A Description of architectures

Motion Compensation 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 [29] 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).

Complex Steerable Pyramid We consider a fixed multiscale oriented representation of image content: a steerable pyramid [11, 10] covering 16 orientations and 5 scales on the DAVIS dataset (resp. 16 orientations and 4 scales on VanHateren dataset). This choice of number of orientations and number of sacles maximizes prediction performance on the corresponding datasets.

Polar Predictor We use 16 pairs of convolutional channels with filters of size 17×17 pixels, without biases (no additive constants). For the multiscale version, the representation is computed inside a fixed Laplacian pyramid [9]. We used 4 scales for the DAVIS dataset (and respectively 4 scales for the VANH dataset). Within this multiscale representation, the learned filters are applied with zero padding (ie. "same" boundary condition).

Quadratic Predictor We consider a multiscale architecture (cf. Polar Predictor) with 16 groups of 4 convolutional filters for the analysis (f_w) and synthesis (g_w) mappings. The quadratic predictor (p_w) operates on groups of 4 coefficients and contains 12 quadratic units. It is therefore more expressive than the Polar Predictor architecture and contains phase advance as a special case.

Vanilla CNN Finally, we implemented a more direct convolutional neural network predictor (**CNN**), that maps two successive observed frames to an estimate of the next frame [12]. 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. To facilitate learning, a skip connection copies the current frame $x(t)$ and the network only outputs residuals that get added to the current frame in order to predict the next frame: $\hat{x}(t+1) = x(t) + f_w([x(t), x(t-1)]$. This model jointly transforms and processes pairs of frames to generate predictions, while both polar predictor (PP) and quadratic predictor (QP) separate spatial processing and temporal extrapolation.

B Description of datasets and optimization

DAVIS To train, test and compare these models, we use the DAVIS dataset [14], 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.

VanHateren We also consider a smaller dataset of natural image sequences obtained from Hans van Hateren [13], as described in [3] and downloaded from https://github.com/cadieu/twolayer. The top missing band of the images is cropped from 128 by 128 pixels to 112 by 128 pixels. The dataset is standardized to zero mean and unit variance, and it is split into 292 snippets of 11 frames for training and 33 snippets of 11 frames each for testing. There is no spatial downsampling or whitening.

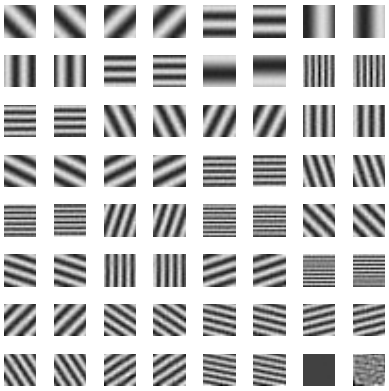
Boundary handling The computation of this prediction error is restricted to the center of the image because moving content that enters from outside the video frame is inherently unpredictable. Specifically, we trim a 17-pixel strip from each side, yielding frames of size 94×94 pixels.

Training procedure 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 200 epochs on DAVIS using the Adam optimizer [30] with default parameters and a learning rate of $3 \cdot 10^{-4}$. The learning rate is automatically halved when the test loss plateaus. In the CNN, we use batch normalization before every half-wave rectification, rescaling by the standard deviation of channel coefficients (but with no additive bias terms).

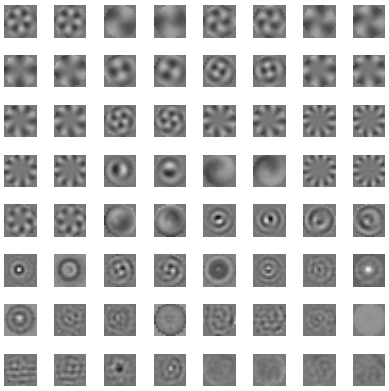
C Planted symmetries

D Learned filters

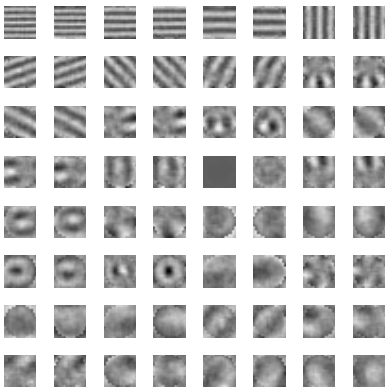
E Predictions examples



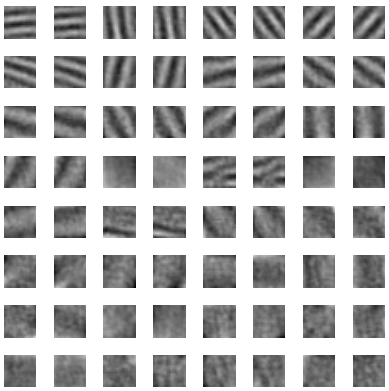
(a) Translation (cyclic boundary)



(b) Rotation

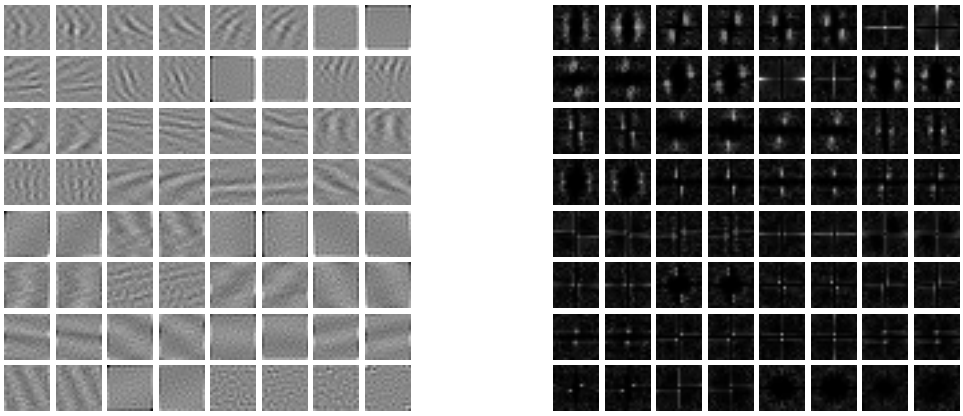


(c) Translation and rotation



(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 some filters are less structured. 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.

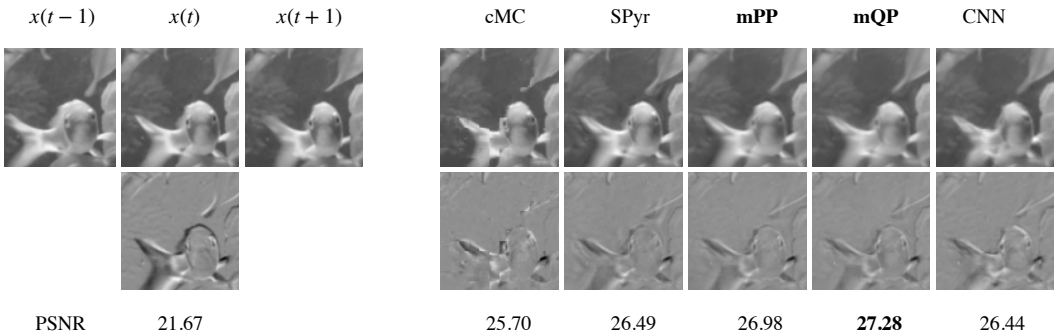


Figure 7: A typical example image sequence from the DAVIS test set. The first three frames on the top row display the unprocessed images, and last five frames show the respective prediction for each method (with their shorthand above). The bottom row displays error maps computed as the difference between 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.