-wise predictions of foreground masks and optical flow. These two additional outputs have a twofold objective: 1) they force the discriminator to learn motion-related features, rather than (for example) learn to identify the visual features of objects that are more likely to be part of the foreground (e.g., people, animals, vehicles); 2) they enable the discriminator to perform additional tasks from unlabeled data.

Fig. 3 shows the architecture of the discriminator. The input to the model is a video clip (either real or produced by the generator), that goes first through a series of convolutional and residual layers, encoding the video dynamics into a more compact representation, which in turn is provided as input to two separate streams: 1) a *discrimination stream*, which applies 3D convolutions to the intermediate representation and then makes a prediction on whether the input video is real or fake; 2) a *motion stream*, feeding the intermediate representation to a cascade of 3D transposed convolutional and residual layers, which fork at the final layer and return the frame-by-frame foreground segmentation maps (each as a 2D binary map) and optical flow estimations (each as a two-channel 2D map) for the input video.

The discrimination path of the model (i.e., the initial shared convolutional layers and the discrimination stream) follows a standard architecture for video discrimination [53], while the motion path, based on transposed convolutions, decodes the video representation in order to provide the two types of dense predictions [27]. Formally, we de-

fine the outputs returned by the discriminator as $D_{adv}(x)$, $D_{seg}(x)$ and $D_{opt}(x)$, which are, respectively, the probability that the provided input is real, the foreground segmentation maps and the optical flow estimated for the video; input x may be either a real video or the output of the generator.

3.3 Learning Procedure

We jointly train the generator and the discriminator in a GAN framework, with the former trying to maximize the probability that the discriminator predicts fake outputs as real, and the latter trying to minimize the same probability. Additionally, when training the discriminator, we also include loss terms related to the accuracy of the estimated foreground segmentation and optical flow.

The main problem in computing these additional losses is that, while optical flow ground truth for fake videos can be easily obtained by assuming the output of a state-of-the-art algorithm to be sufficiently accurate, there are no video segmentation approaches that provide the needed accuracy. To solve this problem, we propose what is — to the best of our knowledge — the first approach for video object segmentation trained in an unsupervised manner: since the architecture of the generator internally computes the foreground masks of the generated videos, we use those to supervise the prediction of the masks computed by the discriminator. Of course, this kind of self-referential ground-truth will not be very meaningful at first, but will become more and more akin to real ground-truth from real videos as the generator improves.

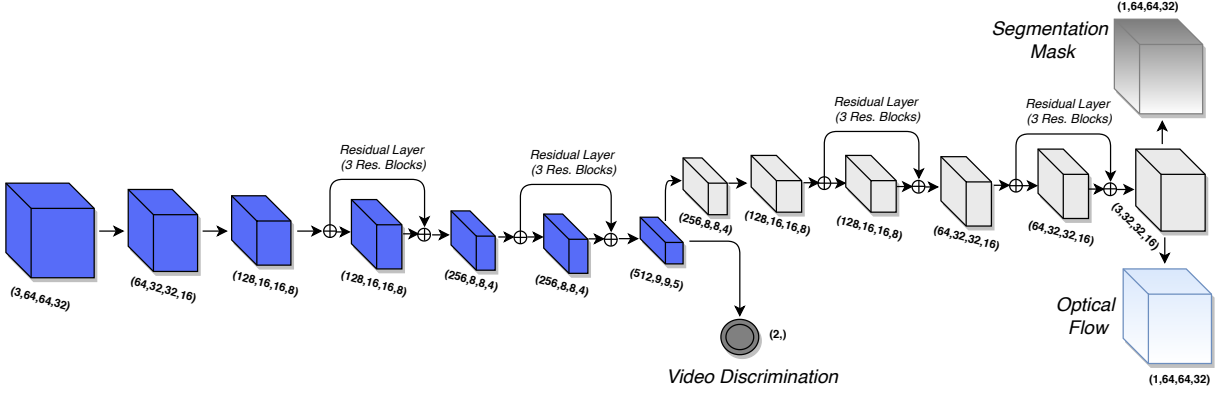


Fig. 3: **Discriminator architecture:** the *motion stream* (top) predicts foreground map and optical flow of the input video; the *discrimination stream* (bottom) outputs a single value, used for adversarial training, predicting whether the input video is real or fake.

The discriminator loss is then defined as follows (for the sake of compactness, we will define $z = (z_C, z_M)$):

$$\begin{aligned} \mathcal{L}_D = & -\mathbb{E}_{x \sim p_{\text{real}}} [\log D_{\text{adv}}(x)] \\ & -\mathbb{E}_{z \sim p_z} [\log (1 - D_{\text{adv}}(G(z)))] + \\ & +\mathbb{E}_{z \sim p_z} [\text{NLL}_{2D}(D_{\text{seg}}(G(z)), m(z))] + \\ & +\alpha \left(\mathbb{E}_{z \sim p_z} [\|D_{\text{opt}}(G(z)) - \text{OF}(G(z))\|^2] \right) + \\ & +\alpha \left(\mathbb{E}_{x \sim p_{\text{real}}} [\|D_{\text{opt}}(x) - \text{OF}(x)\|^2] \right) \end{aligned} \quad (2)$$

In the equation above, the first two lines encode the adversarial loss, which pushes the discriminator to return high likelihood scores for real videos and low ones for the generated videos. The third line encodes the loss on foreground segmentation, and it computes the average pixel-wise negative log-likelihood of the predicted segmentation map, using the generator’s foreground mask $m(z)$ as source for correct labels (in our notation, $\text{NLL}_{2D}(\hat{y}, y)$ is the negative log-likelihood of predicted label map \hat{y} given correct label map y). The last two lines encode the loss on optical flow estimation, as the squared L_2 norm between the predicted optical flow and the one calculated on the input video using the $\text{OF}(\cdot)$ function, implemented as per [8]. It should be noted that the non-adversarial term related to object segmentation is only computed on the generated videos (hence in a fully-unsupervised manner) for which foreground masks are provided by the generator in Fig. 2, since segmentation ground-truth may not be available for real videos. Optical flow is, instead, computed on both generated and real videos, and it serves to provide supervision to the discriminator (especially to the motion stream shown in Fig. 3) in learning motion cues from real videos. We additionally introduce an α

term to control the influence of optical flow and segmentation estimations over the traditional discriminator loss. α is initially set to 1 and then increased by a step s_α at each epoch since a specific epoch E_α . The role of α is specifically to stabilize the GAN training procedure by first letting it learn the general appearance of the scene and, as training goes on, focusing more on learning motion cues.

The generator loss is, more traditionally, defined as:

$$\mathcal{L}_G = -\mathbb{E}_{z \sim p_z} [\log D_{\text{adv}}(G(z))] \quad (3)$$

In this case, the generator tries to push the discriminator to increase the likelihood of its output being real.

During training, we follow the common approach for GAN training, by sampling real videos (from an existing dataset) and generated videos (from the generator) and alternately optimizing the discriminator and the generator.

4 Performance Analysis

Our system is specifically designed for supporting unsupervised dense and global prediction in videos, but it requires first to train the GAN-based generation model. For this reason, we initially report the video generation performance followed by the performance obtained for video object segmentation (pixel-wise dense prediction) and video action recognition (global prediction).

4.1 Datasets and Metrics

4.1.1 Video Generation

For ease of comparison with publicly-available implementations of state-of-the-art methods, we train our video generation model on two different datasets: “Golf course” videos

(over 600,000 video clips, each 32 frames long) from the dataset proposed in [53] and also employed in [41], and the Weizmann Action database [12] (93 videos of people performing 9 actions), employed in [50]. To have data in the same format, we pre-process the Weizmann Action dataset by splitting the original videos into 32-frame partially-overlapping (i.e., shifted by 5 frames) sequences, resulting in 683 such video clips.

For testing video generation capabilities, we compute qualitative and quantitative results. For the former, we carry out a user study aiming at assessing how the generated videos are perceived by humans in comparison to other GAN-based generative methods, and measure the preference score in percentage. To assess video generation performance quantitatively, we evaluate the appearance and motion of the generated videos, using the following metrics:

- **Foreground Content Distance (FCD)**. This score assesses foreground appearance consistency by computing the average L_2 distance between visual features of foreground objects in two consecutive frames. Feature vectors are extracted from a fully-connected layer of a pre-trained Inception network [47], whose input is the bounding box containing the foreground region, defined as the discriminator’s segmentation output.
- **Fréchet Inception Distance (FID)** [16] suitably adapted to videos as in [57]. FID is a widely adopted metric for implicit generative models, as it has been demonstrated to correlate well with visual quality of generated content. We employ the variant for videos proposed in [57], that projects generated videos into a feature space corresponding to a specific layer of a pre-trained video recognition CNN. The embedding layer is then considered as a continuous multivariate Gaussian, and consequently mean and covariance are computed for both generated data and real data. The Fréchet distance between these two Gaussians (i.e., Wasserstein-2 distance) is then used to quantify the quality of generated samples. As pre-trained CNN model for feature extraction we use ResNeXt [59, 14].
- **Inception score (IS)** [42] is the most adopted quality score in GAN literature. In our case, we compute the Inception score by sampling a random frame from each video in a pool of generated ones.

4.1.2 Video Object Segmentation

The capabilities of the motion stream of our discriminator network (see Fig. 3) for pixel-wise dense prediction are tested on several benchmarks for video object segmentation: DAVIS 2016 and DAVIS 2017 [36] and SegTrack-v2 [49]. Each dataset provides accurate pixel-level annotations for all video frames. The employed datasets show diverse features, useful to test the performance of video object segmen-

tation methods in different scenarios. In particular, DAVIS 2017 contains 150 video sequences (50 of which constitute the DAVIS 2016 dataset), and includes challenging examples with occlusion, motion blur and appearance changes. SegTrack-v2 is a dataset containing 14 videos showing camera motion, slow object motion, object/background similarity, non-rigid deformations and articulated objects.

We employ standard metrics [36] for measuring video object segmentation performance to ease comparison to state-of-the-art methods, namely: a) *region similarity* \mathcal{J} , computed as pixel-wise intersection over union of the estimated segmentation and the ground-truth mask; b) *contour accuracy* \mathcal{F} , defined as the F_1 -measure between the contour points of the estimated segmentation mask and the ground-truth mask. For each of the two metrics, we compute the mean value as:

$$\mathcal{M}_{\mathcal{C}}(R) = \frac{1}{|R|} \sum_{S_i \in R} \bar{\mathcal{C}}(S_i), \quad (4)$$

where $\bar{\mathcal{C}}(S_i)$ is the average of measure \mathcal{C} (either \mathcal{J} or \mathcal{F}) on S_i and R is the set of video sequences S . Also, for comparison with state of the art methods we compute recall \mathcal{O} and decay \mathcal{D} of the above metrics. The former quantifies the portion of sequences on which a segmentation method scores higher than a threshold, and is defined as:

$$\mathcal{O}_{\mathcal{C}}(R) = \frac{1}{|R|} \sum_{S_i \in R} \mathbb{I}[\bar{\mathcal{C}}(S_i) > \tau], \quad (5)$$

where $\mathbb{I}(p)$ is 1 if p is true and 0 otherwise, and $\tau = 0.5$ in our experiments. Decay measures the performance loss (or gain) over time:

$$\mathcal{D}_{\mathcal{C}}(R) = \frac{1}{|R|} \sum_{Q_i \in R} \bar{\mathcal{C}}(Q_i^1) - \bar{\mathcal{C}}(Q_i^4), \quad (6)$$

with $Q_i = \{Q_i^1, Q_i^2, Q_i^3, Q_i^4\}$ being a partition of S_i in quartiles.

The results are computed on test or validation sets where available; otherwise, we split the videos into training and test sets with proportions of 70% and 30%. In particular, ablation studies and analysis of the different architectural settings are done on DAVIS 2017 (because of its larger size). We use DAVIS 2016 and SegTrack-v2 for showing generalization capabilities of our approach and for comparison to state-of-the-art methods. All available videos are divided into 32-frame shots and downsampled to fit the input size allowed by the video segmentation network (i.e., the discriminator architecture in Fig. 3), i.e., 64×64 , while output segmentation maps are rescaled (through bi-linear interpolation) to ground-truth size for metrics computation.

Accurate evaluation of optical flow estimation is not performed, since our model is designed primarily for performing prediction by self-supervision (i.e., adversarial generation of foreground masks), while optical flow, provided by

a state-of-the-art method, is used only to guide the discriminator towards learning motion features from real videos. It should be noted that we did not use any deep learning-based optical flow, e.g., [18], but, instead, employed the traditional approach in [8] as it exploits physical properties of object motion in a purely unsupervised way. This avoids to include any form of “human-supervision” in the segmentation pipeline, making the proposed approach fully unsupervised.

4.1.3 Action Recognition

The capabilities of our model (namely, the discriminator stream in Fig. 3) in learning global motion cues are tested in a video action recognition scenario. In particular, we evaluate its accuracy in classifying actions in videos on two benchmarks, UCF101 [44] (13,320 videos from 101 action categories) and the Weizmann Action database [12]. We use average classification accuracy (the ratio of correctly-classified samples over the total number of test samples) as metric. In the evaluation on Weizmann Action dataset, due to the small number of video sequences, we perform 10-fold cross validation to average accuracy scores. Performance on video action recognition serves also to provide an additional metrics for quantitative evaluation for video generation.

4.2 Training Settings

The architectures of the generator and discriminator networks in terms of kernel sizes, padding, stride, activation functions and use of batch normalization [19] are given, respectively, in Tables 1 and 2.

In the video generation and segmentation experiments, we performed gradient-descent using ADAM, with an initial learning rate of 0.0002, $\beta_1 = 0.5$, $\beta_2 = 0.999$ and batch size of 16. For video generator training we used $\alpha = 1$, $s_\alpha = 0.2$ and $E_\alpha = 2$ (values set empirically). We trained the video generation models for 25 epochs and the video segmentation ones for 200 epochs. For video action recognition, we used SGD with momentum and dampening of 0.9, weight decay of 0.001 and learning rate of 0.1, reduced by a factor of 10 when no improvement in validation accuracy occurred for 10 epochs. Batch size was 128 and the number of epochs was 130.

4.3 Video Generation

4.3.1 Qualitative evaluation

To evaluate qualitatively our video generation approach, we used Amazon Mechanical Turk (MTurk) in order to measure how generated videos are perceived by humans.

Our generated videos are compared to those synthesized by VGAN [53], TGAN [41], MoCoGAN [50].

On MTurk, each job is created by randomly choosing two of the models under comparison and generating a 16-video batch from each of them; workers are then asked to choose which of the two sets of videos look more realistic. All workers have to provide answers for all the generated batches. We consider only the answers by workers with a lifetime Human Intelligent Task rate over than 90%. The achieved results are reported in Tab. 3 and show how our method generates more visually-plausible “golf” videos compared to VGAN, TGAN, and MoCoGAN. MoCoGAN, instead, outperforms our approach on the Weizmann Action dataset. It should be noted that MoCoGAN, differently from our approach, employs two discriminators — one for single frames and one for the whole video — and given the low scene variability in the Weizmann Action dataset, the frame generator is able to produce high-quality frames. In our approach, instead, background and foreground are integrated in videos and the discriminator is mainly trained (see Eq. 2) to capture motion features rather than visual appearance. A direct consequence from our learning schema is that VOS-GAN requires many samples to generate good resolution videos, and this explains why performance is better in the “Golf course” dataset (about 600,000 videos) than in Weizmann Action dataset (less than 100 videos). However, the capability of our approach to learn better motion cues than MoCoGAN is demonstrated by the results obtained on quantitative generation evaluation (see Sect. 4.3.2) and in action recognition task (see Sect. 4.5).

Samples of generated videos for VGAN, TGAN, MoCoGAN and our method are shown in Fig. 4, while comparisons with MoCoGAN on the Weizmann Action Dataset are shown in Fig. 5.

4.3.2 Quantitative evaluation

Quantitative evaluation of video generation performance is carried out by measuring FCD, FID and IS scores on 20 sets of 50,000 videos generated by the compared models trained on “Golf course” of [53], and on 20 sets of 500 videos generated on the Weizmann Action Dataset. Results are computed in terms of mean and standard deviation of each metrics over the sets of generated samples.

Firstly, we perform an ablation study on “Golf course” to understand how our GAN design choices affect the quality of the generated videos, by evaluating the above-mentioned metrics when each proposed term is included in the model. In particular, we computed the performance of VOS-GAN excluding all components (i.e., loss on segmentation and optical flow and replacing the RNN-SLERP based trajectory latent space modeling with a random latent space) — VOS-GAN (baseline) — and when gradually including in-

Bkg stream	Kernel	Stride	Padding	Activation	BatchNorm	Output shape
z_C	—	—	—	—	—	100x1x1
ConvTran2D	4x4	1x1	—	LReLU($\alpha = 0.2$)	Yes	512x4x4
ConvTran2D	4x4	2x2	1x1	LReLU($\alpha = 0.2$)	Yes	256x8x8
ConvTran2D	4x4	2x2	1x1	LReLU($\alpha = 0.2$)	Yes	128x16x16
ConvTran2D	4x4	2x2	1x1	LReLU($\alpha = 0.2$)	Yes	64x32x32
ConvTran2D	4x4	2x2	1x1	Tanh	No	3x64x64
Motion features	Kernel Size	Stride	Padding	Activation	BatchNorm	Output shape
RNN(z_M)	—	—	—	—	No	100x1x1
ConvTran3D	2x4x4	1x1x1	—	LReLU($\alpha = 0.2$)	Yes	512x4x4x2
ConvTran3D	3x3x3	3x3x3	1x2x2	LReLU($\alpha = 0.2$)	Yes	256x8x8x4
ConvTran3D	4x4x4	2x2x2	1x1x1	LReLU($\alpha = 0.2$)	Yes	128x16x16x8
Fg features	Kernel Size	Stride	Padding	Activation	BatchNorm	Output shape
Conv3D	1x1x1	1x1x1	—	—	—	128x16x16x8
Residual Layer						
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	Yes	128x16x16x8
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	Yes	128x16x16x8
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	Yes	128x16x16x8
Residual Layer						
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	Yes	128x16x16x8
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	Yes	128x16x16x8
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	Yes	128x16x16x8
Residual Layer						
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	No	128x16x16x8
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	No	128x16x16x8
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	No	128x16x16x8
ConvTran3D	4x4x4	2x2x2	1x1x1	LReLU($\alpha = 0.2$)	Yes	64x32x32x16
Fg raw	Kernel Size	Stride	Padding	Activation	BatchNorm	Output shape
ConvTran3D	4x4x4	2x2x2	1x1x1	Tanh	No	3x64x64x32
Fg mask	Kernel Size	Stride	Padding	Activation	BatchNorm	Output shape
ConvTran3D	4x4x4	2x2x2	1x1x1	Sigmoid	No	1x64x64x32

Table 1: Architecture of the generator. *Bkg stream* contains the layers included in the background stream of the model, that returns $b(z_C)$ (see Eq. 1 in the paper). *Motion features* lists the layers in the initial shared part of the foreground stream. *Fg features* considers the layers used to process concatenation over the channel dimension of the output of *motion features* with the output from the third layer of the *background stream*. By ”3D Residual Block” we simply denote standard shape-preserving residual blocks using 3D convolution. The output of *Fg raw* is $f(z_C, z_M)$ and the output of *Fg mask* is $m(z_C, z_M)$. *LReLU* stands for Leaky ReLU, while layers marked with *ConvTran* execute transposed convolution over their input.

dividual components. The results of our ablation study are given in Tab. 4. Both loss terms (on segmentation and optical flow) contribute to the model’s accuracy: optical flow has the largest impact, likely because foreground regions usually correspond to clusters of oriented vectors in the optical flow, hence learning to compute it accurately is also informative from the segmentation perspective. Both SLERP and LSTM also contribute significantly to generating visually-plausible videos by modeling better motion as demonstrated by the increase in FID and FCD metrics.

Tab. 5 shows the comparison with existing methods on the “Golf course” and on the Weizmann Action datasets. The results show that VOS-GAN outperforms VGAN, TGAN and

MoCOGAN on the three metrics on “Golf course”, while MoCoGAN is better than the proposed approach, on two out of the three adopted metrics, on the Weizmann Action Dataset. The reasons behind the different behavior of VOS-GAN and MoCoGAN on the two datasets are similar to those mentioned above in the qualitative analysis: 1) The Weizmann action dataset is characterized by few videos with constrained motion and background, thus the variability among frames is low and MoCoGAN’s image generator is able to model scene appearance with good quality. On the contrary, our approach is able to learn motion (as demonstrated by the results in action recognition given below), but it does not receive enough training data (given the size of the

Shared	Kernel Size	Stride	Padding	Activation	BatchNorm	Out shape
Input	—	—	—	—		3x64x64x32
Conv3D	4x4x4	2x2x2	1x1x1	LReLU($\alpha = 0.2$)	No	64x32x32x16
Conv3D	4x4x4	2x2x2	1x1x1	LReLU($\alpha = 0.2$)	Yes	128x16x16x8
Residual Layer						
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	Yes	128x16x16x8
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	Yes	128x16x16x8
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	Yes	128x16x16x8
Conv3D	4x4x4	2x2x2	1x1x1	LReLU($\alpha = 0.2$)	Yes	256x8x8x4
Residual Layer						
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	Yes	256x8x8x4
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	Yes	256x8x8x4
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	Yes	256x8x8x4
Conv3D	2x2x2	1x1x1	1x1x1	LReLU($\alpha = 0.2$)	Yes	512x9x9x5
<hr/>						
Motion	Kernel Size	Stride	Padding	Activation	BatchNorm	Out shape
ConvTran3D	2x2x2	1x1x1	1x1x1	ReLU	Yes	256x8x8x4
ConvTran3D	4x4x4	2x2x2	1x1x1	ReLU	Yes	128x16x16x8
Residual Layer						
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	Yes	128x16x16x8
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	Yes	128x16x16x8
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	Yes	128x16x16x8
ConvTran3D	4x4x4	2x2x2	1x1x1	ReLU	Yes	64x32x32x16
Residual Layer						
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	Yes	64x32x32x16
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	Yes	64x32x32x16
3D Residual Block	3x3x3	1x1x1	1x1x1	ReLU	Yes	64x32x32x16
ConvTran3D	4x4x4	2x2x2	1x1x1	Sigmoid	No	3x64x64x32
<hr/>						
Discr.	Kernel Size	Stride	Padding	Activation	BatchNorm	Out shape
Conv3D	4x4x4	2x2x2	1x1x1	LReLU($\alpha = 0.2$)	Yes	1024x4x4x2
Conv3D	4x4x4	4x4x2	—	Softmax	No	2x1x1x1

Table 2: Discriminator architecture. Note that the structure of transposed convolutions used in *Motion Stream* is symmetrical to the one of convolutions in the *Shared* part. *LReLU* stands for Leaky ReLU, while the layers marked with *ConvTran* are transposed convolutions. The output of the *motion stream* has 3 channels: 1 for segmentation and 2 for optical flow.

User preference %	
Golf course	
VOS-GAN vs VGAN [53]	80.5 / 19.5
VOS-GAN vs TGAN [41]	65.1 / 34.9
VOS-GAN vs MoCoGAN [50]	58.2 / 41.8
Weizmann Action dataset	
VOS-GAN vs VGAN [53]	82.2 / 17.8
VOS-GAN vs TGAN [41]	70.3 / 29.7
VOS-GAN vs MoCoGAN [50]	39.7 / 60.3

Table 3: User preference score (percentage) on video generation quality on different types of generated videos.

dataset) for learning well the scene; 2) “Golf course” contains many video sequences with high variability in terms of

motion and appearance — variability that MoCoGAN is not able to learn. However, VOS-GAN confirms its capabilities to learn better motion, indeed, it achieves a higher FID than MoCoGAN, substantiating also our previous claims on the different performance of MoCoGAN on the two employed datasets.

4.4 Video Object Segmentation

The first part of this evaluation aims at investigating the contribution of adversarial training for video object segmentation, by assessing the quality of the foreground maps computed by our video object segmentation approach in four different settings:

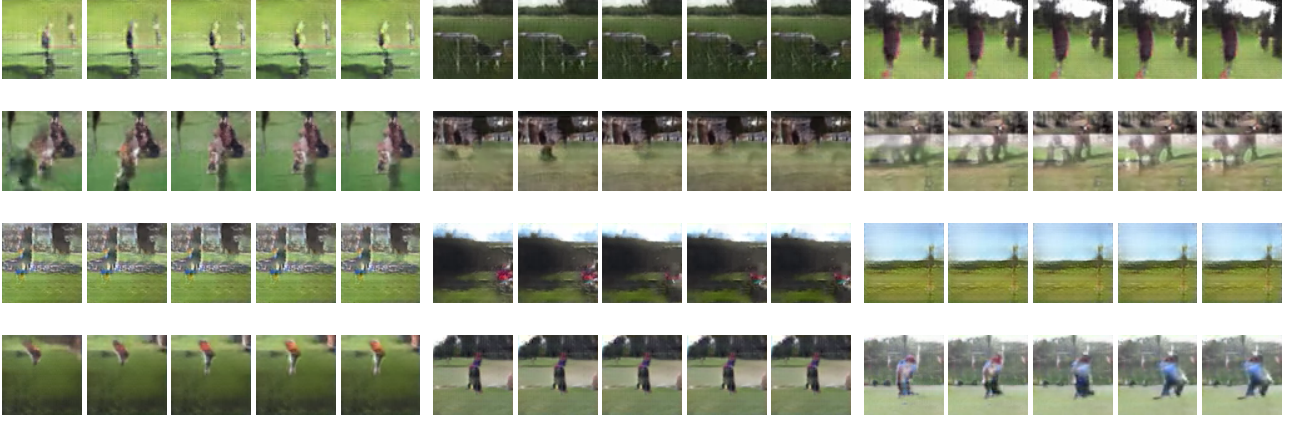


Fig. 4: **Frame samples on “Golf course”**. (First row) VGAN-generated video frames show very little object motion, while (second row) TGAN-generated video frames show motion, but the quality of foreground appearance is low. (Third row) MoCoGAN-generated videos: background quality is high, but there is little object motion. VOS-GAN (fourth row) generates video frames with a good compromise between object motion and appearance.



Fig. 5: **Frame samples on Weizmann Action dataset**. Both MoCoGAN (first row) and VOS-GAN (second row) are able to generate realistic videos: the former with higher resolution, but lower motion quality as shown in the quantitative video generation performance.

	SLERP	LSTM	Segmentation	OF	α	FCD ↓	FID ↓	IS ↑
Baseline						5.41 ± 0.020	44.02 ± 0.45	1.98 ± 0.019
Model 1	✓					5.27 ± 0.024	41.76 ± 0.53	2.07 ± 0.024
Model 2	✓	✓				4.77 ± 0.018	40.54 ± 0.41	2.29 ± 0.019
Model 3	✓	✓	✓			4.61 ± 0.026	38.79 ± 0.53	2.73 ± 0.011
Model 4	✓	✓	✓	✓		4.22 ± 0.022	33.18 ± 0.43	3.09 ± 0.011
VOS-GAN	✓	✓	✓	✓	✓	4.11 ± 0.018	31.32 ± 0.46	3.16 ± 0.032

Table 4: **Ablation studies**. Quantitative evaluation on the “Golf course” dataset of different configurations of the proposed model. The baseline is the GAN architecture described in Sect. 3 trained with traditional adversarial loss and using a random latent variable for foreground content in addition to the background latent space.

- *synthetic*: we use the motion stream, i.e. segmentation subnetwork, from the architecture of our discriminator and train it from scratch with the foreground masks synthesized by our generator trained on “Golf course” video dataset;
- *adversarial*: we use segmentation subnetwork from the discriminator of VOS-GAN trained on the “Golf dataset”;
- *fine-tuned synthetic*: the segmentation network trained in the *synthetic* modality is then fine-tuned on ground-truth

masks of the benchmark datasets’ (DAVIS 2016/2017 and SegTrack-v2) training sets;

- *fine-tuned adversarial*: analogously, we use the segmentation model from the *adversarial* training scenario and fine-tune it on real segmentation masks.

As baseline we select the segmentation network of our GAN model trained, in a supervised way (i.e., using annotations at training time), from scratch on the training splits of the employed datasets. As metrics, we use mean values for region similarity and contour accuracy, i.e., $\mathcal{M}_{\mathcal{J}}$ and $\mathcal{M}_{\mathcal{F}}$.

	FCD ↓	FID ↓	IS ↑
Golf course			
VGAN	10.61 ± 0.015	45.32 ± 0.31	1.74 ± 0.021
TGAN	5.74 ± 0.025	42.58 ± 0.39	2.02 ± 0.019
MoCoGAN	5.01 ± 0.023	34.53 ± 0.42	2.47 ± 0.013
VOS-GAN	4.11 ± 0.018	31.32 ± 0.46	3.16 ± 0.032
Weizmann Action dataset			
VGAN	5.18 ± 0.021	7.64 ± 0.041	2.78 ± 0.027
TGAN	4.53 ± 0.029	7.00 ± 0.027	2.94 ± 0.016
MoCoGAN	4.07 ± 0.018	5.74 ± 0.031	3.76 ± 0.025
VOS-GAN	4.24 ± 0.016	5.71 ± 0.034	3.29 ± 0.020

Table 5: Comparison of quantitative generation performance, in terms of foreground content distance (FCD), Fréchet Inception Distance (FID) and Inception Score (IS), against VGAN, TGAN and MoCoGAN, respectively, on the “Golf course” and the Weizmann action dataset.

Tab. 6 shows the obtained performance: our segmentation network obtains fair results even when trained without labeled data (first two rows of the table). There is a significant gap between region similarity $\mathcal{M}_{\mathcal{J}}$ (higher) and contour accuracy $\mathcal{M}_{\mathcal{F}}$ (lower), meaning that the model performs better in motion localization than in object segmentation. Results also show that our adversarial model fine-tuned on real data (fourth line) significantly outperforms, by a margin of about 10%, our baseline (fifth line), indicating that pre-training, with self-supervision, the segmentation model obtains better accuracy. Pre-training the segmentation model with synthetic videos before fine-tuning on real data (third line) leads also to increased (about 2%) performance than the baseline. Thus, our video generation model acts as a data distillation approach [39], i.e., through injecting synthetic video masks into the training procedure, our approach combines predictions from arbitrary transformations (enabled by the generator) of unlabeled videos. We further quantify the percentage of training examples needed to obtain satisfactory results both for our baseline and the adversarial fine-tuned model. The results on the three employed datasets are given in Fig. 7 and indicate that our adversarial model requires fewer annotations to reach satisfactory performance than our baseline, which requires at least twice as many annotations. The obtained results, thus, demonstrate that our approach not only yields better segmentation results, but it also allows us to reduce the needed training data. Fine-tuning the segmentation subnetwork on real data (third and fourth rows in Tab. 6) has the effect to focus better on target objects and to cope with camera motion (that is not considered by the video generation model), as demonstrated by the performance improvement reported in rows three and four. This effect is also shown in Fig. 6, that shows that the segmentation maps predicted by our purely-unsupervised method highlight object motion, but object contours are not

accurate. Training the segmentation network in an adversarial framework (rather than directly on the fake maps) leads to a significant performance increase, both with and without fine-tuning on real data. Fine-tuning the whole GAN video generation on the three employed video benchmarks, instead, does not yield a significant performance improvement, but relaxes the requirements on available manual annotations.

On the other hand, the advantage of using less annotated data comes at the cost of training the GAN for video generation using unlabeled videos. For this reason, we quantify how the amount of unlabeled videos for video generation affects video object segmentation performance. This evaluation is carried out on the adversarial configuration for video object segmentation, and uses region similarity and contour accuracy as metrics. Fig. 8 shows an almost linear dependence between video object segmentation accuracy and the number of unlabeled videos used for generation by our VOS-GAN, suggesting that using more unlabeled videos in training the adversarial models may ideally lead to better results as the approach effectively learns motion dynamics.

Since our proposed adversarial framework is used for unsupervised pre-training video segmentation models, we compare it, on DAVIS 2017, to: a) two approaches that perform optical flow estimation (both non-learning-based [8] and learning-based [18] optical flow) and assign foreground pixels based on the empirically-optimal threshold over flow magnitude; b) the approach proposed in [54], relaxing its constraint of having the first annotated label at inference time by replacing the needed frame with a binary image obtained by thresholding the optical flow between the first two frames, as done in the previous methods. We also compare the performance of our adversarial learning method fine-tuned on DAVIS 2017 to our discriminator network trained adversarially to estimate only optical flow and then fine-tuned on the video segmentation task (“OF estimation + FT” in Table 7). Results, in terms of $\mathcal{M}_{\mathcal{J}}$ and $\mathcal{M}_{\mathcal{F}}$, are given in Table 7. Our purely adversarial unsupervised approach learns better features (segmentation performance higher by about 8-10%) than both optical flow methods and [54]. Also, when fine-tuned, our approach gives better segmentation performance than our generation model trained without segmentation mask self-supervision (i.e., the GAN model using only optical flow) suggesting that, indeed, our self-supervision strategy through synthetic masks during video generation learns better representations for the segmentation task.

We finally compare our approach to state-of-the-art unsupervised video object segmentation methods (i.e., not using any annotations at test time), namely, [7], [34], [48], [20], [24] on DAVIS 2017; [7], [34], [21] and [24] on DAVIS 2016; [34], [37], [46], [13] on SegTrack-v2. For a fair comparison we did not test any of the existing code, but just re-

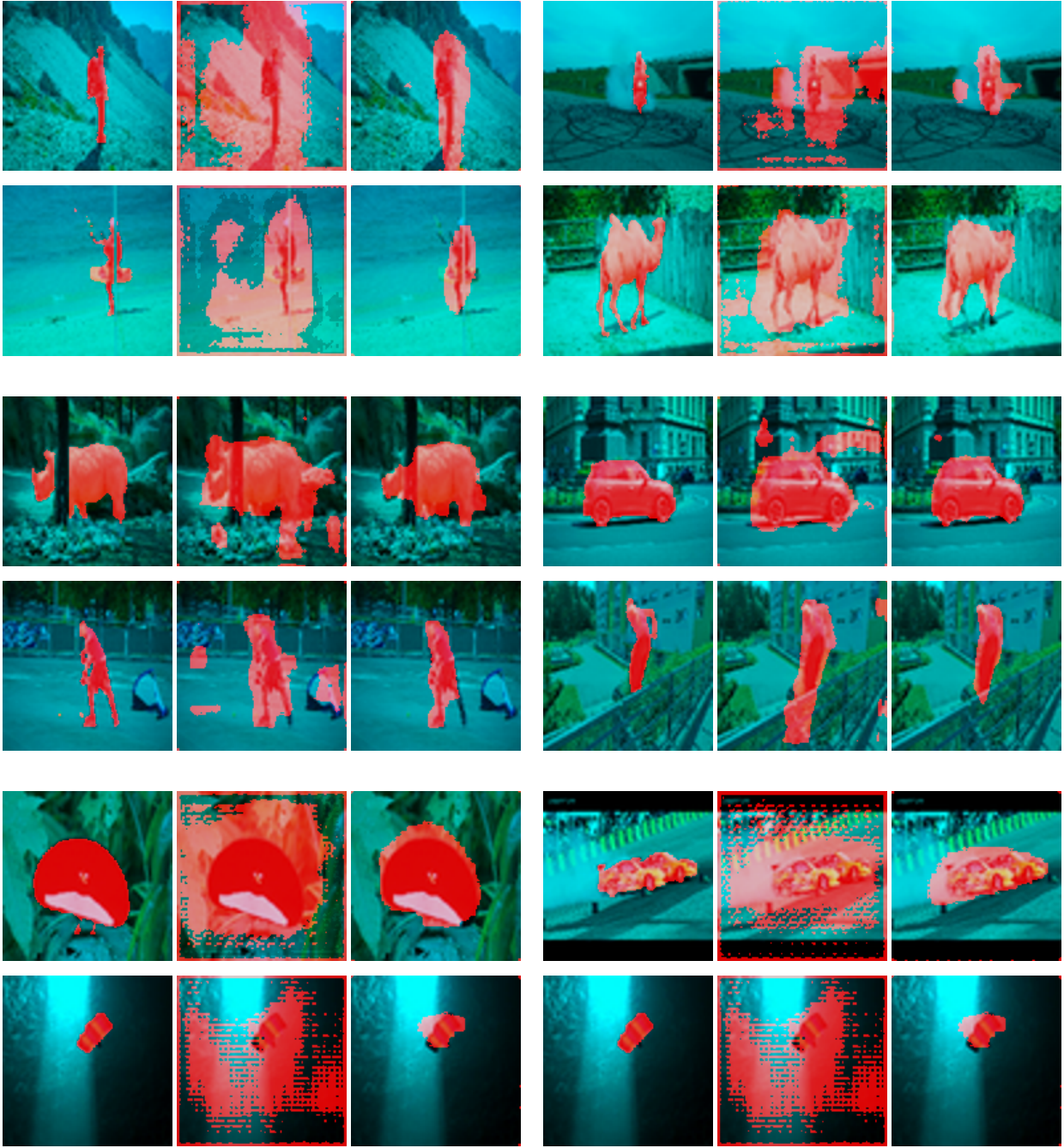


Fig. 6: **Video object segmentation results on multiple datasets.** First two rows: DAVIS 2017; second two rows: DAVIS 2016; last two rows: SegTrack-v2. Each image block shows the original image with ground truth, the image with the map obtained by the adversarially trained VOS-GAN and the image with the segmentation mask when fine-tuning the model on the given dataset. Images are shown at the original resolution provided by our models, i.e., 64×64 . The best segmentation masks are obtained on DAVIS 2016, as also demonstrated by the results in Table 6. Our purely unsupervised approach is still far from the performance of supervised training ones, but it is able to detect object movements in videos characterized by strong camera motion (as those hereby reported).

Learning	Model	DAVIS-16		DAVIS-17		SegTrack-v2	
		$\mathcal{M}_{\mathcal{F}}$	$\mathcal{M}_{\mathcal{J}}$	$\mathcal{M}_{\mathcal{F}}$	$\mathcal{M}_{\mathcal{J}}$	$\mathcal{M}_{\mathcal{F}}$	$\mathcal{M}_{\mathcal{J}}$
Unsupervised	Synthetic VOS	25.41	33.66	19.83	21.87	20.32	23.72
	Adversarial VOS	31.22	38.11	22.57	27.01	24.11	27.42
Supervised	Synthetic VOS FT	60.85	64.66	52.54	55.12	56.43	59.95
	Adversarial VOS FT	67.35	71.24	56.10	61.65	61.14	65.02
	Baseline	57.05	62.85	50.96	53.41	53.26	57.03

Table 6: Video object segmentation results (in percentage). The first two rows report the results obtained by training the model without annotations, while the third and fourth rows report the performance when fine-tuning on the video benchmarks. The last row shows the results achieved by our baseline trained purely in a supervised way.

	Method	$\mathcal{M}_{\mathcal{J}}$	$\mathcal{M}_{\mathcal{F}}$
Unsupervised	Optical Flow [8]	14.55	10.32
	Optical Flow [18]	17.21	15.43
	Unsupervised [54]	19.86	17.47
	Adversarial VOS	27.01	22.57
Supervised	OF estimation + FT	54.32	47.73
	Adversarial VOS + FT	61.65	56.10

Table 7: Performance on DAVIS 2017 by unsupervised (i.e., not using any annotated data at inference time) video object segmentation methods, as well as of unsupervised models fine-tuned on DAVIS 2017.

port performance from the literature, specifically from [29] for DAVIS 2016, from [24] for DAVIS 2017 and from [13] for SegTrack-v2. The achieved results are given in Tab. 8 and show that our approach is almost on par with the best performing method on the DAVIS benchmarks and outperforms existing methods on SegTrack-v2. Most importantly, according to Fig. 7 and results in Table 8, it leads to performance enhancement requiring less annotated data.

4.5 Video Action Recognition

Finally, in order to evaluate the representational power of features learned by the methods under comparison, we employ the models' discriminators (after the initial training for video generation) to perform action recognition on the UCF101 and Weizmann Action datasets. Furthermore, results on video action recognition serve also as an additional means to verify video generation performance especially in learning motion dynamics.

This analysis is carried out on two different training settings: a) *transfer learning*: each model's discriminator is used as a feature extractor by removing the final real/fake output layer and training a linear classifier on the exposed features; b) *fine-tuning*: the real/fake output layer is replaced with a softmax layer and the whole discriminator is fine-tuned for classification.

DAVIS 2016						
	[7]	[34]	[48]	[20]	[24]	Ours
\mathcal{J} - Region Similarity						
Mean \mathcal{M}	64.1	57.5	70.0	70.7	76.3	71.2
Recall \mathcal{O}	73.1	65.2	85.0	83.5	89.2	84.6
\mathcal{F} - Contour accuracy						
Mean \mathcal{M}	59.3	53.6	65.9	65.3	71.1	67.3
Recall \mathcal{O}	65.8	57.9	79.2	73.8	82.8	77.9
$\mathcal{J}\&\mathcal{F}$ - Average						
Mean \mathcal{M}	61.7	55.5	67.9	68.0	73.7	69.2
Recall \mathcal{O}	69.4	61.5	82.1	78.6	86.0	81.2
DAVIS 2017						
	[7]	[34]	[21]	[24]	Ours	
\mathcal{J} - Region Similarity						
Mean \mathcal{M}		51.4	49.6	45.0	63.3	61.7
Recall \mathcal{O}		55.5	52.9	46.4	72.9	61.4
\mathcal{F} - Contour accuracy						
Mean \mathcal{M}		48.6	48.0	44.8	61.2	56.1
Recall \mathcal{O}		49.4	46.8	43.0	67.8	53.8
$\mathcal{J}\&\mathcal{F}$ - Average						
Mean \mathcal{M}		50.0	48.8	44.9	62.2	58.9
Recall \mathcal{O}		52.4	49.8	44.7	70.3	57.6
SegTrack-v2						
	[34]	[37]	[46]	[13]	Ours	
\mathcal{J} - Region Similarity						
Mean \mathcal{M}		50.1	30.4	49.9	61.1	65.0

Table 8: Comparison to state-of-the-art unsupervised methods on DAVIS 2016, DAVIS 2017 and SegTrack-v2 benchmarks in terms of region similarity (\mathcal{J}) contour accuracy (\mathcal{F}). On SegTrack-v2 we report only mean \mathcal{J} as comparing methods employ only that measure since recall and contour accuracy metrics were introduced only in [36]. In bold best performance, in italic the second best performance.

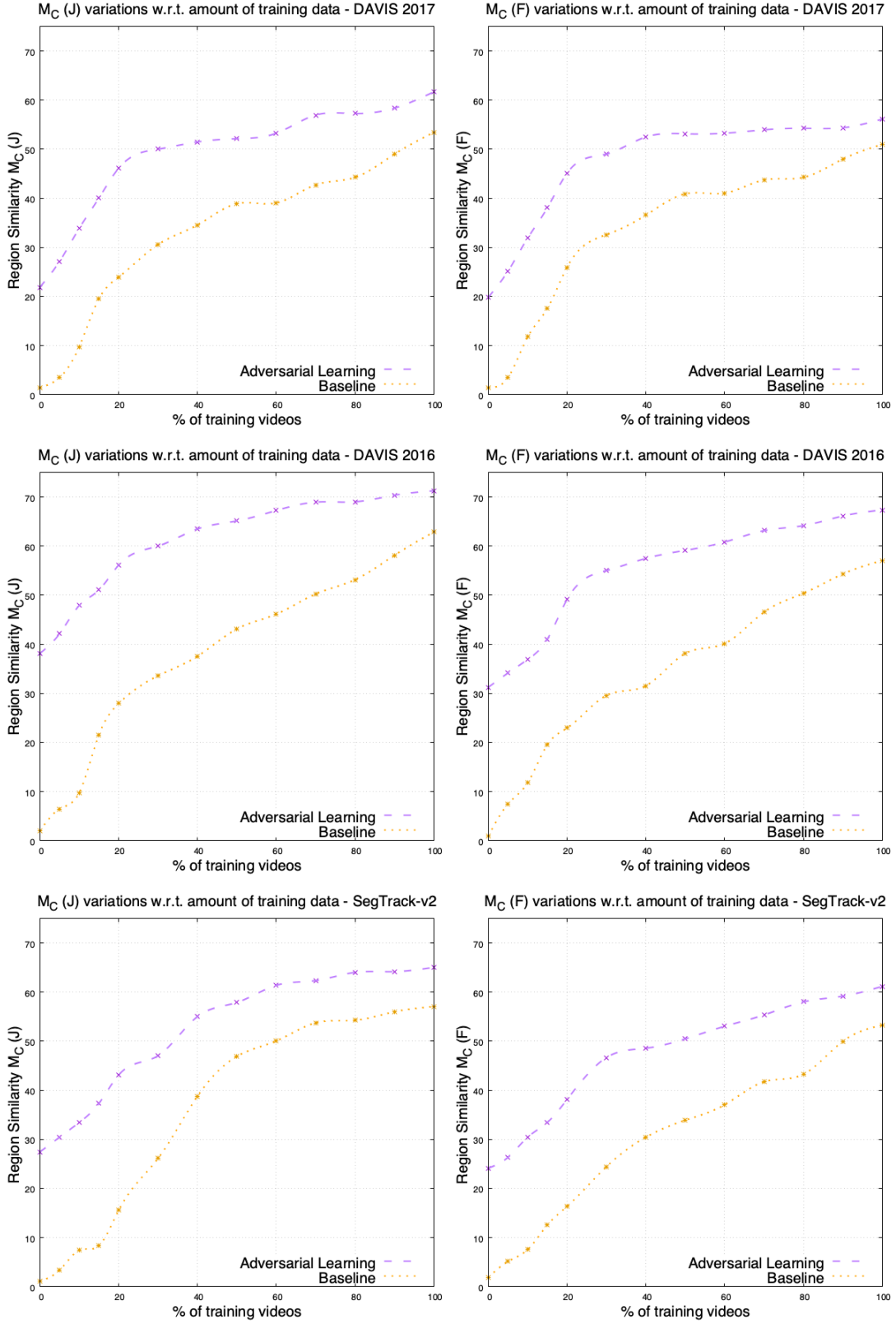


Fig. 7: Performance, in terms of region similarity $\mathcal{M}_{\mathcal{J}}$ and contour accuracy $\mathcal{M}_{\mathcal{F}}$, w.r.t. to the percentage of training images of DAVIS 2017 (first row), DAVIS 2016 (second row) and SegTrack-v2 (third row) datasets. Note that “% of training frames” is related to the size of each video benchmark employed in the evaluation.

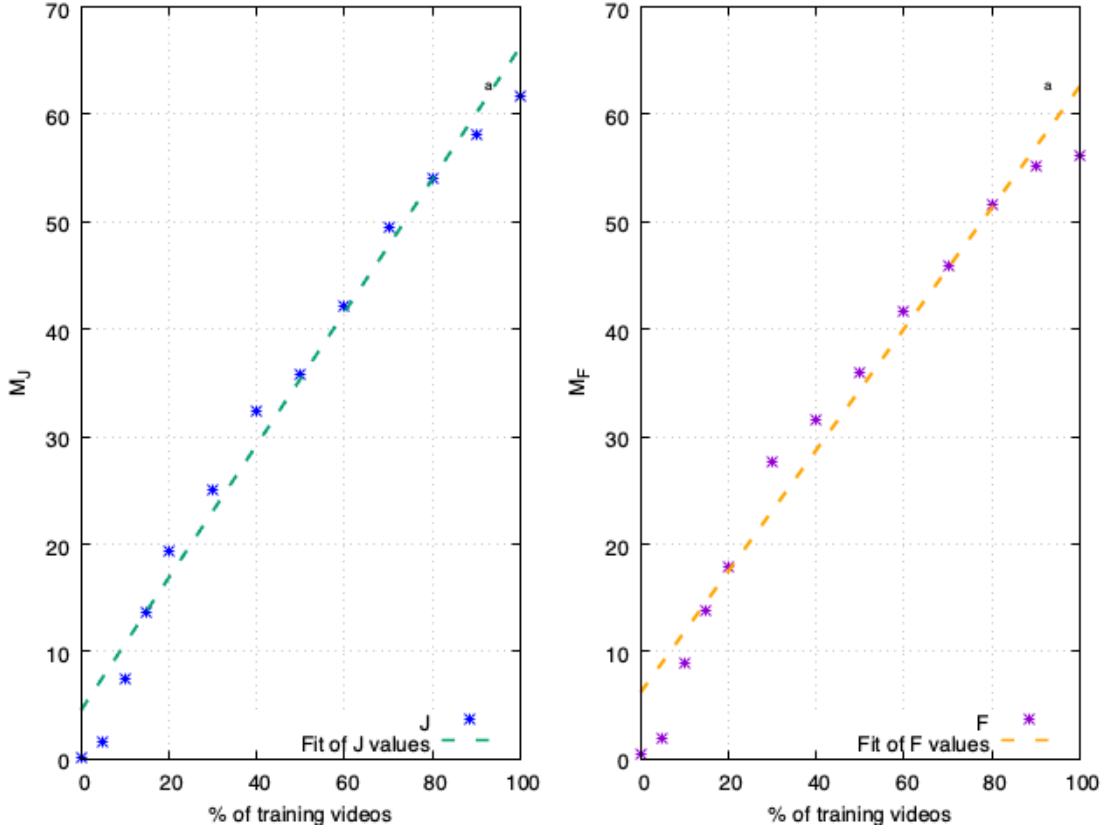


Fig. 8: Region similarity $\mathcal{M}_{\mathcal{J}}$ and contour similarity $\mathcal{M}_{\mathcal{F}}$ w.r.t. to the percentage of videos used for training our GAN-based video generation model fine-tuned on the DAVIS 2017 dataset. Line fitting is performed by least square regression.

In order to make the comparison fair and consistent with the initial adversarial training, we separately evaluate the models originally trained on “Golf course” from those trained on Weizmann Action dataset: hence, Tab. 9 and 10 report the classification accuracy on UCF101 and Weizmann Action dataset by comparing the discriminators originally trained for video generation on “Golf course”; similarly, Tab. 11 and 12 report the classification accuracy by the discriminators originally trained on Weizmann Action dataset. The results show that our VOS-GAN discriminator outperforms VGAN, TGAN and MoCoGAN in all the considered scenarios. Note that all models quickly overfit on Weizmann Action dataset, due to the small size of the dataset, which likely explains why transfer learning performs better than fine-tuning.

5 Conclusion

We propose a framework for unsupervised learning of motion cues in videos. It is based on an adversarial video generation approach — VOS-GAN — that disentangles background and foreground dynamics. Motion generation is improved by learning a suitable object motion latent space

Settings	Models		
	VGAN	TGAN	VOS-GAN _G
Transfer learning	39.19	32.45	41.02
Fine-tuning	45.43	36.94	49.33

Table 9: Video action recognition accuracy on UCF101 of the models originally trained on “Golf course” (classification accuracy, in percentage).

Settings	Models	
	MoCoGAN	VOS-GAN _W
Transfer learning	18.47	29.42
Fine-tuning	32.83	45.66

Table 10: Video action recognition accuracy on UCF101 of the models originally trained on Weizmann Action dataset (classification accuracy, in percentage).

and by controlling the discrimination process with pixel-wise dense prediction of moving objects through a self-supervision mechanism, in which segmentation masks are internally synthesized by the generator.

Settings	Models		
	VGAN	TGAN	VOS-GAN _G
Transfer learning	65.87	54.19	68.79
Fine-tuning	64.41	52.30	66.80

Table 11: Video action recognition accuracy on Weizmann Action dataset of the models originally trained on “Golf course” (classification accuracy, in percentage).

Settings	Models	
	MoCoGAN	VOS-GAN _w
Transfer learning	70.76	74.29
Fine-tuning	67.02	71.63

Table 12: Video action recognition accuracy on Weizmann Action Dataset of the models originally trained on the same dataset (classification accuracy, in percentage).

Extensive experimental evaluation showed that our VOS-GAN outperforms existing video generation methods, namely, VGAN [53], TGAN [41], MoCoGAN [50], especially in modeling object motion. The capability of our approach to better model object motion is further demonstrated by the fact that the learned (in an unsupervised way) features can be used for effective video object segmentation and video action recognition tasks.

Finally, it has to be noted that, although we introduce a new segmentation network as part of the adversarial framework, it is quite general and can easily integrated into any segmentation model, by adding the optical flow dense prediction stream and using masks synthesized by the generator for supervision.

References

- Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: ICML (2017)
- Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: CVPR (2017)
- Brox, T., Malik, J.: Object segmentation by long term analysis of point trajectories. In: ECCV (2010)
- Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: CVPR (2017)
- Denton, E.L., Chintala, S., Szlam, A., Fergus, R.: Deep generative image models using a laplacian pyramid of adversarial networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) NIPS (2015)
- Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction (2015)
- Faktor, A., Irani, M.: Video segmentation by non-local consensus voting. In: BMVC (2014)
- Farneback, G.: Two-frame motion estimation based on polynomial expansion. In: Proceedings of the 13th Scandinavian Conference on Image Analysis. pp. 363–370. SCIA’03, Springer-Verlag, Berlin, Heidelberg (2003)
- Fragkiadaki, K., Zhang, G., Shi, J.: Video segmentation by tracing discontinuities in a trajectory embedding. In: CVPR (2012)
- Giordano, D., Murabito, F., Palazzo, S., Spampinato, C.: Superpixel-based video object segmentation using perceptual organization and location prior. In: CVPR (2015)
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS (2014)
- Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(12), 2247–2253 (2007)
- Haller, E., Leordeanu, M.: Unsupervised object segmentation in video by efficient selection of highly probable positive features. In: ICCV (2017)
- Hara, K., Kataoka, H., Satoh, Y.: Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In: CVPR (2018)
- He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NIPS (2017)
- Huang, X., Li, Y., Poursaeed, O., Hopcroft, J., Belongie, S.: Stacked generative adversarial networks. In: CVPR (2017)
- Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet 2.0: Evolution of optical flow estimation with deep networks. In: CVPR (2017)
- Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
- Jain, S.D., Xiong, B., Grauman, K.: Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In: CVPR (2017)
- Jang, Y., Kim, G., Song, Y.: Video prediction with appearance and motion conditions. In: ICML (2018)
- Jang, Y., Kim, G., Song, Y.: Video prediction with appearance and motion conditions. In: ICML (2018)
- Keuper, M., Andres, B., Brox, T.: Motion trajectory segmentation via minimum cost multicuts. In: ICCV (2015)
- Koh, Y.J., Kim, C.: Primary object segmentation in videos based on region augmentation and reduction. In: CVPR (2017)
- Lai, W.S., Huang, J.B., Yang, M.H.: Semi-supervised learning for optical flow with generative adversarial networks. In: NIPS (2017)
- Lee, Y.J., Kim, J., Grauman, K.: Key-segments for video object segmentation. In: ICCV (2011)
- Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
- Mahasseni, B., Lam, M., Todorovic, S.: Unsupervised video summarization with adversarial lstm networks. In: CVPR (2017)
- Maninis, K.K., Caelles, S., Chen, Y., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: Video object segmentation without temporal information. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018)
- Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: ICCV (2017)
- Odena, A.: Semi-supervised learning with generative adversarial networks. arXiv preprint arXiv:1606.01583 (2016)
- Ohnishi, K., Yamamoto, S., Ushiku, Y., Harada, T.: Hierarchical video generation from orthogonal information: Optical flow and texture. In: AAAI (2018)
- Ohnishi, K., Yamamoto, S., Ushiku, Y., Harada, T.: Hierarchical video generation from orthogonal information: Optical flow and texture. In: AAAI (2018)
- Papazoglou, A., Ferrari, V.: Fast object segmentation in unconstrained video. In: ICCV (2013)

35. Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., Sorkine-Hornung, A.: Learning video object segmentation from static images. In: CVPR (2017)
36. Perazzi, F., Pont-Tuset, J., McWilliams, B., Gool, L.V., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR (2016)
37. Prest, A., Leistner, C., Civera, J., Schmid, C., Ferrari, V.: Learning object class detectors from weakly annotated video. In: CVPR. pp. 3282–3289 (2012)
38. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. ICLR (2016)
39. Radosavovic, I., Dollár, P., Girshick, R.B., Gkioxari, G., He, K.: Data distillation: Towards omni-supervised learning. In: CVPR (2018)
40. Roth, K., Lucchi, A., Nowozin, S., Hofmann, T.: Stabilizing training of generative adversarial networks through regularization. In: NIPS (2017)
41. Saito, M., Matsumoto, E., Saito, S.: Temporal generative adversarial nets with singular value clipping. In: ICCV (2017)
42. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., Chen, X.: Improved techniques for training gans. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) NIPS (2016)
43. Shoemake, K.: Animating rotation with quaternion curves. SIGGRAPH (1985)
44. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild (2012)
45. Souly, N., Spampinato, C., Shah, M.: Semi supervised semantic segmentation using generative adversarial network. In: ICCV (2017)
46. Stretcu, O., Leordeanu, M.: Multiple frames matching for object discovery in video. In: BMVC (2015)
47. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)
48. Tokmakov, P., Alahari, K., Schmid, C.: Learning motion patterns in videos. In: CVPR (2017)
49. Tsai, D., Flagg, M., Nakazawa, A., Rehg, J.M.: Motion coherent tracking using multi-label mrf optimization. *International Journal of Computer Vision* **100**(2), 190–202 (2012)
50. Tulyakov, S., Liu, M.Y., Yang, X., Kautz, J.: Mocogan: Decomposing motion and content for video generation. In: CVPR (2018)
51. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR (2017)
52. Villegas, R., Yang, J., Hong, S., Lin, X., Lee, H.: Decomposing motion and content for natural video sequence prediction. ICLR (2017)
53. Vondrick, C., Pirsaviash, H., Torralba, A.: Generating videos with scene dynamics. In: NIPS (2016)
54. Vondrick, C., Shrivastava, A., Fathi, A., Guadarrama, S., Murphy, K.: Tracking emerges by colorizing videos. In: ECCV (2018)
55. Vondrick, C., Torralba, A.: Generating the future with adversarial transformers. In: CVPR (2017)
56. Vondrick, C., Torralba, A.: Generating the future with adversarial transformers. In: CVPR (2017)
57. Wang, T.C., Liu, M.Y., Zhu, J.Y., Liu, G., Tao, A., Kautz, J., Catanzaro, B.: Video-to-video synthesis. In: NeurIPS (2018)
58. Wang, W., Shen, J., Porikli, F.: Saliency-aware geodesic video object segmentation. In: CVPR (2015)
59. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: CVPR (2017)
60. Yi, Z., Zhang, H., Tan, P., Gong, M.: Dualgan: Unsupervised dual learning for image-to-image translation. In: ICCV (2017)
61. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: ICCV (2017)
62. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV (2017)