Polar prediction of natural videos

Pierre-Étienne H. Fiquet [1] **Eero P. Simoncelli** [1 2]

Abstract

Observer motion and continuous deformations of objects and surfaces imbue natural videos with distinct temporal structures, enabling partial prediction of future frames from past ones. Conventional methods first estimate local motion, or optic flow, and then use it to predict future frames by warping or copying content. Here, we explore a more direct methodology, in which each frame is mapped into a learned representation space where the structure of temporal evolution is more readily accessible. Motivated by the geometry of the Fourier shift theorem and its group-theoretic generalization, we formulate a simple architecture that represents video frames in learned local polar coordinates. Specifically, we construct networks in which pairs of convolutional channel coefficients are treated as complex-valued, and are optimized to evolve with slowly varying amplitudes and linearly advancing phases. We train these models on next-frame prediction in natural videos, and compare their performance with that of conventional methods using optic flow as well as predictive neural networks. We find that the polar predictor achieves better performance while remaining interpretable and fast, thereby demonstrating the potential of a flow-free video processing methodology that is trained end-to-end to predict natural video content.

1. Introduction

One way to frame the fundamental problem of vision is that of representing the signal in a form that is more useful for performing visual tasks, be they estimation, recognition, or motor action. Perhaps the most general "task" is that of temporal prediction, which has been proposed as a fundamental goal for unsupervised learning of visual representations (Földiák, 1991). But previous research along these lines has generally focused on estimating temporal transformations rather than using them to predict: for example, extracting slow features (Wiskott & Sejnowski, 2002), or finding sparse codes that have slow amplitudes and phases (Cadieu & Olshausen, 2012).

In video processing and computer vision, a common strategy for temporal prediction is to first estimate local translational motion, and to then (assuming no acceleration) use this to warp and/or copy previous content to predict the next frame. Such motion compensation is a fundamental component in video compression schemes like MPEG (Wiegand et al., 2003). These video coding standards are the result of decades of engineering efforts, and have enabled reliable and efficient digital video communication that is now commonplace. But motion estimation is a difficult nonlinear problem, and existing methods fail in regions where temporal evolution is not translational and smooth: for example, expanding or rotating motions, discontinuous motion at occlusion boundaries, or mixtures of motion arising from semi-transparent surfaces (e.g., viewing the world through a dirty pane of glass). In compression schemes, these failures of motion estimation lead to prediction errors, which must then be repaired by sending additional corrective bits.

Human perception does not seem to suffer from such failures - subjectively, we can anticipate the time-evolution of visual input even in the vicinity of these commonly occurring non-translational changes. In fact, those changes are often highly informative, as they reveal object boundaries, and provide ordinal depth and other information about the visual scene. This suggests that the human visual system uses a different strategy, perhaps bypassing altogether the estimation of local motion, to represent and predict evolving visual input. Toward this end, and inspired by recent hypotheses that primate visual representations support prediction by "straightening" the temporal trajectories of naturally-occurring input (Hénaff et al., 2019), we formulate an objective for learning an image representation that facilitates prediction by linearizing the temporal trajectories of frames of natural video.

To motivate the separation of instantaneous spatial representation from temporal prediction, we first consider the special case of rigidly translating video content. When ex-

[1]Center for Neural Science, New York University, New York, USA [2]Center for Computational Neuroscience, Flatiron Institute, New York, USA. Correspondence to: Pierre-Étienne H. Fiquet <pef246@nyu.edu>.

Technical report, January 2023.

pressed in the frequency domain, translation corresponds to linear phase advancement (section 2.1), and prediction of rigidly translating content reduces to angular extrapolation (section 2.2). We generalize this factorization using group representation theory (section 2.3), and describe a neural network architecture that maps individual video frames to a latent complex-valued representation. Within this latent space, coefficients can be temporally predicted by phase advancement and then mapped back to generate an estimated frame. The entire systems may then be trained end-to-end to minimize next frame prediction errors (section 3). We report training results of several such systems, and show that they produce systematic improvements in predictive performance over both conventional motion compensation methods, and direct predictive neural networks (section 4). Finally, we relate this approach to previous work (section 5) and discuss its significance and implications (section 6).

2. Background

2.1. Base case: the Fourier shift theorem

Our approach is motivated by the well-known behavior of Fourier representations with respect to signal translation (note that this elementary example will later lead to our proposed generalization). Specifically, the complex exponentials that make up the Fourier basis are the eigenfunctions of the translation operator, and translation of inputs produces systematic phase advances of frequency coefficients. Let $x \in \mathbb{R}^N$ be a discrete signal indexed by spatial location $n \in [0, N-1]$, and let $\widetilde{x} \in \mathbb{C}^N$ be its Fourier transform indexed by $k \in [0, N-1]$. We write $x^v(n) = x(n-v)$, the translation of x by v modulo N (ie. circular shift with period N). Defining $\phi = e^{i2\pi/N}$, the primitive N-th root of unity, and $\mathcal{F}_{nk} = \phi^{nk}$, the $N \times N$ Fourier matrix, we can express the Fourier shift theorem[1] as: $x^v(n) = \frac{1}{N}\mathcal{F}D(v)\mathcal{F}^* x(n)$, where $\mathcal{F}^*$ is the conjugate transpose of the Fourier matrix and $D(v) = \mathrm{diag}(\phi^0, ..., \phi^{-(n-1)v})$ is a diagonal matrix.

This relationship may be depicted in a compact diagram:

$$\begin{array}{ccc} \widetilde{x}(k) & \xrightarrow[\text{advance phase}]{} & \phi^{-kv}\widetilde{x}(k) \\ \uparrow{\scriptstyle \mathcal{F}^*} & & \downarrow{\scriptstyle \mathcal{F}} \\ x(n) & \xrightarrow[\text{shift}]{} & x(n-v). \end{array} \tag{1}$$

[1] Proof by substituting $m = n - v$:

$$\widetilde{x^v}(k) = \sum_{n=0}^{N-1} \phi^{-kn} x(n-v) = \sum_{m=-v}^{N-1-v} \phi^{-kv}\phi^{-km} x(m)$$
$$= \phi^{-kv} \sum_{n=0}^{N-1} \phi^{-kn} x(n) = \phi^{-kv}\widetilde{x}(k).$$

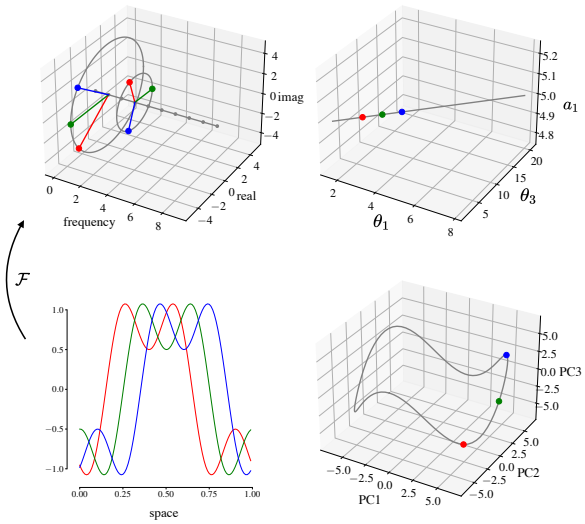


Figure 1. Translation of a 1D signal consisting of a sum of two sinusoidal components: $x(n,t) = \sin(2\pi(n-t)) + \sin(2\pi 3(n-t))/2$. Lower left: three snapshots of the signal as it translates. Lower right: In the high-dimensional space representing the signal (each axis corresponding to the signal value at one location), the temporal trajectory is highly curved. Shown is the projection of the signal vector into the 3D space of the top three principal components. Three colored points indicate the three snapshots in lower left panel. Upper left: Fourier transform of the signal, showing complex-valued coefficients as a function of frequency. In this representation the temporal trajectory corresponds to linearly increasing phase of the two sinusoidal components, each at a rate proportional to its frequency. Upper right: a polar coordinate transform to amplitude and phase of each frequency component leads to a representation that evolves along a straight line, and is thus readily predictable (phases are unwrapped for display purposes).

In the context of our goals, the diagram illustrates the point that transforming to the frequency domain renders translation a "simpler" operation: a phase advance is a rotation in the two dimensional (complex) plane.

2.2. Prediction via angular extrapolation

Now consider observations of a signal that translates at a constant velocity over time, $x(n,t) = y(n - vt)$. Although the temporal evolution is easy to describe, it traces a highly non-linear trajectory in the signal state space, rendering prediction difficult (specifically, linear extrapolation fails). As an example, Figure 1 shows a signal consisting of a sum of two sinusoidal components. Transforming the signal to the frequency domain simplifies the description. In particular, the translational motion now corresponds to circular motion of the two (complex-valued) Fourier coefficients associated with the constituent sinusoids.

The motion is further simplified by a polar coordinate transform to extract phase and amplitude of each Fourier co-

efficient. Specifically, the motion is now along a straight trajectory, with both phases advancing linearly (but at different rates), and both amplitudes constant. Note that this is a geometric property that holds for any rigidly translating signal, and offers a simple means of predicting content over time. Indeed, we can use the shift property (see section 2.1) on $x(n, t+1) = x^v(n, t)$ and observe that prediction is now reduced to linear extrapolation of each coefficient's phase. We have the three step process:

$$\widetilde{x}(k,t) = \sum_{n=0}^{N-1} \phi^{-kn} x(n,t), \qquad \text{(analyze)}$$

$$\widetilde{x}(k,t+1) = \phi^{-kv} \widetilde{x}(k,t), \qquad \text{(advance phase)}$$

$$x(n,t+1) = \frac{1}{N} \sum_{k=0}^{N-1} \phi^{nk} \widetilde{x}(k,t+1). \qquad \text{(synthesize)}$$

Since we assumed that the motion from time t to $t+1$ is identical to that from time $t-1$ to t (ie. no acceleration), the phase advance kv can be computed from the past two representations as $kv = \angle\widetilde{x}(k,t) - \angle\widetilde{x}(k,t-1)$, where $\angle z$ indicates the phase of the complex number z. Thus, a polar coordinate transformation in the frequency domain converts translational motion into trajectories that are predictable via linear phase extrapolation.

2.3. Generalization: representing commutative Lie groups

Natural videos are replete with rich temporal transformations, such as continuous deformations of objects and surfaces. Assuming that these transformations can be described as groups, we will aim to learn their group representation from data. To this end, we seek a parameterization that generalizes beyond translation and the frequency domain. Remarkably, Fourier analysis can be seen as a special case of the representation theory of compact commutative Lie (ie. smooth) groups (Mackey, 1980).

In harmonic analysis, the celebrated Peter-Weyl Theorem (1927) establishes the completeness of the irreducible representations of any compact continuous group (an irreducible representation is a subspace that is invariant to group action and that can not be further decomposed). Furthermore, it follows that every compact Lie group admits a faithful (ie. injective) representation given by an explicit complete orthogonal basis, constructed from finite-dimensional irreducible representations (Hall, 2013). Accordingly, the action of a compact Lie group can be expressed as a rotation within each irreducible representation - thereby generalizing the Fourier shift property (an example is the construction of steerable filters (Freeman et al., 1991) in the computational vision literature).

In the case of compact commutative Lie groups, the irreducible representations are one-dimensional and complex valued: they are pairs of real valued basis functions. Therefore, the angular extrapolation mechanism described in the previous section (2.2) can be employed for prediction in a much wider setting than that of translational motion. We will rely on the parameterization suggested by the representation theory of compact commutative Lie groups to learn the harmonic basis functions of the transformations at play in natural videos.

3. Learning to predict with angular extrapolation

To generalize beyond translation and the Fourier transform, we aim to learn a representation of video frames that enables prediction via angular extrapolation. Specifically, we focus on next frame prediction, and optimize two parameterized mappings: one for the analysis and one for the synthesis transform. This framework is illustrated in Figure 2, which provides a generalization of the Fourier shift diagram (1).

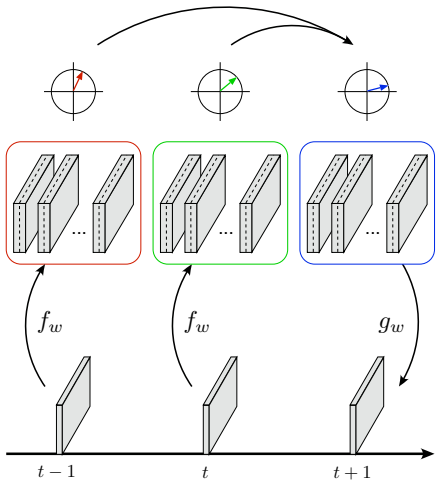


Figure 2. Unsupervised predictive representation learning framework. Each video frame is transformed using a parametric mapping f_w, to an internal representation consisting of pairs of coefficients arranged in spatial channels. Predictions of individual complex coefficients at time $t+1$ are computed by advancing the phase of the current coefficients by an amount equal to the phase advance over the interval from $t-1$ to t. At each time step, one such coefficient is depicted as a vector in two dimensions and the top arrows indicates how they are combined (each vector corresponds to the complex coefficient at a particular location within one channel pair). Predicted frames are then generated by applying the parameterized inverse mapping g_w on the advanced coefficients. Forward and inverse mappings are jointly trained to minimize mean squared prediction error between the predicted and actual frame at time $t+1$.

3.1. Architecture and objective function

When focusing on a small region in an image sequence, the transformation observed as time passes can be approximated as a *local* translation. That is to say, in a spatial neighborhood around position n, $m \in N(n)$, we have: $x(m, t+1) \approx x(m-v, t)$. We can use the decomposition described for global rigid translation, replacing the Fourier transform with a local convolutional operator (Fleet & Jepson, 1990), processing each spatial neighborhood of the image independently and in parallel, and applying angular extrapolation to the coefficients computed at each position.

We use the same weights for the encoding and decoding stages, that is to say the analysis operator is the transpose of the synthesis operator (also true of the Fourier transform and its inverse). Sharing these weights reduces the number of parameters and simplifies interpretation of the learned solution. This "polar predictor" (hereafter, **PP**) is consistent with the general scheme described in figure 2 where f_w is taken to be linear and convolutional, and g_w is its transpose. In practice, we assumed 64 convolutional channels with filters of size 17×17 pixels, with no additive constants.

At every position in the image (spatial indices are omitted for clarity of notation), each coefficient $y_j(t)$ is computed as an inner product between the input $x(t)$ and the filter weights w_j of each channel $j \in [0, 63]$: $y_j(t) = w_j^T x(t)$. In order to obtain phases, we combine coefficients in pairs, indexed by $k \in [0, 31]$, which can be written as single complex coefficient as: $z_k(t) = y_{2k}(t) + i y_{2k+1}(t) \in \mathbb{C}$, and expressed in polar coordinates as: $z_k(t) = a_k(t) e^{i\theta_k(t)}$. This polar coordinate transformation is the only non-linear step used in the PP architecture, and serves as a bivariate non-linear activation function, differing markedly from the typical (pointwise) rectification operations found in convolutional neural networks.

With this notation, linear phase extrapolation reduces to $\hat{z}_k(t+1) = a_k(t) e^{i(\theta_k(t) + \Delta\theta_k(t))}$, where the phase advance $\Delta\theta_k(t)$ is equal to the phase difference over the interval from $t-1$ to t: $\Delta\theta_k(t) = \theta_k(t) - \theta_k(t-1)$. The advanced coefficients can be written in a more compact form, using complex arithmetic, as:

$$\hat{z}_k(t+1) = \frac{z_k(t)^2 \overline{z_k(t-1)}}{|z_k(t)||z_k(t-1)|}, \qquad (2)$$

where $\overline{z}$ and $|z|$ respectively denote complex conjugation and complex modulus of z. This formulation in terms of complex coefficients has the benefit of handling phases implicitly, bypassing the discontinuities of phase unwrapping and the instability of angular variables (phase is unstable when amplitude is low). We find that such an indirect formulation of phase processing is necessary for the stability of training, as previously noted in the texture modeling literature (Portilla & Simoncelli, 2000). Finally, the estimated next frame is generated by applying the transposed convolution g_w (with the same weights as f_w) to the advanced coefficients.

As a more substantial generalization of polar prediction, we use deep convolutional neural networks to instantiate non-linear mappings for both the encoder f_w and the decoder g_w (each with independent filters). Specifically, the "deep polar predictor" (**deepPP**) operates by transforming two frames of input into the encoding space, $z(t-1) = f_w(x(t-1))$ and $z(t) = f_w(x(t))$, applying the polar prediction of equation 2 to this encoded representation, and then decoding the next frame from this prediction, $\hat{x}(t+1) = g_w(\hat{z}(t+1))$. While the PP model learns a linear representation, the deepPP model is nonlinear, with potential to enhance prediction by adapting to signal properties.

In order to isolate the effects of non-linearities from those of spatial scale, we chose the number of layers and the kernel sizes of deepPP so that the effective receptive field size was matched to that of the PP model. Specifically, both the encoder and the decoder are composed of 4 convolutional layers, each with 64 channels, and using filter kernels of size 5×5 followed by half-wave rectification (ReLU).

For both PP and deepPP models, convolutional kernels w are learned by minimizing the average squared prediction error:

$$\min_w \mathbb{E}_t \|x(t+1) - \hat{x}(t+1)\|_2^2.$$

The computation of this prediction error is restricted to the center of the image because moving content that enters from outside the video frame is inherently unpredictable. Specifically, we trim a 17-pixel strip from each side. Note that we only perform valid convolutions to avoid artificial interference with prediction (zero-padding creates undesirable boundary artifacts).

3.2. Comparison models

We compare our method to the traditional motion-compensated coding approach that forms the core of inter-picture coding in well established compression standards such as MPEG. Block matching is an essential component of these standards, allowing the compression of video content by up to three orders of magnitude with moderate loss of information. For each block in a frame, typical coders search for the most similar spatially displaced block in the previous frame (typically measured with MSE), and communicate the displacement coordinates to allow prediction of frame content by translating blocks of the (already transmitted) previous frame. We implemented a "diamond search" algorithm (Zhu & Ma, 2000) operating on blocks of 8×8 pixels, with a maximal search distance of 8 pixels which balances accuracy of motion estimates and speed of estimation (the search step is computationally intensive). We

use the estimated displacements to perform causal motion compensation (**cMC**), using displacement vectors estimated from the previous two observed frames (x_{t-1} and x_t) to predict the *next* frame (x_{t+1}) rather than the current one (as in MPEG).

To isolate the effects of the polar prediction, we also implemented a predictor using *linear extrapolation* of the responses of a deep neural network (**deepL**), with architecture identical to that of the deep polar predictor. That is to say, we replace equation 2 by: $\hat{y}_j(t+1) = 2y_j(t) - y_j(t-1)$, which amounts to enforcing linear dynamics in the latent space of the non-linear representation.

Finally, we implemented a more direct convolutional neural network predictor (**CNN**), that maps two successive observed frames to an estimate of the next frame (Mathieu et al., 2016). This predictor jointly transforms and predicts visual signals without explicitly partitioning spatial content representation and temporal feature extrapolation. For this, we used a CNN composed of 20 stages, each consisting of 64 channels, and computed with 3×3 filters without additive constants, followed by half-wave rectification. Note that, unlike all other predictors, this model jointly processes pairs of frames to generate predictions.

3.3. Datasets and training

To train, test and compare these models, we use the DAVIS dataset (Pont-Tuset et al., 2017), which was originally designed as a benchmark for video object segmentation. Image sequences in this dataset contain diverse motion of scenes and objects (eg., with fixed or moving camera, and objects moving at different speeds and directions), which make next frame prediction challenging. Each clip is sampled at 25 frames per second, and is approximately 3 seconds long. The set is subdivided into 60 training videos (4741 frames) and 30 test videos (2591 frames).

We pre-processed the data, converting all frames to monochrome luminance values, and scaling their range to the interval $[-1, 1]$. Frames are cropped to a 256×256 central region, where most of the motion tends to occur, and then spatially down-sampled to 128×128 pixels. We assume the temporal evolution of natural signals to be sufficiently and appropriately diverse for training, and do not apply any additional data augmentation procedures. We train on brief temporal segments containing 11 frames, which allows for prediction of 9 frames, processing these in batches of size 4. We train each model for one hundred epochs using the Adam optimizer (Kingma & Ba, 2015) with default parameters and a learning rate of $3 \cdot 10^{-4}$. The learning rate is halved at epochs 50, 60, 70, 80, 90, 100. We use batch normalization before every half-wave rectification, rescaling by the standard deviation of channel coefficients (but with no additive terms).

Similarly, we also trained on the larger UCF-101 dataset (Soomro et al., 2012). This dataset, initially designed for action recognition, contains about 2.5 million frames, which amounts to over 27 hours of video data. Note that, unlike the DAVIS dataset, the clips are only available in compressed video formats and may contain motion artifacts (due to inter-frame coding). We used the same pre-processing procedure, except that we reduced frames by directly cropping a 128×128 central region (without any down-sampling). We employ the same training procedure, except that we only run training for 25 epochs.

4. Unsupervised representation learning

4.1. Recovery analysis

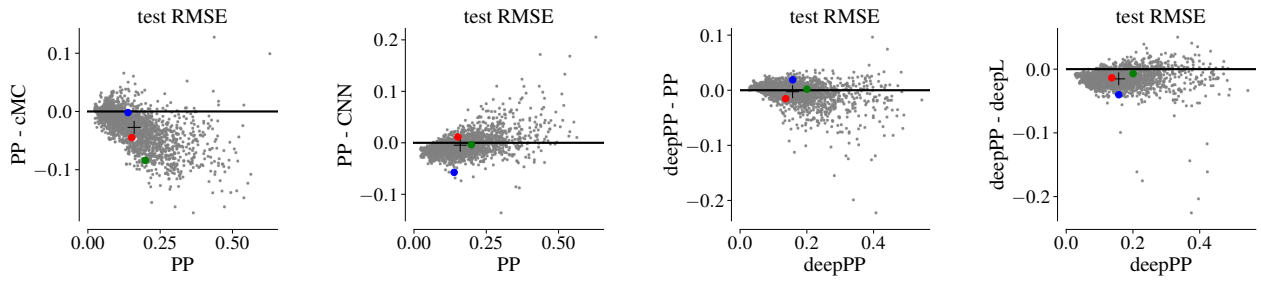
To experimentally validate our approach, we first verified that a the PP model can robustly recover known symmetries in small synthetic datasets consisting of translating or rotating image patches. For these experiments, we applied encoding and decoding transforms to the entire patch (i.e., non-convolutionally). Wen trained on translating image patches, the PP model learned approximately sinusoidal filters, shifted in phase by $\pi/2$ - i.e., a local Fourier transform. Similarly, when trained on rotating patches, the learned filters represented circular harmonics. We also found that PP extracts meaningful representations when multiple kinds of transformations are at play (eg. mixing both translations and rotations), and when the transformation are not perfectly translational (eg. translation with open boundary condition). Learned filters for each of these cases are provided in Figure 5 in the appendix.

4.2. Performance on Natural Videos

Table 1. Prediction error computed on the DAVIS and UCF-101 datasets. Values indicate average Mean Squared Error.

Algo.	DAVIS		UCF-101		# param.
	train	test	train	test	
Copy	0.064	0.065	0.0302	0.0286	
cMC	0.048	0.049	−−	0.0299	
deepL	0.034	0.037	0.0220	0.0217	665,856
CNN	0.031	0.035	**0.0210**	0.0215	666,496
PP	0.036	0.035	0.0245	0.0229	18,496
deepPP	**0.028**	**0.032**	0.0216	**0.0210**	665,856

We summarize the main prediction results in Table 1. First, observe that the predictive algorithms considered in this study perform significantly better than the baseline obtained by simply copying the last frame. Second, the polar predictor (PP) performs nearly as well as the convolutional neural network (CNN) in terms of test mean squared error on DAVIS. This demonstrates the remarkable power of the polar predictor: the PP model has roughly 30 times fewer



(a) (b) (c) (d)

Figure 3. Detailed performance comparison (in Root Mean Squared Error) of predictive algorithms. Each point corresponds to a frame in the test set. Vertical axes represent difference in performance: For points lying below the horizontal axis, the method whose performance is plotted on the horizontal axis achieves a lower RMSE than the comparison method. Black crosses indicate average RMSE. Red point corresponds to example in Figure 4, green to Figure 8 and blue to Figure 9.

parameters and uses a single non-linearity, while the CNN is composed of 20 non-linear layers. Finally, observe that deepPP achieves the lowest mean squared error, notably outperforming the deepL model which uses linear extrapolation in an otherwise identical architecture. It also outperforms the CNN on the UCF-101 test dataset while remaining significantly simpler. Thus, the prediction task benefits substantially from use of a fixed nonlinear transformation to polar coordinates.

While average performance values provide a compact summary, it is also informative to examine the distribution of prediction errors on individual frames from the test set. Figure 3 shows pairwise comparison of the predictive algorithms for each frame in the DAVIS dataset. To make the contrast more apparent, we display performance difference on the vertical axis. Note that while the models have been optimized to reduce mean squared error, we show root mean squared error (RMSE) in order to facilitate visual inspection of the results (the concavity of the square root spreads out small differences). We see that (a) The polar predictor representation systematically outperforms causal motion compensation, especially on difficult examples. (b) The polar predictor outperforms the CNN on the bulk of easy to medium cases but this advantage is reversed for harder examples. (c) The deep polar predictor outperforms the single layer polar predictor overall, indicating that the non-linearity in representation can help. (d) The deep polar predictor clearly outperforms the deep linear predictor, revealing the strong benefit of using a polar extrapolation mechanism over a linear one.

4.3. Learned filters

In order to better understand these results, we visualized the learned PP filters trained on the DAVIS dataset and observed that the learned filter are selective for orientation and spatial frequency, and that that they tile the frequency domain. Filters in each pair have a similar frequency preference, and are related by a 90 degrees phase shift (see in Figure 6a and 6b in the Appendix). This relationship is analogous to that of sines and cosines and is consistent with the structure of the angular extrapolation described in equation 2.

4.4. Examples

Consider a set of example videos, chosen to illustrate behaviors of the methods being compared. In Figure 4, we see a wall, its shadow and their sharp boundaries against a grass background as the camera moves. Both PP and deepPP generate good results, cMC produces a sharp prediction at the expense of significant blocking artifacts, and both the CNN and deepL tend toward excessive blurring. Here, the cMC is is significantly sharper than the others, but introduces substantial artifacts. Again, the PP methods produce sharper results than either the CNN or deepL methods. A few additional informative examples are displayed in the appendix (see Figure 8, 9, 10).

5. Related work

Our method is conceptually related to sparse coding with complex-valued coefficients (Cadieu & Olshausen, 2012) in that it factorizes natural videos into form and motion. But it differs in a number of important ways: (1) sparse coding focuses on representing, not predicting, the signal; (2) we do not promote sparsity of either amplitude or phase components; (3) finally, the discontinuity arising from selection of sparse subsets of coefficients seems at odds with the representation of continuous group actions, while our explicit mapping into polar coefficients aims for a smooth and continuous parameterization of the transformations that occur in natural videos. Several other studies have aimed to learn representations that decompose signal identity and attribute (ie. a *what-where*, or *invariance-equivariance* factorization).

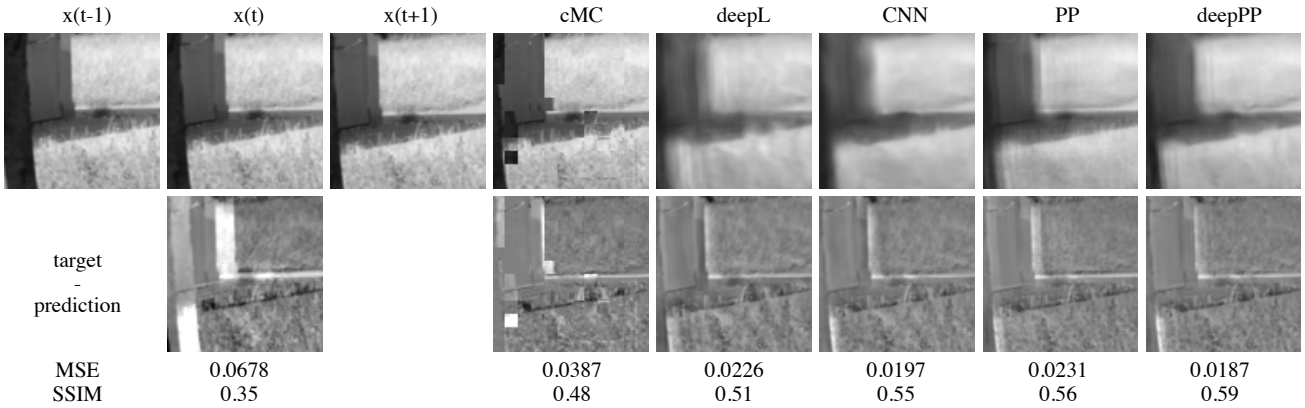


Figure 4. A typical example image sequence from the DAVIS test set. The first three frames on the top row display the unprocessed images, and last five frames show the respective prediction for each method (with their shorthand above). The bottom row displays error maps computed as the difference between the target image $x(t+1)$ and each predicted next frame on the corresponding position in the first row. Images, predictions and error maps are all shown on the same scale.

In particular learning linearized features from video was explored using a heuristic extrapolation mechanism (Goroshin et al., 2015). The authors developed specialized "soft max-pooling" and "soft argmax-pooling" modules and tested them on the small NORB dataset. A related approach aimed at finding video representations which decompose content and pose in order to enable prediction (Hsieh et al., 2018). This work explicitly identifies spatial components that are easier to predict in the moving MNIST and bouncing balls datasets. More sophisticated architectures have been developed to tackle the challenge of natural video prediction. In particular, a recurrent instantiation of the predictive coding theory (Rao & Ballard, 1999) introduced a stacked convolutional LSTM architecture (Lotter et al., 2017). In contrast, our framework scales to prediction of natural videos while remaining simple: we rely on principles of signal processing and representation theory to employ a polar non-linearity (and we describe an effective and stable implementation), but we do not explicitly model the stochastic nature of the video prediction problem.

Our method is also related to work that adopts a Lie group formalism in representation learning. Since the seminal work that proposed learning Lie group generators from dynamic signals (Rao & Ruderman, 1998), the polar parametrization was explored in (Cohen & Welling, 2014) to identify irreducible representations in a synthetic dataset. The continuous group formalism has also been combined with sparse coding (Chen et al., 2018; Chau et al., 2020) to model natural images as points on a latent manifold. More recently, bispectral neural networks (Sanborn et al., 2022) have been shown to learn image representations invariant to a given global transformation (in particular cyclic translation and rotation of MNIST digits). In contrast to the coding approach, our formulation relies on a prediction objective to jointly discover and exploit the symmetries implicit in data. In order to scale to natural video data, where multiple unknown and noisy transformations are play, we developed a convolutional approach that adapts to the local structure of transformations. This formulation can represent a very large family of local symmetries (including diffeomorphisms and non-smooth fields of local translations). This generality comes at the cost of precisely identifying what groups of transformations are captured by the learned representation.

Finally, in the fluid mechanics literature, the Koopman operator approach (Mezić, 2005) has been used to lift a system from its original state-space to a higher dimensional representation space where its dynamics can be linearized - a dynamical analog of the well known kernel trick. This formalism has spurred a line of work in machine learning that relies on autoencoders to learn coordinate systems that approximately linearize dynamics (Lusch et al., 2018; Azencot et al., 2020). In this perspective, our work can also be interpreted as learning the spectral properties of an abstract Koopman operator operating on video data, specifically estimating its complex eigenvectors. Our approach makes an inertial assumption and does not require an auxiliary network to compute velocities. Moreover it relies on a convolutional approach and is able to predict raw videos (which tend to contain richer structure than typical fluid flows).

6. Discussion

We have presented a simple self-supervised representation-learning framework based on next-frame prediction. It unveils the temporal structure of natural videos using local polar coordinates. Our approach jointly discovers and exploits the local symmetries present in the temporal evolution of image sequences, in particular the spatio-temporal redun-

dancies due to local deformation of image content. We assumed that spatial processing and temporal extrapolation can be partitioned into i) the learned parameterized mappings, one that extract pairs of local features from individual frames and one that generates a frame from the coefficients; and ii) a fixed angular extrapolation mechanism that advances coefficients and embodies an inertial hypothesis (ie. evolving content will continue to evolve in the same way). Our empirical results demonstrate that these assumptions, far from being too limiting, correspond well to the structure of natural videos and provide a natural representation thereof.

Specifically, we used the the polar coordinate transformation as a bivariate non-linear activation function acting on pairs of coefficients in the representation. Predictions in this representation were computed by phase advancement, which was implemented implicitly (Eq. 2). Compared to linear extrapolation, angular extrapolation achieved higher prediction accuracy on natural video. Using terminology from group theory, our polar models factorize signals into an invariant part, which is stable in time, and an equivariant part, which evolves linearly. This choice of prediction mechanism, motivated by principles of signal processing and harmonic analysis, acts as a structural prior. Although the conventional deep convolutional network (CNN) considered here could in principle have discovered this solution, it failed to do so (within the constraints of our architecture). The polar predictor, on the other hand, is well-matched to the task, and achieves a good solution using only a fraction of the number of parameters. It is optimized on a mean squared error objective, without any other additional regularization - which facilitates interpretability. This exemplifies a fundamental theme in computational vision and machine learning: when possible, let the representation do the analysis.

Our approach to prediction has the advantage of being motion-informed while not relying on explicit motion estimation. Because it is not constrained to assigning a single motion vector at every location and instead represents a distribution of phases, this method bypasses known difficulties of motion estimation in handling non-translational motions and outperforms a conventional causal motion compensated algorithm. In the era of GPU computing, it admits a very fast implementation that has potential for applications in video compression. Moreover, the polar predictor takes the form of a predictive auto-encoder that associates a latent representation vector to each frame. This representation may prove useful for other tasks like object categorization, segmentation, or estimation of heading direction for a moving observer. Several natural extensions of the work presented here can be further explored: (i) treating the angular extrapolation prediction mechanism as a more general building block that is cascaded in a multi-layer architecture; (ii) optimizing representation layers deeper in the hierarchy to make predictions at longer timescales; (iii) measuring prediction error directly in the representation domain, while avoiding representation collapse - such a local objective function would allow a potential connection with biological neural architecture and to human visual perception. (iv) examining and interpreting what is learned in the deepPP model, especially around occlusion boundaries (which is not invertible, and therefore not a group action).

References

Azencot, O., Erichson, N. B., Lin, V., and Mahoney, M. Forecasting sequential data using consistent koopman autoencoders. In *International Conference on Machine Learning*, 2020.

Cadieu, C. F. and Olshausen, B. A. Learning intermediate-level representations of form and motion from natural movies. *Neural computation*, 2012.

Chau, H. Y., Qiu, F., Chen, Y., and Olshausen, B. Disentangling images with lie group transformations and sparse coding. *arXiv preprint arXiv:2012.12071*, 2020.

Chen, Y., Paiton, D., and Olshausen, B. The sparse manifold transform. *Advances in neural information processing systems*, 31, 2018.

Cohen, T. and Welling, M. Learning the irreducible representations of commutative lie groups. In *International Conference on Machine Learning*. PMLR, 2014.

Fleet, D. J. and Jepson, A. D. Computation of component image velocity from local phase information. *International journal of computer vision*, 1990.

Földiák, P. Learning invariance from transformation sequences. *Neural Computation*, 1991.

Freeman, W. T., Adelson, E. H., et al. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 1991.

Goroshin, R., Mathieu, M. F., and LeCun, Y. Learning to linearize under uncertainty. In *Advances in Neural Information Processing Systems*, 2015.

Hall, B. C. Lie groups, lie algebras, and representations. In *Quantum Theory for Mathematicians*. Springer, 2013.

Hénaff, O. J., Goris, R. L., and Simoncelli, E. P. Perceptual straightening of natural videos. *Nature neuroscience*, 2019.

Hsieh, J.-T., Liu, B., Huang, D.-A., Fei-Fei, L. F., and Niebles, J. C. Learning to decompose and disentangle representations for video prediction. *Advances in neural information processing systems*, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *ICLR*, 2015.

Lotter, W., Kreiman, G., and Cox, D. Deep predictive coding networks for video prediction and unsupervised learning. *International Conference on Learning Representations*, 2017.

Lusch, B., Kutz, J. N., and Brunton, S. L. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 2018.

Mackey, G. W. Harmonic analysis as the exploitation of symmetry–a historical survey. *Bulletin (New Series) of the American Mathematical Society*, 3(1. P1):543–698, 1980.

Mathieu, M., Couprie, C., and LeCun, Y. Deep multi-scale video prediction beyond mean square error. *ICLR*, 2016.

Mezić, I. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 2005.

Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., and Van Gool, L. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017.

Portilla, J. and Simoncelli, E. P. A parametric texture model based on joint statistics of complex wavelet coefficients. *International journal of computer vision*, 2000.

Rao, R. and Ruderman, D. Learning lie groups for invariant visual perception. *Advances in neural information processing systems*, 1998.

Rao, R. P. and Ballard, D. H. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 1999.

Sanborn, S., Shewmake, C., Olshausen, B., and Hillar, C. Bispectral neural networks. *arXiv preprint arXiv:2209.03416*, 2022.

Soomro, K., Zamir, A. R., and Shah, M. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

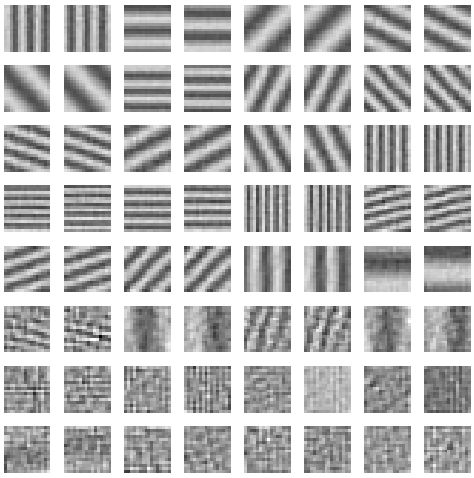
Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 2004.

Wiegand, T., Sullivan, G. J., Bjontegaard, G., and Luthra, A. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 2003.

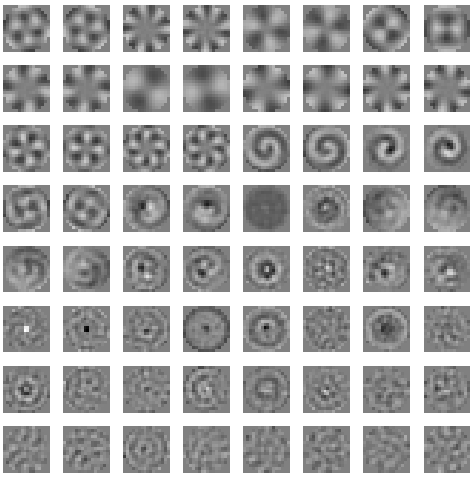
Wiskott, L. and Sejnowski, T. J. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 2002.

Zhu, S. and Ma, K.-K. A new diamond search algorithm for fast block-matching motion estimation. *IEEE transactions on Image Processing*, 2000.

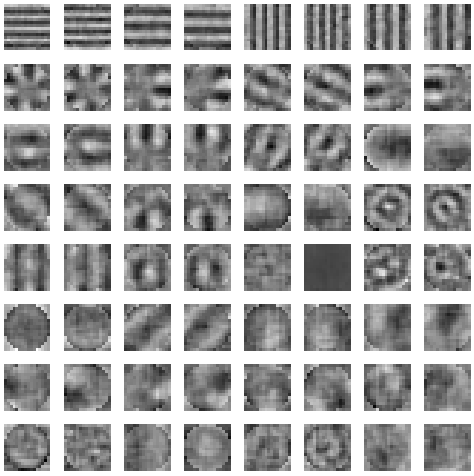
A. Appendix



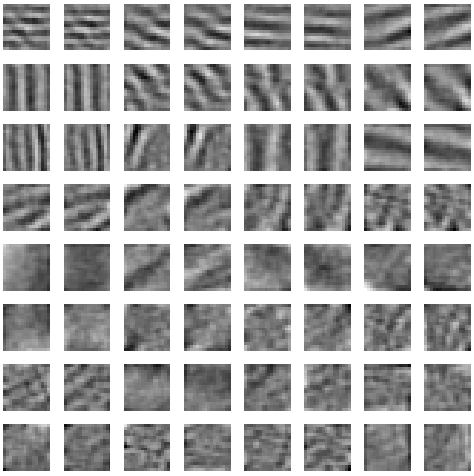
(a) Translation (cyclic boundary)



(b) Rotation

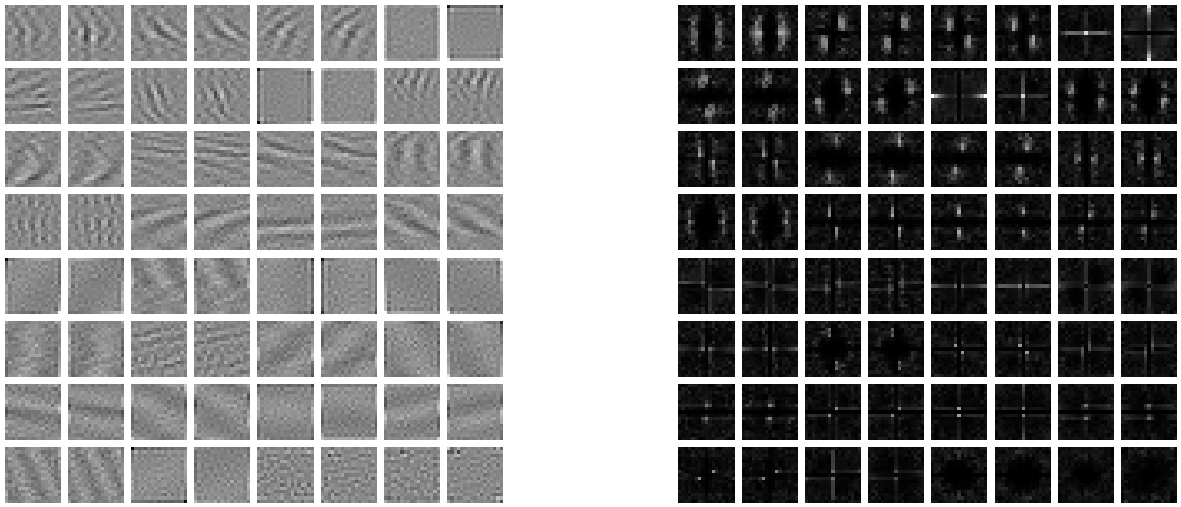


(c) Translation and rotation



(d) Translation (open boundary)

Figure 5. Filters of polar predictor networks trained to predict small synthetic sequences. We randomly select 100 image patches of size 16×16 from the DAVIS dataset and generate training data by manually transforming them - applying translations or rotations. We verify that PP recovers the known harmonic functions: Fourier modes for translation (panel a), and disk harmonics for rotation (panel b). To show that the recovery of harmonics is robust, we design two additional synthetic datasets. i) the combination of translational and rotational sequences. In this case, PP learns filters that correspond to either group, suggesting that our approach can generalize to situations with more than than one group at play (panel c); ii) generalized translation sequences: spatially sliding a square window on a large image (ie. new content creeps in and falls off at boundaries), instead of using cyclic boundary condition (ie. content wraps around the edges). In this case, PP learns localized Fourier-like modes (panel d), indicating that approximate group actions still provide meaningful training signal - although it is much more noisy. In each panel, the 32 pairs of filters are sorted by their norm. Notice that some of the filters are not structured and generally miss high frequency harmonics. This is due to the spectral properties of the datasets, which have more power at lower frequencies, and to the discretization of the transformations.



(a) spatial domain filters

(b) corresponding Fourier amplitude spectra

Figure 6. Filters of a polar predictor trained to predict natural videos from the DAVIS dataset. The 32 pairs of convolutional filters are sorted by their norm and their amplitude spectrum is displayed at corresponding locations on the right panel. Observe that the filters are selective for orientation and spatial frequency, tile the frequency spectrum, and form quadrature pairs. Notice that some of the learned filters do not exactly conform to the idealized description just given.

Table 2. Prediction error computed on the DAVIS dataset. Values indicate average PSNR and SSIM. All methods obtain relatively low structural similarity scores (Wang et al., 2004), a perceptual measure of similarity that equals one when both images are identical. This indicates that prediction on this dataset is quite challenging.

		Algorithm					
metric	set	Copy	cMC	deepL	CNN	PP	deepPP
PSNR ↑	train	21.32	23.82	23.18	23.84	24.49	**24.52**
	test	20.06	22.37	22.30	22.82	**23.46**	23.35
SSIM ↑	train	0.52	0.64	0.58	0.62	**0.65**	0.64
	test	0.50	0.62	0.55	0.59	**0.63**	0.61

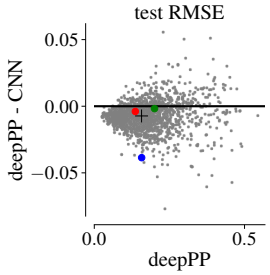


Figure 7. Another comparison - see caption of Fig. 3. The deep polar predictor generally outperforms the CNN over the whole range of difficulties.

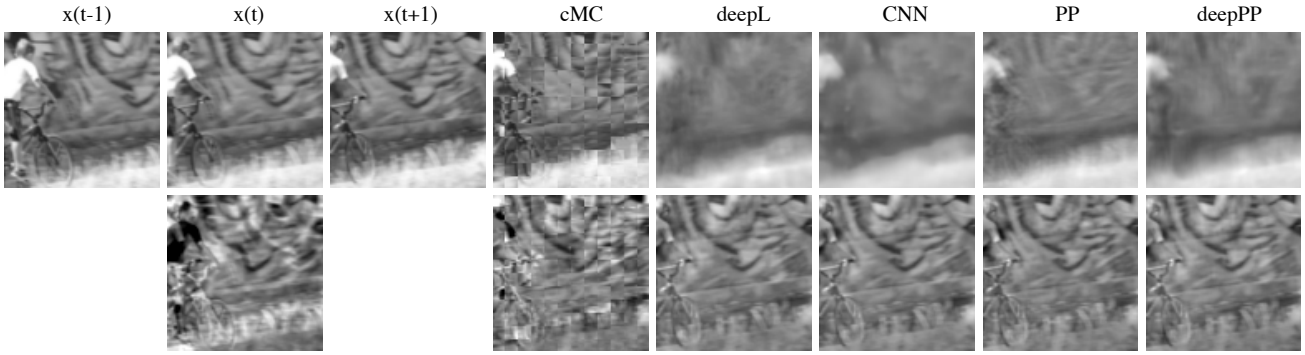


Figure 8. Another example image sequence with nonrigid motions/deformations - see caption of Fig. 4. As the biker advances to the right, the camera tracks and leads its displacement.

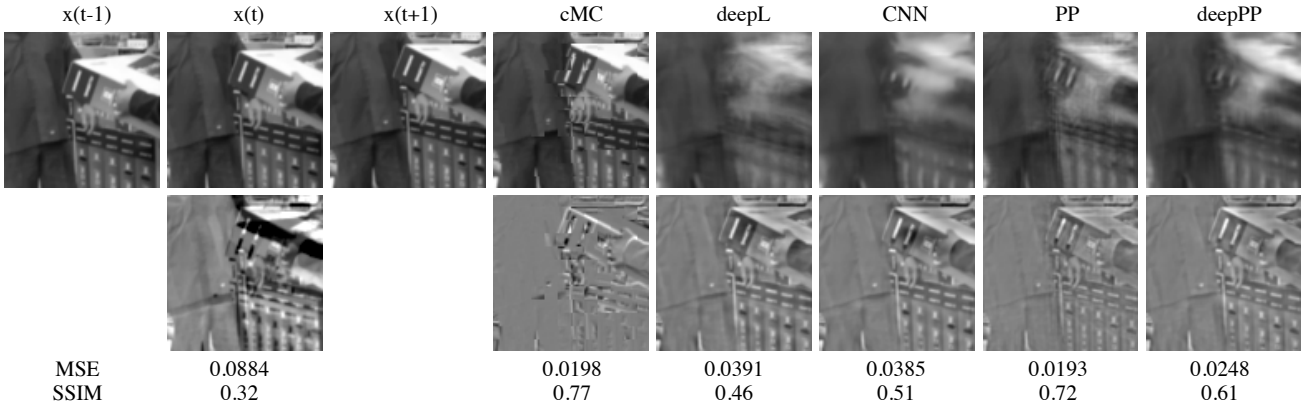


Figure 9. Another example image sequence - see caption of Fig. 4. A video with one portion not moving. In these regions the cMC is most effective, and PP and deepPP outperform CNN and deepL.

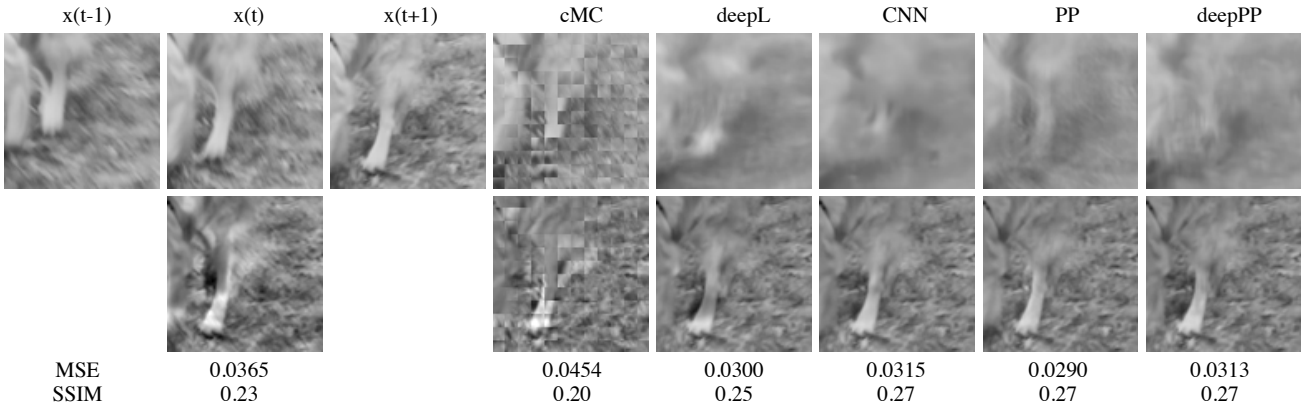


Figure 10. Another image sequence - see caption of Fig. 4. As expected, in presence of large non-rigid motion, with texture and object deformations, prediction is very difficult and none of the tested methods performs well.