MMVP: Motion-Matrix-based Video Prediction

Yiqi Zhong[1*†] Luming Liang[2*‡] Ilya Zharkov[2] Ulrich Neumann[1‡]
[1]University of Southern California [2]Microsoft
[1]{yiqizhon, uneumann}@usc.edu [2]{lulian,zharkov}@microsoft.com

Abstract

A central challenge of video prediction lies where the system has to reason the objects' future motions from image frames while simultaneously maintaining the consistency of their appearances across frames. This work introduces an end-to-end trainable two-stream video prediction framework, Motion-Matrix-based Video Prediction (MMVP), to tackle this challenge. Unlike previous methods that usually handle motion prediction and appearance maintenance within the same set of modules, MMVP decouples motion and appearance information by constructing appearance-agnostic motion matrices. The motion matrices represent the temporal similarity of each and every pair of feature patches in the input frames, and are the sole input of the motion prediction module in MMVP. This design improves video prediction in both accuracy and efficiency, and reduces the model size. Results of extensive experiments demonstrate that MMVP outperforms state-of-the-art systems on public data sets by non-negligible large margins (≈ 1 db in PSNR, UCF Sports) in significantly smaller model sizes (84% the size or smaller). Please refer to this link for the official code and the datasets used in this paper.

1. Introduction

Video prediction aims at predicting future frames from limited past frames. It is a longstanding yet unsolved problem studied for decades [13, 3]. Advancing research in this area benefits various applications such as video compression [29, 53, 28], surveillance systems [57, 59, 7], and robotics [9, 16, 8]. The task can be essentially broken down into two sub-tasks: i) motion prediction and ii) frame synthesis. Each sub-task has its unique goal that cannot be simply accomplished by achieving the other one. For the sub-task of *motion prediction*, systems need to reason the future movements of objects/backgrounds by discovering the motion cues hidden in the past frames. Whereas for the sub-task of *frame synthesis*, systems need to maintain appearance features and generate future frames that keep appearance consistency. These separated goals make video prediction inherently much more difficult than the individual task of normal motion prediction or content synthesis.

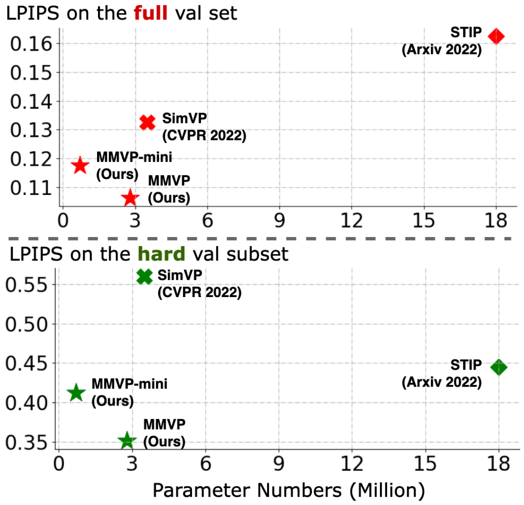

Figure 1: Performance comparison on UCF Sports with STIP [4] and SimVP [12]. The *hard* subset contains samples where SSIM between the last observed frame and the first future frame is smaller than 0.6, which indicates drastic motion patterns (data is from Tables 2 and 6).

Most existing works in video prediction are based on a single-stream pipeline that conducts motion prediction and appearance feature extraction for frame synthesis within the same set of modules. Their systems [47, 45, 46, 55, 5, 4] usually grow out of advanced network structures for sequential data analysis, such as recurrent neural networks (RNNs) [31] and transformers [42]. One shared characteristic of those methods is that they show excellent capabilities in capturing complex motion patterns but lower capabilities in appearance maintenance, yielding "correct" but not "good" synthesized frames. The reason behind this is that those methods usually contain complicated spatial-temporal feature extraction and state transition operations, which can cause unavoidable appearance information loss.

[*] Equal contributions.
[†] The work is done during Yiqi Zhong's internship at Microsoft.
[‡] Corresponding authors.

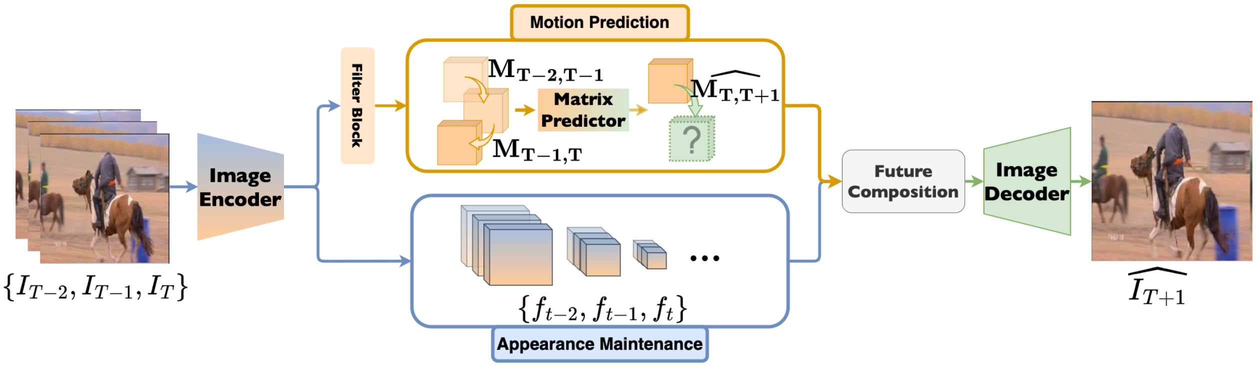

Figure 2: MMVP is a two-stream video prediction framework. It decouples motion prediction and appearance maintenance, and it reunites motion and appearance features through feature composition operation.

Researchers have proposed several solutions to mitigate appearance information loss; the most common approaches are introducing sophisticated appearance-aware state transition unit [5, 4, 54], or adding frequent feature shortcuts from previous frames [12, 46, 47, 45]. However, the former solution tends to build cumbersome models with a huge number of parameters; and for the latter solution, too much residual information from previous frames can cause a larger performance drop for hard cases such as videos with fast movements and/or moving cameras. Figure 1 shows the comparison between STIP [4] (an example of the former solution) and SimVP [12] (an example of the latter solution).

To avoid running into possible trade-offs between motion and appearance in single-stream pipelines, a few works have explored two-stream pipelines [27, 2, 10, 43, 11], decoupling motion prediction and appearance maintenance. However, they either require auxiliary sub-networks such as optical flow estimator [27, 52] and key point detector [11] to generate motion representations, which complicates video prediction and reduces the generalizability of systems; or they do not provide an efficient solution to reunite the predicted motion and the appearance features [6, 43].

With these gaps in current research, we introduce a novel two-stream, end-to-end trainable framework for video prediction: Motion-Matrix-based Video Prediction (MMVP) (see Figure 2 for the framework overview). As the name indicates, MMVP uses motion matrices as the decoupled motion representation of video frames. The motion matrix is a 4D matrix representing the image feature patches of consecutive frames (see Figure 3 I). As motion matrices are the sole input of the matrix predictor (i.e., the motion prediction module in MMVP), MMVP specifies the hidden motion information and makes the matrix predictor only focus on motion-related information. For the reunion of motion and appearance features, MMVP gets inspiration from the image autoencoder. It first embeds the past frames individually through an image encoder. Then it composes the embedding of the future frames using the predicted motion matrices output by the matrix predictor and the past frames' embeddings through matrix multiplication (see Figure 3 II). Then, an image decoder decodes the composed embeddings into the predicted future frames.

The advantage of MMVP is three-fold: (i) MMVP decouples motion and appearance by constructing motion matrices, requiring no extra construction modules; (ii) unlike optical flow that describes the one-to-one relationship between pixels, motion matrices describe the many-to-many relationship between feature patches, and are more flexible and applicable for real-world data; (iii) MMVP reunites the appearance and motion prediction results through matrix multiplication, which is interpretable and of little information loss. The advantages make MMVP a much more compact model with significantly fewer parameters yet still matching SOTA methods in performance. We validate MMVP on three datasets, UCF Sports [36], KTH [38], and MovingMNIST [41]. Experiments show that MMVP matches or surpasses SOTA methods on all three datasets across metrics. Specifically, compared to STIP [4], MMVP uses **84%** fewer parameters (18M vs. 2.8M) but achieves **38%** better performance in the LPIPS metric (12.73 vs. 7.88) on the UCF Sports dataset (Table 4).

2. Related Works

Good video prediction systems should not only accurately reason the future motions of the objects but also maintain their appearances and synthesize consistent future frames. As the video resolution has become increasingly higher today, researchers should additionally consider the scalability and efficiency of video prediction systems [33].

Most video prediction works to date adopt a single-stream pipeline that grows out of advanced techniques in sequential data analysis. Those techniques usually contain sophisticated spatial-temporal feature extraction operations and state transitions, which result in appearance information loss. Thus, researchers tend to modify the techniques to let them attend more to appearance mainte-

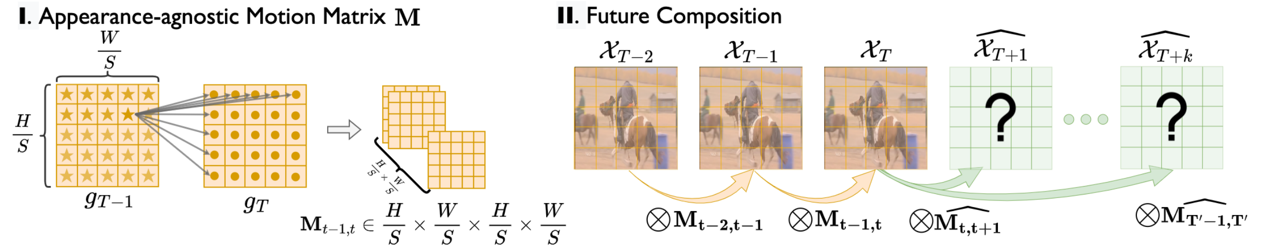
I. Appearance-agnostic Motion Matrix $\mathbf{M}$
$\frac{W}{S}$
$\frac{H}{S}$
g_{T-1}
g_T
$\frac{H}{S} \times \frac{W}{S}$
$\mathbf{M}_{t-1,t} \in \frac{H}{S} \times \frac{W}{S} \times \frac{H}{S} \times \frac{W}{S}$
II. Future Composition
$\mathcal{X}_{T-2}$
$\mathcal{X}_{T-1}$
$\mathcal{X}_T$
$\widehat{\mathcal{X}_{T+1}}$
$\widehat{\mathcal{X}_{T+k}}$
?
?
$\otimes \mathbf{M_{t-2,t-1}}$
$\otimes \mathbf{M_{t-1,t}}$
$\otimes \widehat{\mathbf{M_{t,t+1}}}$
$\otimes \widehat{\mathbf{M_{T'-1,T'}}}$

Figure 3: MMVP relies on the motion matrix $\mathbf{M}$. $\mathbf{M}$ is an appearance-agnostic motion representation that measures the cosine similarity between the feature patches of two consecutive frames. Information of the past frames $\{\mathcal{X}_0, ..., \mathcal{X}_T\}$ can compose the future information $\{\widehat{\mathcal{X}}_{T+1}, ..., \widehat{\mathcal{X}_{T'}}\}$ through matrix multiplication $\otimes$ with $\mathbf{M}$.

nance [47, 40, 45, 5, 4, 46, 18]. Based on ConvLSTM [40], PredRNN [47] propose a novel memory state transition method between the recurrent nodes to help the appearance features from observations be delivered to the final prediction. E3D-LSTM [46] integrated 3D convolution operations to help emphasize the short-term appearance. CrevNet [23] uses a conditionally reversible network to preserve the information from the past frames. More recently, the progress in temporal modeling made by transformer-based methods [42] has drawn more attention. The field starts to see works that use transformers in video prediction tasks [15, 50, 35]. These works have a better information aggregation capability and less appearance information loss, but they usually have a huge number of parameters and lack scalability to higher-resolution videos.

Seeking an alternative solution to video prediction and overcoming the drawbacks of the aforementioned methods, a few works have also explored the potential of decoupling motion and appearance and building a two-stream pipeline. Using optical flow is an intuitive idea. Several previous works [27, 2, 52, 10] use built-in or off-the-shell systems to predict the optical flow from the past to future frames; they use the predicted optical flow to warp past frames into final results. The drawbacks of this line of work are two-fold: i) time-consuming, especially for the warping procedure of high-resolution video sequences; ii) more importantly, the one-to-one relationship defined by the optical flow may not be applicable for some cases such as when one pixel or superpixel moves half of the pixel size, or when one pixel has impacts on several pixels in the next frames. Besides optical flow, pose [43] and keypoint [11] are also used to represent the motion of the objects in the scene for video prediction. However, this approach requires extra pose or keypoint detectors to process videos. For videos with multiple objects, more complex motions, and higher resolution, the efficiency of these extra modules cannot be guaranteed. Alternatively, MCNet [6] uses the difference between frames to represent motion information. It does not require extra modules, but using concatenation to reunite motion and appearance features is inefficient.

Compared to the small number of two-stream video prediction systems above, our MMVP has three advantages: i) its motion matrix is able to describe the many-to-many relationship of the pixels or super-pixels, which makes MMVP much more flexible and reasonable for real-world data; ii) MMVP does not require an extra module to produce motion representation; and iii) the reunion of motion and appearance features is more intuitive and interpretable than the approaches used in previous works. This paper will demonstrate how MMVP makes a more capable and compact video prediction framework.

3. Motion-Matrix-based Video Prediction

3.1. Framework Overview

Given a video sequence $\mathcal{I} = \{I_t\}_{t=1}^{T}$, where I_t denotes the tth frame, usually in the RGB format, MMVP estimates the future T' frames $\widehat{I_{T+1}}$ to $\widehat{I_{T+T'}}$ from I_1 to I_T. In comparison to the known frame set $\mathcal{I}$, we denote the estimated frame set as $\mathcal{I}' = \{I'_t\}_{t=T+1}^{T+T'}$. The training of the framework is solely supervised by mean-squared error (MSE) loss. MMVP consists of three steps as follows:

Step 1: Spatial feature extraction. We use an RRDBs [44] based network architecture to model an image encoding function Ω, defined in eq. 1. This image encoder embeds each frame $I_i, i \in \{1, 2, ..., T\}$ in a down-sample hidden space separately and outputs their corresponding hidden features f_i. In MMVP, besides the image encoder, we have an extra filter block that models a filtering function Θ defined in eq. 2. It takes f_i as the input and aims to generate g_i. As the module name indicates, the filter block aims to filter out motion-irrelevant features of f_i for motion matrix construction. We will introduce the details in Sec. 3.2.

Step 2: Motion matrix construction and prediction. MMVP generates a set of motion matrices $\{\mathbf{M}_{i,i+1}\}, i \in \{1, 2, .., T-1\}$ for every two consecutive frames based on their feature pairs $\{g_i, g_{i+1}\}$. Then, a matrix predicting function Φ defined in eq. 4 takes $\{\mathbf{M}_{i,i+1}\}$ as the input and predicts future matrices $\{\widehat{\mathbf{M}_{T,T+j}}\}, j \in \{T, ..., T+T'\}$. Sec. 3.3 will elaborate on the definition and construction procedures of the motion matrices, and Sec. 3.4 will demonstrate the inner structure of the matrix predictor.

Step 3: Future composing and decoding. Using the output of steps 1 and 2, MMVP composes the unknown information for future frames. Then, a future decoding module takes the composed features as the input and outputs the final prediction. We will introduce the feature composition procedure in Sec. 3.5 and the future decoding module in Sec. 3.6.

3.2. Spatial Feature Extraction

Spatial feature extraction involves two components of the MMVP framework: image encoder, and filter block.

The image encoder Ω in MMVP encodes every I_i from the input data sequence to their corresponding features f_i individually. The filter block Θ subsequently processes f_i and makes it ready for the construction of a motion matrix. Formally,

$$f_i = \Omega(I_i), \tag{1}$$

$$g_i = \Theta(f_i), \tag{2}$$

where $i \in \{1, 2, ..., T\}$. We use a convolutional network with residual in residual dense blocks (RRDBs) [44] to implement the image encoder, and we use a two-layer convolutional network to implement the filter block. See the detailed network architecture in Appendix.

Next, the output features of the image encoder will take participate in the future feature composition, while the output of the filter block is only used for motion matrices construction. The existence of the filter block helps the model to filter out irrelevant features from the image encoder output and allows the construction of motion matrices to focus more on motion-related features. See Table 2 for the ablation study about the filter block.

3.3. Motion Matrix Construction

Given the output of the filter block $g_i \in \mathbb{R}^{\frac{H}{S} \times \frac{W}{S}}, i \in \{1, 2, ..., T\}$, we denote the feature patch at $(h, w), w \in \{0, 1, 2, \frac{W}{S} - 1\}, h \in \{0, 1, 2, \frac{H}{S} - 1\}$ as $g_i^{h,w}$, where H and W are the input images' height and width, and S is the downsampling ratio to the original image; thus $\frac{H}{S}$ and $\frac{W}{S}$ are respectively the feature map's height and width.

For two consecutive frames' feature $\{g_i, g_{i+1}\}$, we calculate the cosine similarity for each and every pair of feature patches to construct a 4D motion matrix $\mathbf{M}_{i,i+1} \in \mathbb{R}^{\frac{H}{S} \times \frac{W}{S} \times \frac{H}{S} \times \frac{W}{S}}$. We denote the element of the matrix $\mathbf{M}_{i,i+1}$ at $(h_i, w_i, h_{i+1}, w_{i+1})$ as $\mathbf{M}_{i,i+1}^{h_i,w_i,h_{i+1},w_{i+1}}$, and let

$$\mathbf{M}_{i,i+1}^{h_i,w_i,h_{i+1},w_{i+1}} = D_c(g_i^{h_i,w_i}, g_{i+1}^{h_{i+1},w_{i+1}}). \tag{3}$$

In the equation, D_c is the cosine similarity, $g_i^{h_i,w_i}$ is the feature patch of frame i with the index of (h_i, w_i), and $g_{i+1}^{h_{i+1},w_{i+1}}$) is the feature patch of frame $i+1$ with the index of (h_{i+1}, w_{i+1}). With a feature patch at (h_i, w_i) of g_i as an example, the matrix $\mathbf{M}_{i,i+1}^{h_i,w_i} \in \mathbb{R}^{H \times W}$ can be regarded as a heatmap that reflects how much impact $g_i^{h_i,w_i}$ has on g_{i+1}, or more intuitively, the motion tendency of $g_i^{h_i,w_i}$ (see Figure 5 for illustration). It is similar to the definition of optical flow [17], which determines the movement of a pixel or superpixel. The difference between optical flow and motion matrix is that motion matrix does not strictly define a one-to-one relationship between the pixels or superpixels of the current frame and those of the next frame. It is a more practical assumption in the video prediction task that one current feature patch may influence several future feature patches.

3.4. Matrix Prediction

Given the motion matrices of the past T frames $\{\mathbf{M}_{1,2}, \mathbf{M}_{2,3}, ..., \mathbf{M}_{T-1,T}\}$, the matrix predictor Ψ predicts the future matrices $\{\mathbf{M}_{T,T+1}, \mathbf{M}_{T,T+2}, ..., \mathbf{M}_{T,T'}\}$. Instead of predicting the motion matrices between the consecutive frames, we predict the ones between the last observed frame I_T and every future frame $I_{T+j}, j \in \{1, 2, ..., T'\}$, as

$$\{\widehat{\mathbf{M}_{T,T+j}^{w,h}}\} = \Psi(\{\mathbf{M}_{i,i+1}^{w,h}\}), \forall i \in \{1, 2, ..., T-1\}, \tag{4}$$

where $\mathbf{M}_{i,i+1}^{w,h} \in \mathbb{R}^{H \times W}$, $w \in \{1, 2, ..., \frac{W}{S}\}$ and $h \in \{1, 2, ..., \frac{H}{S}\}$. This design aims to reduce the accumulative error during feature composition and is validated by the long-term prediction setting shown in Table 3. Since for this work, we focus on testing the function and performance of the MMVP framework, we report the use of a simple *3D fully convolutional* architecture to implement Ψ. Fellow researchers can choose to easily replace the implementation of Ψ with more advanced temporal modules if they wish to pursue better performances.

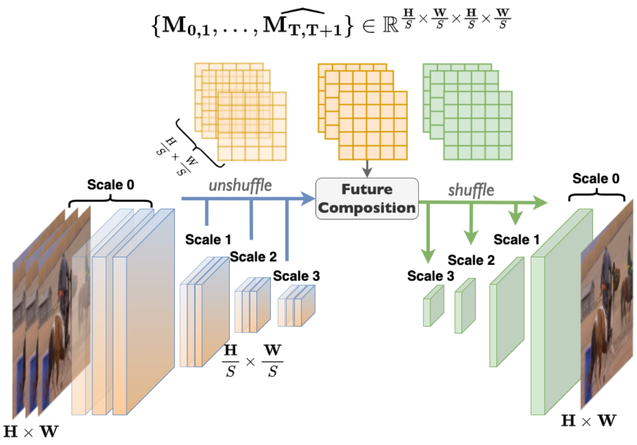
$\{\mathbf{M_{0,1}}, \ldots, \widehat{\mathbf{M_{T,T+1}}}\} \in \mathbb{R}^{\frac{\mathbf{H}}{S} \times \frac{\mathbf{W}}{S} \times \frac{\mathbf{H}}{S} \times \frac{\mathbf{W}}{S}}$
$\frac{H}{S} \times \frac{W}{S}$
Scale 0
unshuffle
Future Composition
shuffle
Scale 0
Scale 1
Scale 2
Scale 3
Scale 1
Scale 2
Scale 3
$\frac{\mathbf{H}}{S} \times \frac{\mathbf{W}}{S}$
H × W
H × W

Figure 4: All scales of image features and the original frames can join the future composition with predicted motion matrices through pixel unshuffle and shuffle operations.

3.5. Multi-scale Future Composition

The future composition step generates information for the future frames using the observed information and the motion matrices. It is formulated as

$$\widehat{\mathcal{X}_{T+j}} = \sum_{i=1}^{T}(\mathcal{X}_i \times \prod_{n=i}^{T-1} \mathbf{M}_{n,n+1} \times \widehat{\mathbf{M}_{T,T+j}}). \quad (5)$$

As the equation indicates, instead of only using the information of the last observed frame, we use all observed information for future composition and lower the weight of earlier frames through repeated multiplication of the motion matrices. The $\mathcal{X}$ in the equation represents the observed information of the past frames. The information can be the output features of the image encoder with different scales $f_i \in \mathbb{R}^{H^s \times W^s \times C}$, where C is the feature length; the information can also be the observed frames $I_i \in \mathbb{R}^{H \times W \times 3}$. Since the motion matrices are constructed from a certain scale of image features, there will be incompatibilities between the matrices and some features. To enable matrix multiplication between the motion matrices and the observed features (at any scales) or images, we borrow the pixel unshuffle and shuffle operations from [39]. The pixel unshuffle operation reshapes features or images into the identical scale as the motion matrices for matrix multiplication. Afterward, the pixel shuffle operation reshapes the results of the matrix multiplication back to the original scales of the features or images. See Figure 4 for demonstration. This whole process involves little information loss. In Table 6, we examine the multi-scale feature composition design. We find out that in general, the system achieves better performance when more scales of features take part in the feature composition.

3.6. Future Decoding

The future decoding procedure is used for aggregating and processing the composed features to formulate the final output. Since in the feature composition procedure, all scales of features are able to compose their corresponding scales of features for future frames, we adopt the decoder structure of UNet [37] with RRDB blocks [44] in the end to implement MMVP's image decoder. This design allows the composed features from all scales of image features, as well as the original images, to contribute to the final output. We use MSE loss to supervise the framework training.

4. Experiment

In this section, we evaluate MMVP quantitatively and qualitatively against existing state-of-the-art methods on various publicly available datasets. In addition, we analyze MMVP through a set of ablation studies on the UCF Sports dataset to better illustrate our design logic.

4.1. Datasets

We briefly review three widely used datasets and their configurations for MMVP evaluation.

Table 1: Experiment settings for each dataset.

Dataset	Resolution	Train	Test
UCF Sports	512×512	$4 \rightarrow 1$	$4 \rightarrow 6$
KTH	128×128	$10 \rightarrow 20$ or 40	$10 \rightarrow 20$ or 40
Moving MNIST	64×64	$10 \rightarrow 10$	$10 \rightarrow 10$

UCF Sports [36] contains 150 video sequences collected from various sports scenes with 10 different action types including running, kicking, and diving. It is regarded as a challenging dataset given its low frame rate (10 fps), high resolution (480 × 720), and complex motion patterns. The video sequences in this dataset contain both rapid foreground (e.g. athletes) and background movements (e.g. the camera motion). We perform two different training/validation splits of this dataset, splits from STRPM [5] and our own splits, to facilitate different evaluation purposes. We use the STRPM splits for performance comparison with other methods (Table 4). In STRPM, long video sequences are cut into several short clips; for a particular sequence, some of its clips may be put into the training split, and others into the validation split. To avoid the appearance features of certain sequences in the validation set to be exposed in the training set, and to thoroughly evaluate MMVP's capability of video prediction, we generate our own splits for UCF Sports, which choose 90% of the video sequences from each action type to form the training samples, and the rest 10% to form the validation samples.

Notably, within the same validation set, the difficulty level of different samples varies a lot. Some easier video clips contain static backgrounds and slow-moving objects, while others are more difficult for involving drastic camera movement and/or fast-moving objects. To better understand the model's prediction ability in different scenarios, we apply certain thresholds of the structural similarity index measure (SSIM) between the last observed frame and the first future frame to divide the UCF Sports validation set into three subsets: the easy ($\text{SSIM} \leq 0.9$), intermediate ($0.6 \leq \text{SSIM} < 0.9$), and hard subsets ($\text{SSIM} < 0.6$), which respectively take up 66%, 26%, and 8% of the full set.

We use our own UCF Sports splits in the ablation studies (Sec. 4.3). For comparisons, we also train and test STIP [4] (extension of STRPM [5]) and SimVP [12] on our own UCF Sports splits; see Table 6.

KTH [38] contains videos of 25 individuals performing six types of actions, i.e., walking, jogging, running, boxing, hand-waving, and hand-clapping. Following previous works [43, 46, 12], we use persons 1-16 for training and persons 17-25 for validation. The videos in KTH are in grayscale, but the challenging part of this dataset is its experiment setting, which requires the system to output 20 or 40 frames given only 10 past frames.

Moving MNIST [41] is a synthetic dataset. Each sequence in the dataset consists of two digits moving independently within the 64 × 64 grid and bouncing off the boundary. During the training time, by assigning different initial locations and velocities to each digit, one can generate an infinite number of sequences, and train the system to predict the future 10 frames from the previous 10 frames. For fair comparisons, we use the pre-generated 10000 sequences[12] for validation.

4.2. Metrics

We use the peak signal-to-noise ratio (**PSNR**) and structural similarity index measure (**SSIM**) to evaluate the image quality of the predicted frames. We use the implementation of PSNR and SSIM from the package *scikit-learn* [34]. For the UCF Sports dataset, we use Learned Perceptual Image Patch Similarity (**LPIPS**) [58] for better comparison with existing methods. LPIPS represents the perceptual quality of the predicted frames. It measures the feature distance of corresponding image patches between the predicted frames and the ground truth. Following the setting of these previous works, in the experiments on Moving MNIST, we use the sum of the mean squared error (**MSE**) from the entire frame to evaluate the image quality.

4.3. Ablation Study

We conduct the ablation study of future composition (Table 6), filter block (3rd and 4th rows of Table 2), and hyperparameters for both the image encoder/decoder and the matrix predictor (1st, 2nd, and 4th rows of Table 2) on our splits of the UCF Sports dataset.

We first examine how the number of feature scales for feature composition impacts the results. Since we use the UNet structure with three times downsampling/upsampling operations to encode, we have four different scales of features for composition. For UCF Sports, the scales are $1, \frac{1}{2}, \frac{1}{8}, \frac{1}{16}$. The motion matrices are constructed using features of $\frac{1}{8}$ scale. From Table 6 3rd to 5th rows, we observe consistent performance boosts on all metrics and subsets when we involve more scales of features for composition. We then test if using images for feature composition also boosts performance. By comparing the 5th and 6th rows of Table 6, we can still see an increasing pattern. However, since the scale of the image is identical to the largest scale of the features, to avoid redundancy and possible conflicts, we remove the features of the largest scale and produce the results of the last row in Table 6. We then see a comparable result to the one that uses all scales of features as well as the image. We adopt the model setting of the last row for the rest of the experiments.

As mentioned in Sec. 3.2, we do not directly use the output of the image encoder to construct the motion matrices. Instead, we add a filter block to filter out irrelevant features and help the motion matrices focus more on describing the temporal similarity regarding the motion-related features. Table 2 shows that the model with the filter block generally achieves better results on the UCF Sports validation set.

Table 2: Ablation study on UCF Sports with LPIPS metrics (the lower the better), including the feature length of image and motion, and the usage of the filter block (F-Block).

Img	Motion	F-Block	Full	Easy	Intermediate	Hard	Param #
16	4	✓	0.1184	0.0592	0.1811	0.4168	0.70M
16	8	✓	0.1175	0.0600	0.1768	0.4122	0.71M
32	8	×	0.1124	**0.0574**	0.1729	0.3819	2.57M
32	8	✓	**0.1062**	0.0580	**0.1569**	**0.3510**	2.79M
STIPHR [4]			0.1626	0.1066	0.2271	0.4450	18.05M
SimVP [12]			0.1326	0.0584	0.1951	0.5600	3.47M

Furthermore, Table 2 also shows an ablation study on the feature length of the appearance-related modules (image encoder and decoder) and the motion-related module (matrix predictor). We define the configuration of the second row as the MMVP-mini for Figure 1.

4.4. Motion Matrix Visualization

In Figure 5, we visualize the predicted motion matrices for certain selected feature patches. In all the demonstrated samples, the many-to-many relationships described by the predicted motion matrices are able to accurately capture the feature patches moving tendency. They show that the motion matrix can describe a wide range of motion patterns and scenarios, including multiple objects (1st sample of UCF Sports), the single person moving (KTH), multiple persons moving and camera moving (2nd sample of UCF Sports), and synthesized motions (Moving MNIST).

4.5. Comparison with SOTA Methods

We compare MMVP with existing SOTA methods on three popular datasets: UCF Sports (STRPM [5] splits, Table 4), KTH (Table 3), and Moving MNIST (Table 5).

On the UCF Sports dataset, we observe large performance gains by MMVP compared to other methods. As mentioned above, the motion patterns in UCF Sports are the most complex of the three datasets due to many difficult cases, e.g., fast movement, camera moving, and motion blur. But the two-stream design of MMVP has shown its ability in such complex scenarios. Meanwhile, the video resolution of UCF Sports is also the highest among the three datasets. Most existing methods in Table 4 are inherently not designed for high-resolution videos, which requires larger network capacities to maintain appearance information. Although STRPM [5] and STIP [4] proposed residual temporal modules designed for high-resolution videos and they could largely surpass previous methods on both

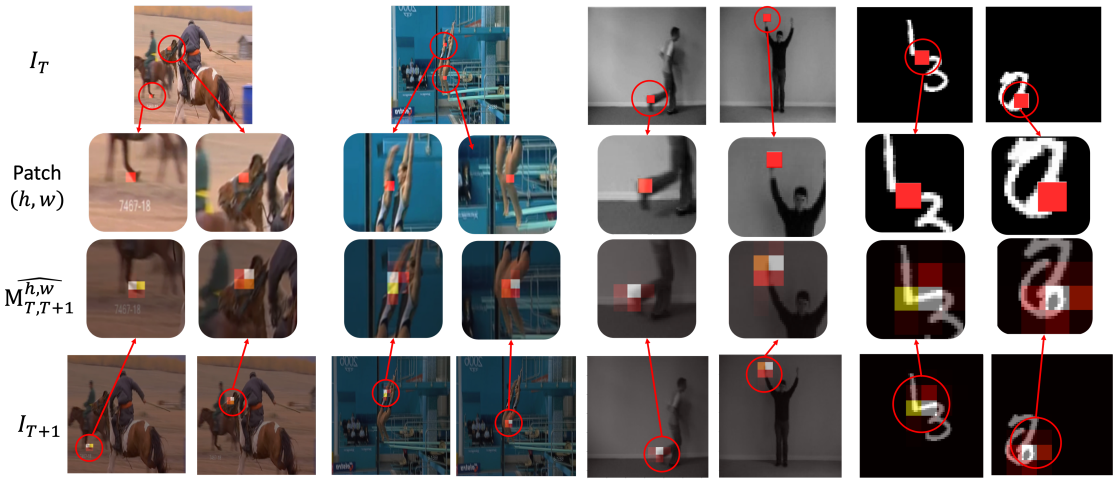

Figure 5: Predicted motion matrix visualization. We highlight the selected feature patch(es) at (h, w) in the last observed frame I_T in red and visualize their corresponding predicted motion matrices $\mathrm{M}^{h,w}_{T,T+1} \in \mathbb{R}^{H \times W}$ overlaying with the first future frame I_{T+1}. A brighter color indicates a higher predicted value. We select two samples for each dataset. From left to right, samples originate in the validation set of UCF Sports, KTH, and Moving MNIST.

metrics, the appearance information loss in their methods is still unavoidable. MMVP, nevertheless, allows the image and multi-scale features to reach the decoder through a feature composing procedure, successfully minimizes the information loss, and achieves the best performance among all methods across all metrics. MMVP's success on the UCF Sports dataset validates its readiness for real-world applications, and its scalability for high-resolution videos.

Another notable result is MMVP's ability in long-term information preservation; see Table 3. Despite having lower video resolutions and less complex motion patterns than UCF Sports, KTH as a real-world dataset still presents a difficulty as it requires systems to do long-term prediction given short-term observations. On the KTH dataset, MMVP achieves the #1 performance on SSIM metrics for both experiment settings with a small performance gap (27.54 to 26.35) in PSNR between the two settings. This is contributed by the design that the matrix predictor of MMVP predicts the temporal similarity matrices between the future frames and the last observed frame instead of their predecessors. Then, every predicted frame is composed of valid observed information. It reduces accumulated errors and ensures good performance for long-term prediction.

Compared to the performance gains on UCF Sports, MMVP does not show much advantage on the Moving MNIST dataset. One speculation is that with video prediction research advancing, the problem of two-digits Moving MNIST with low resolution (64×64) is nearly solved. With sufficient training time [12], current video prediction systems may all be able to achieve promising results. To further promote the growth of this field, researchers can consider more challenging datasets and experiment settings.

Table 3: Performance comparison on the KTH dataset

Method	KTH10 → 20		KTH10 → 40	
	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑
MCnet (ICLR2017) [43]	0.804	25.95	0.73	23.89
ConvLSTM (NeurIPS2015) [40]	0.712	23.58	0.639	22.85
SAVP (arXiv2018) [25]	0.746	25.38	0.701	23.97
VPN (PMLR2017) [23]	0.746	23.76	–	–
DFN (NeurIPS2016) [20]	0.794	27.26	0.652	23.01
fRNN (ECCV2018) [32]	0.771	26.12	0.678	23.77
Znet (ICME2019) [56]	0.817	27.58	–	–
SV2Pi (ICLR2018) [1]	0.826	27.56	0.778	25.92
SV2Pv (ICLR2018) [1]	0.838	27.79	0.789	26.12
PredRNN (NeurIPS2017) [47]	0.839	27.55	0.703	24.16
VarNet (IROS2018) [22]	0.843	28.48	0.739	25.37
SVAP-VAE (arXiv2018) [25]	0.852	27.77	0.811	26.18
PredRNN++ (ICML2018) [45]	0.865	28.47	0.741	25.21
MSNET (BMVC2019) [26]	0.876	27.08	–	–
E3d-LSTM (ICLR2019) [46]	0.879	29.31	0.810	27.24
STMFANet (CVPR2020) [21]	0.893	**29.85**	0.851	**27.56**
MMVP (ours)	**0.906**	27.54	**0.888**	26.35

Moreover, we analyze MMVP's performance on data of different difficulty levels using our own split of the UCF Sports dataset. We define the difficulty of a video using the SSIM between the last observed frame and the first frame to be predicted. This is because a lower SSIM indicates a larger difference between the two frames, and less possibility for a system to take the shortcut by using the residual information from past frames and not actually predicting the motion. Table 6 shows the quantitative evaluation compared with SimVP [12] and STIP [4]: as the difficulty level of the validation subset increases, the performance gap between MMVP and other methods also increases.

Besides quantitative evaluation, we also showcase several qualitative visualizations in Figure 6 and compare them

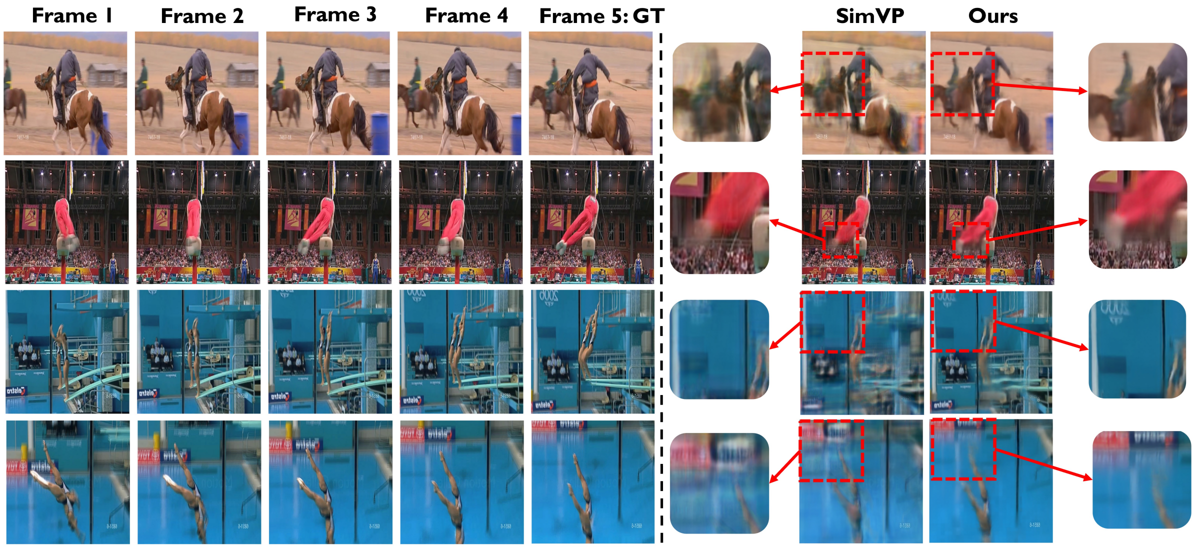

Figure 6: Qualitative results on our own splits of the UCF Sports dataset.

Table 4: Performance comparison on UCF Sports STRPM split

Method	$t = 5$		$t = 10$	
	PSNR ↑	LPIPS$_{\times 100}$ ↓	PSNR ↑	LPIPS$_{\times 100}$ ↓
ConvLSTM (NeurIPS2015) [40]	26.43	32.20	17.80	58.78
BeyondMSE (ICLR2016) [30]	26.42	29.01	18.46	55.28
PredRNN (NeurIPS2017) [47]	27.17	28.15	19.65	55.34
PredRNN++ (ICML2018) [45]	27.26	26.80	19.67	56.79
SAVP (arXiv2018) [25]	27.35	25.45	19.90	49.91
SV2P (ICLR2018) [1]	27.44	25.89	19.97	51.33
E3D-LSTM (ICLR2019) [46]	27.98	25.13	20.33	47.76
CycleGAN (CVPR2019) [24]	27.99	22.95	19.99	44.93
CrevNet (ICLR2020) [55]	28.23	23.87	20.33	48.15
MotionRNN (CVPR2021) [51]	27.67	24.23	20.01	49.20
STRPM (CVPR2022) [5]	28.54	20.69	20.59	41.11
STIP (arXiv2022) [4]	30.75	12.73	21.83	39.67
DMVFN (CVPR2023) [19]	30.05	10.24	22.67	22.50
MMVP (Ours)	**31.68**	**7.88**	**23.25**	**22.24**

Table 5: Comparisons on Moving MNIST

Method	MSE	SSIM
ConvLSTM (NIPS 2015) [40]	103.3	0.707
PredRNN (NIPS 2017) [47]	56.8	0.867
PredRNN-V2 (Arxiv 2021) [48]	48.4	0.891
CausalLSTM (ICML 2018) [45]	46.5	0.898
MIM (CVPR 2019) [49]	44.2	0.910
E3D-LSTM (ICLR 2018) [46]	41.3	0.920
PhyDNet (CVPR 2020) [14]	24.4	0.947
CrevNet (ICLR 2020) [55]	22.3	0.949
SimVP (CVPR 2022) [12]	23.8	0.948
MMVP(ours)	**22.2**	**0.952**

Table 6: Ablation study on sources for future composition and the comparison with other SOTA methods on UCF Sports.

Method	Composition source					Full set			Easy (SSIM $\geq$ 0.9)			Intermediate (0.6 $\leq$ SSIM $<$ 0.9)			Hard (SSIM $<$ 0.6)			Param#
	Img	1	$1/2$	$1/8$	$1/16$	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS↓	SSIM ↑	PSNR ↑	LPIPS ↓	
STIPHR [4]			-			0.8817	28.17	0.1626	0.9491	30.65	0.1066	0.8351	23.97	0.2271	0.4673	15.97	0.4450	18.05M
SimVP [12]			-			0.9189	29.97	0.1326	0.9664	32.87	0.0584	0.8845	25.79	0.1951	0.6267	18.99	0.5600	3.47M
MMVP	×	×	×	✓	✓	0.9000	28.31	0.1874	0.9375	30.43	0.1342	0.8759	25.36	0.2304	0.6593	19.90	0.4992	2.75M
	×	×	✓	✓	✓	0.9284	30.14	0.1115	0.9667	32.79	0.0603	0.8937	26.11	0.1693	0.7159	20.71	0.3570	2.79M
	×	✓	✓	✓	✓	0.9296	30.22	0.1064	0.9669	32.87	0.0576	0.8965	26.26	0.1571	0.7199	20.76	0.3555	2.80M
	✓	✓	✓	✓	✓	0.9296	30.29	**0.1051**	**0.9675**	32.99	**0.0567**	0.8958	26.22	**0.1554**	0.7175	20.76	0.3517	2.80M
	✓	×	✓	✓	✓	**0.9300**	**30.35**	0.1062	0.9674	**33.05**	0.0580	**0.8970**	**26.29**	0.1569	**0.7203**	**20.84**	**0.3510**	2.79M

with SimVP [12] which achieved the second-best performance on our splits of the UCF Sports dataset. We especially select four samples in the hard and intermediate subsets for visualization. The first sample in Figure 6 shows a far-away object with long-distance movement. This is caused by the low fps of the dataset. MMVP accurately captures the displacement of the object while not losing the color and shape of the object. The challenge of the second sample in Figure 6 is also typical to UFC Sports, i.e. fast movement with motion blur. MMVP is able to recover the correct shape of the athletes' feet even if they are blurred in the observed frames. The third and fourth samples both show a fast camera movement, with and without large foreground movements. Camera moving with complex backgrounds is extremely difficult for video prediction. Even SimVP, which generated high-quality images for most of the datasets, barely captured the camera movements in these two samples and caused drastic blurs. Benefiting from its appearance-agnostic motion prediction module, MMVP achieved impressive performance in such cases.

5. Discussion

From Table 2 and Table 6 in Sec. 4, we observe that the proposed MMVP framework can always achieve better or comparable performance with significantly fewer parameters compared to the other SOTA methods. To better understand the high efficiency of MMVP, we break down a video prediction system into three components: i) content encoder, which encodes the image sequences; ii) prediction-related modules, which take charge of predicting features for future frames based on the output of the content encoder; iii) content decoder, which decodes the output of the predicted features output by the prediction-related modules. Then we examine the model size for each component in three video prediction systems: STIP [4], SimVP [12], and the proposed MMVP (configurations follow the second row and the fourth row in Table 2); see Table 7.

In Table 7, prediction-related modules in STIP [4] and SimVP [12] handle both motion and appearance features. However, in MMVP, prediction-related modules specifically handle only the motion features, leaving all appearance features to the content encoder and decoder. The most noticeable fact from Table 7 is that in STIP [4] and SimVP [12], the prediction-related modules take majority of the model parameters (the ratios are 99.7% and 99.4% respectively). In contrast, motion-related modules in MMVP only take 14.5% and 15.9% of the parameters. This fact supports our argument in Sec. 1 that decoupling motion prediction and appearance maintenance effectively avoid the cumbersome structures of the prediction modules and largely improve the prediction efficiency. Another observation is that despite the small size, the prediction-related modules in MMVP can still support high-quality motion prediction. This validates the efficiency of the motion matrices when describing the motion information, which results in a lightweight design of the prediction module.

Table 7: Model size breakdown. The numerical values are the number of parameters taken by each component in the video prediction systems.

Method	Content Encoder	Prediction Modules	Content Decoder	Total
STIPHR [4]	29.86K	17994.8K	29.54K	18054.2K
SimVP [12]	7.5k	3447.1K	11.7K	3466.3K
MMVP-mini	369.4K	113.2K	228.3K	710.9K
MMVP	1472.2K	404.9K	911.9K	2789.0K

6. Conclusion

The proposed Motion-Matrix-based Video Prediction framework (MMVP) is an end-to-end trainable two-stream pipeline. MMVP uses motion matrices to represent appearance-agnostic motion patterns. As the sole input of the motion prediction module in MMVP, the motion matrix can i) describe the many-to-many relationships between feature patches without training for extra modules; ii) intuitively compose future features with multi-scale image features through matrix multiplication. It helps the motion prediction become more focused, and efficiently reduces the information loss in appearance. Extensive experiments demonstrate the superiority of MMVP compared to existing SOTA methods in both the model size and performance.

References

[1] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017. 7, 8

[2] Xinzhu Bei, Yanchao Yang, and Stefano Soatto. Learning semantic-aware dynamics for video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 902–912, 2021. 2, 3

[3] Po-Rong Chang and Jen-Tsung Hu. Optimal nonlinear adaptive prediction and modeling of mpeg video in atm networks using pipelined recurrent neural networks. *IEEE Journal on Selected Areas in Communications*, 15(6):1087–1100, 1997. 1

[4] Zheng Chang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. Stip: A spatiotemporal information-preserving and perception-augmented model for high-resolution video prediction. *arXiv preprint arXiv:2206.04381*, 2022. 1, 2, 3, 5, 6, 7, 8, 9

[5] Zheng Chang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. Strpm: A spatiotemporal residual predictive model for high-resolution video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13946–13955, 2022. 1, 2, 3, 5, 6, 8

[6] Emily L Denton et al. Unsupervised learning of disentangled representations from video. *Advances in neural information processing systems*, 30, 2017. 2, 3

[7] Duarte Duque, Henrique Santos, and Paulo Cortez. Prediction of abnormal behaviors for intelligent video surveillance systems. In *2007 IEEE Symposium on Computational Intelligence and Data Mining*, pages 362–367. IEEE, 2007. 1

[8] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018. 1

[9] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017. 1

[10] Hang Gao, Huazhe Xu, Qi-Zhi Cai, Ruth Wang, Fisher Yu, and Trevor Darrell. Disentangling propagation and generation for video prediction. In *Proceedings of the IEEE/CVF*

international conference on computer vision, pages 9006–9015, 2019. 2, 3

[11] Xiaojie Gao, Yueming Jin, Qi Dou, Chi-Wing Fu, and Pheng-Ann Heng. Accurate grid keypoint learning for efficient video prediction. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5908–5915. IEEE, 2021. 2, 3

[12] Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z Li. Simvp: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3170–3180, 2022. 1, 2, 5, 6, 7, 8, 9

[13] Bernd Girod. The efficiency of motion-compensating prediction for hybrid coding of video sequences. *IEEE Journal on selected areas in communications*, 5(7):1140–1154, 1987. 1

[14] Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11474–11484, 2020. 8

[15] Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martín-Martín, and Li Fei-Fei. Maskvit: Masked visual pre-training for video prediction. *arXiv preprint arXiv:2206.11894*, 2022. 3

[16] Noriaki Hirose, Fei Xia, Roberto Martín-Martín, Amir Sadeghian, and Silvio Savarese. Deep visual mpc-policy learning for navigation. *IEEE Robotics and Automation Letters*, 4(4):3184–3191, 2019. 1

[17] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 4

[18] Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li F Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. *Advances in neural information processing systems*, 31, 2018. 3

[19] Xiaotao Hu, Zhewei Huang, Ailin Huang, Jun Xu, and Shuchang Zhou. A dynamic multi-scale voxel flow network for video prediction. *arXiv preprint arXiv:2303.09875*, 2023. 8

[20] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. *Advances in neural information processing systems*, 29, 2016. 7

[21] Beibei Jin, Yu Hu, Qiankun Tang, Jingyu Niu, Zhiping Shi, Yinhe Han, and Xiaowei Li. Exploring spatial-temporal multi-frequency analysis for high-fidelity and temporal-consistency video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4554–4563, 2020. 7

[22] Beibei Jin, Yu Hu, Yiming Zeng, Qiankun Tang, Shice Liu, and Jing Ye. Varnet: Exploring variations for unsupervised video prediction. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5801–5806. IEEE, 2018. 7

[23] Nal Kalchbrenner, Aäron Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. In *International Conference on Machine Learning*, pages 1771–1779. PMLR, 2017. 3, 7

[24] Yong-Hoon Kwon and Min-Gyu Park. Predicting future frames using retrospective cycle gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1811–1820, 2019. 8

[25] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018. 7, 8

[26] Jungbeom Lee, Jangho Lee, Sungmin Lee, and Sungroh Yoon. Mutual suppression network for video prediction using disentangled features. *arXiv preprint arXiv:1804.04810*, 2018. 7

[27] Xiaodan Liang, Lisa Lee, Wei Dai, and Eric P Xing. Dual motion gan for future-flow embedded video prediction. In *proceedings of the IEEE international conference on computer vision*, pages 1744–1752, 2017. 2, 3

[28] Jianping Lin, Dong Liu, Houqiang Li, and Feng Wu. M-lvc: Multiple frames prediction for learned video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3546–3554, 2020. 1

[29] Bowen Liu, Yu Chen, Shiyu Liu, and Hun-Seok Kim. Deep learning in latent space for video prediction and compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 701–710, 2021. 1

[30] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015. 8

[31] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari, 2010. 1

[32] Marc Oliu, Javier Selva, and Sergio Escalera. Folded recurrent neural networks for future video prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 716–731, 2018. 7

[33] Sergiu Oprea, Pablo Martinez-Gonzalez, Alberto Garcia-Garcia, John Alejandro Castro-Vargas, Sergio Orts-Escolano, Jose Garcia-Rodriguez, and Antonis Argyros. A review on deep learning techniques for video prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):2806–2826, 2020. 2

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 6

[35] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent video transformer. *arXiv preprint arXiv:2006.10704*, 2020. 3

[36] Mikel D Rodriguez, Javed Ahmed, and Mubarak Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008. 2, 5

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image com-*

puting and computer-assisted intervention, pages 234–241. Springer, 2015. 5

[38] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36. IEEE, 2004. 2, 5

[39] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 5

[40] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015. 3, 7, 8

[41] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852. PMLR, 2015. 2, 6

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 3

[43] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017. 2, 3, 5, 7

[44] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018. 3, 4, 5

[45] Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and S Yu Philip. Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *International Conference on Machine Learning*, pages 5123–5132. PMLR, 2018. 1, 2, 3, 7, 8

[46] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *International conference on learning representations*, 2018. 1, 2, 3, 5, 7, 8

[47] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Advances in neural information processing systems*, 30, 2017. 1, 2, 3, 7, 8

[48] Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, Philip Yu, and Mingsheng Long. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 8

[49] Yunbo Wang, Jianjin Zhang, Hongyu Zhu, Mingsheng Long, Jianmin Wang, and Philip S Yu. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9154–9162, 2019. 8

[50] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. *arXiv preprint arXiv:1906.02634*, 2019. 3

[51] Haixu Wu, Zhiyu Yao, Jianmin Wang, and Mingsheng Long. Motionrnn: A flexible model for video prediction with spacetime-varying motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15435–15444, 2021. 8

[52] Yue Wu, Qiang Wen, and Qifeng Chen. Optimizing video prediction via video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17814–17823, 2022. 2, 3

[53] Ren Yang, Radu Timofte, and Luc Van Gool. Advancing learned video compression with in-loop frame prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022. 1

[54] Wei Yu, Yichao Lu, Steve Easterbrook, and Sanja Fidler. Crevnet: Conditionally reversible video prediction. *arXiv preprint arXiv:1910.11577*, 2019. 2

[55] Wei Yu, Yichao Lu, Steve Easterbrook, and Sanja Fidler. Efficient and information-preserving future frame prediction and beyond. 2020. 1, 8

[56] Jianjin Zhang, Yunbo Wang, Mingsheng Long, Wang Jianmin, and S Yu Philip. Z-order recurrent neural networks for video prediction. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 230–235. IEEE, 2019. 7

[57] Qianqian Zhang, Guorui Feng, and Hanzhou Wu. Surveillance video anomaly detection via non-local u-net frame prediction. *Multimedia Tools and Applications*, 81(19):27073–27088, 2022. 1

[58] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[59] Xianguo Zhang, Tiejun Huang, Yonghong Tian, and Wen Gao. Background-modeling-based adaptive prediction for surveillance video coding. *IEEE Transactions on Image Processing*, 23(2):769–784, 2013. 1