# Appendix-MMVP: Motion-Matrix-based Video Prediction

Anonymous ICCV submission

Paper ID 4215

## 1. Training strategy

All models in the paper are implemented using Pytorch[4] on a single NVIDIA A100 GPU. The initial learning is set to be $1e^{-3}$ and decayed following a cosine restart learning scheduler[3]. We use AdamW optimizer during the training. We show the other training-related hyperparameters for each dataset in Table 1.

| Dataset | Restart Period | Batch Size | Total Epoch |
|---|---|---|---|
| UCF Sports | 100 | 4 | 300 |
| KTH | 50 | 16 | 150 |
| MNIST | 1000 | 32 | 3000 |

Table 1: Training configuration for each dataset in the paper.

## 2. Framework Implementation

This section we demonstrate the inner structure of each module that we adopted for MMVP implementation in this work. MMVP contains three major steps: i) feature extraction, which includes an image encoder and a filter block, see Figure 1; ii) motion matrix construction and prediction, see Figure 2; and iii) Future composition and decoding, see Figure 3. We will apply a softmax operation to every $\mathbf{M}$ before they take part in the future composition step.

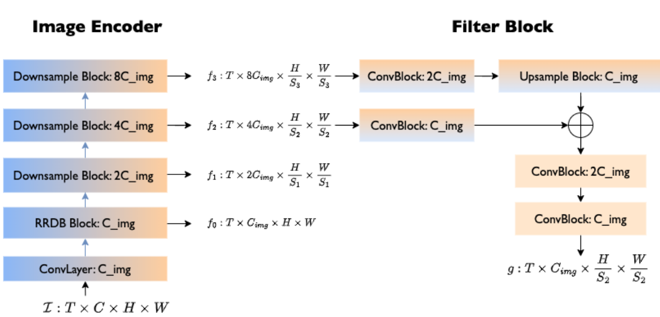


Figure 1: Spatial feature extraction.

For the experiments on each dataset, the implementations all follow the structures shown in Figure 1, 2, and

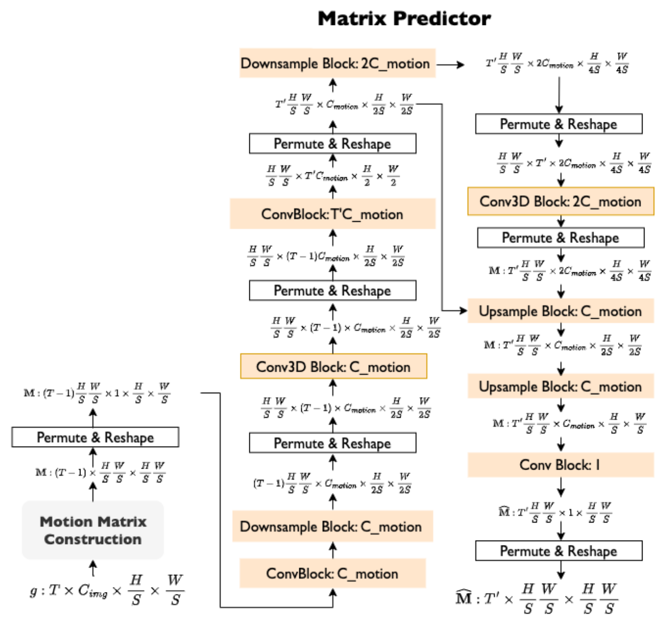


Figure 2: Motion matrix construction and prediction.

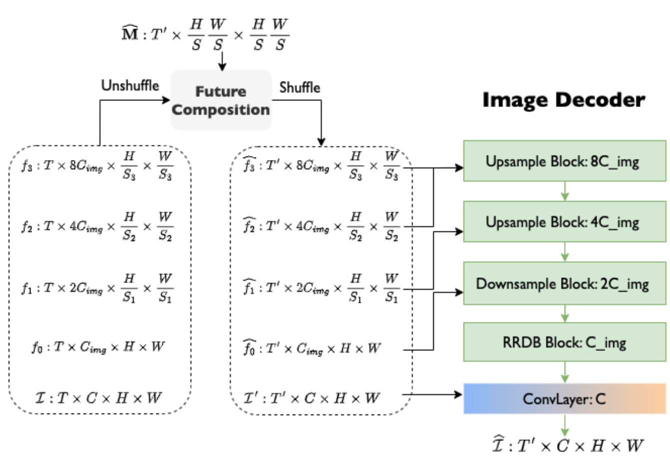


Figure 3: Future composition and decoding.

3. However, there are three hyperparameters that are different when implementing the models for different datasets: i) $C_{img}$, the base channel of the image encoder & decoder; ii) $C_{motion}$, the base channel of the matrix predictor and iii) the down-sample ratio $S$ between the hidden features and the original image. When selecting the two base channel numbers, we consider the video resolution, the complexity of the motion patterns, and the length of the future frames. For most cases, when video resolution is higher, or the images are more informative, $C$ should be set to a larger number; when the motion pattern is more complex or the prediction length is longer, $C_{motion}$ should be larger. Here we show the $C_{img}$ and $C_{motion}$ for each dataset in Tab. 2. Interestingly, it is easy to note that the model we use on Moving-MNIST consists of the largest parameter numbers among the three datasets while Moving-MNIST is a single channel image with only digit numbers. One reason is that the motion pattern of Moving-MNIST has the least constraints among the three, the two digits are bouncing everywhere on the image, which requires the model to have a larger capacity. We will release the code and the pre-trained models later. In MMVP paper, we choose not to exhaustively search for the optimal combination of the hyper-parameters for each dataset setting or the best network architecture for the image encoder, decoder, and matrix predictor. One may modify our code and achieve better results than what we showcase in the paper.

| Dataset | Resolution | Future length | $C_{img}$ | $C_{motion}$ | $S$ | Param # |
|---|---|---|---|---|---|---|
| UCF Sports | $512 \times 512$ | 1 | 32 | 8 | 8 | 2.8M |
| KTH | $128 \times 128$ | 20 | 16 | 96 | 4 | 4.5M |
| KTH | $128 \times 128$ | 40 | 16 | 96 | 4 | 6.1M |
| MNIST | $64 \times 64$ | 10 | 32 | 192 | 4 | 14.6M |

Table 2: Hyper-parameters in the MMVP implementation for different datasets.

When running the experiments on our splits of UCF sports dataset using SimVP[2] and STIP[1], we strictly follow the hyper-parameters released in their official code. Especially, for STIP, we directly copy their hyper-parameters on the UCF Sports dataset.

# 3. Extensive Visualization

## 3.1. UCF Sports Validation subset

In the paper, we have mentioned that we notice that even within the same validation set, the difficulty level of different samples varies a lot. Some video clips only contain static backgrounds and slow-moving objects while others include drastic camera movement or fast-moving objects. To better understand the model's prediction ability for different scenarios, we use 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 (SSIM $\leq$ 0.9), intermediate, hard subsets (SSIM $<$ 0.6), which take 66%, 26%, and 8% of the full set respectively.

Here we showcase two examples from each subset in Figure 4. We can see that for the samples belonging to the easy subset, the difference between the last observed frame $I_T$ and the first future frame $I_{T+1}$ is very minor, which turns the video prediction task into a signal processing or image reconstruction task (especially for the second sample). Methods that rely too much on the feature shortcuts from the previous methods will have leading performances. Comparing the second sample in the intermediate subset and the first sample in the hard subset, we can clearly observe that the sample in the hard subset may contain more camera movement, which is more challenging for the video prediction system.

## 3.2. Motion Matrix Sequence

In this section, we visualize the motion sequences that are input to the matrix predictor and their corresponding output (See Figure 5). Specifically, in KTH, we demonstrate what the output will be like if it is a sequence of matrices. From the visualization we have two observations: i) For long-term prediction in KTH, the highlighted area of the selected matrix can still fall in the correct region; ii) the heatmap of the matrix describes the layout of each frame, and the basic shapes of the objects in the video. Furthermore, it can be regarded as a semantic segmentation map while the sequence of the matrices reflects the changing pattern of the semantic meaning. All those information provides essential hints for motion prediction.

## 3.3. Extra Qualitative Results

In this section, we show the qualitative results for the other two datasets: Moving-MNIST (Fig. 6) and KTH (Fig. 7

# References


[1] 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. 2

[2] 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. 2

[3] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 1

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An

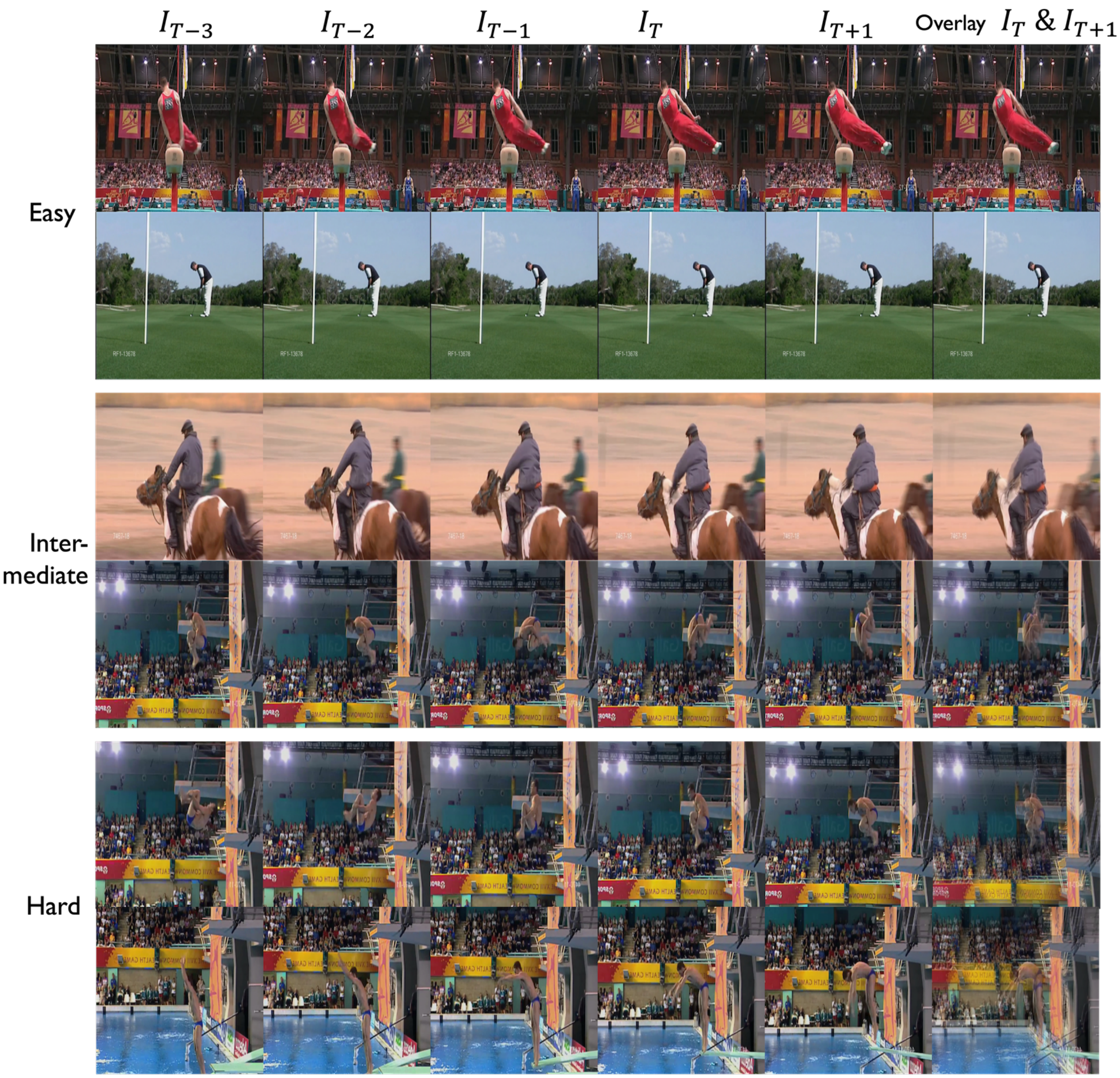


Figure 4: Samples from different subsets of the validation set in UCF Sports. The last column is the overlay of the last observed frame $I_T$ and the first future frame $I_{T+1}$.

imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 1

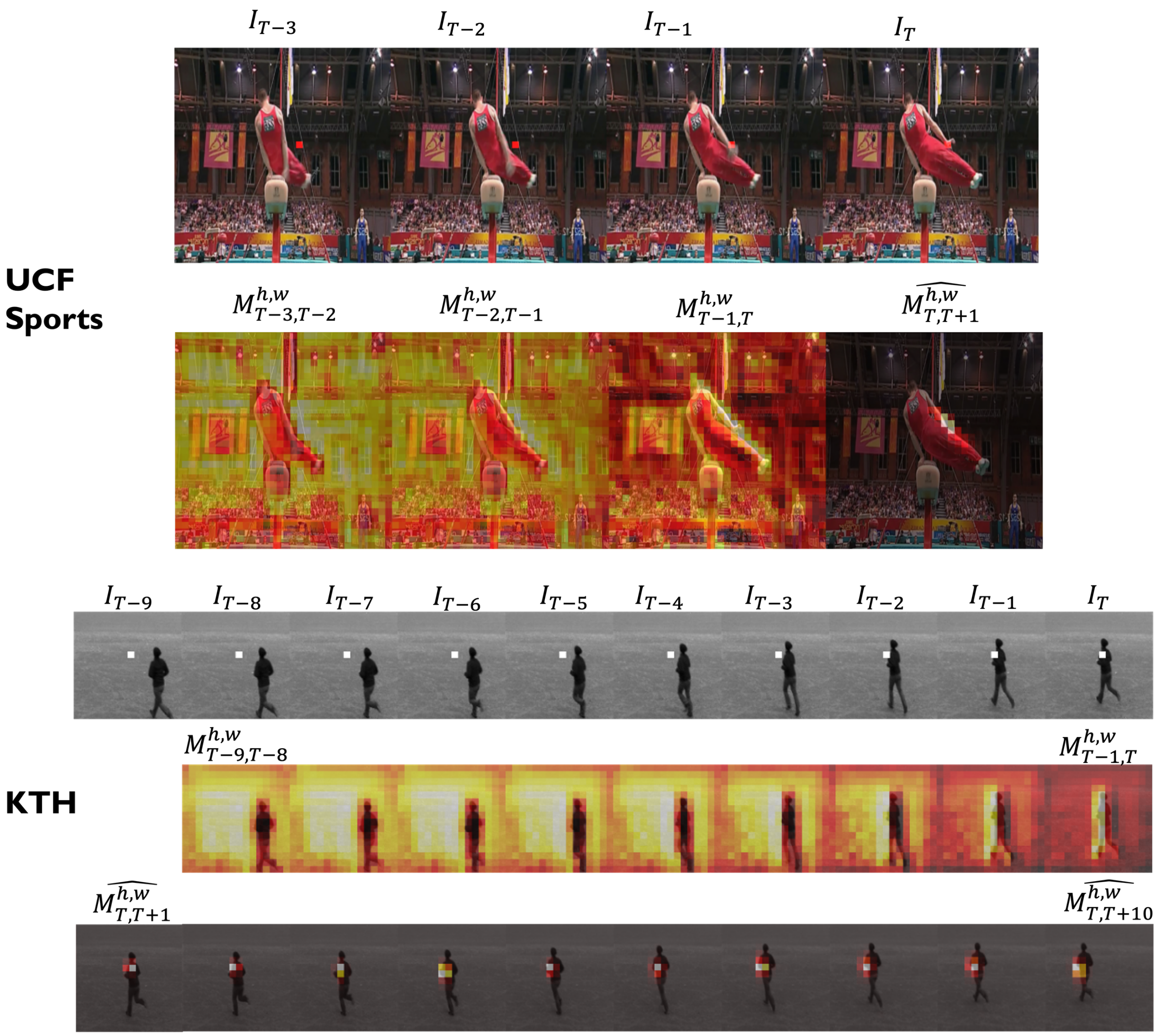


Figure 5: Visualization of the motion matrices. We selected one patch for each video sequence at $(h, w)$ and visualize its corresponding sequence of the matrices as well as the predicted matrices output by the matrix predictor. The selected patch is red in the UCF Sports data sample and white in the KTH data sample.

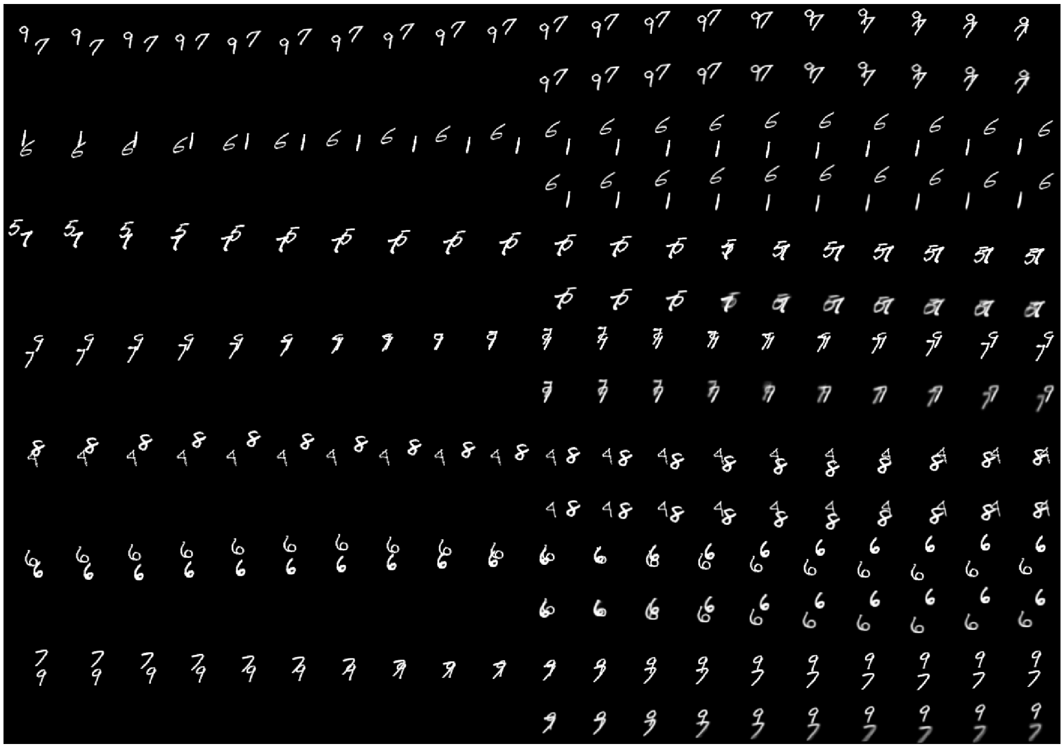

Figure 6: Qualitative results for Moving-MNIST. The upper row of each sample shows the ground truth for 10 future frames and the lower row is the output of MMVP.

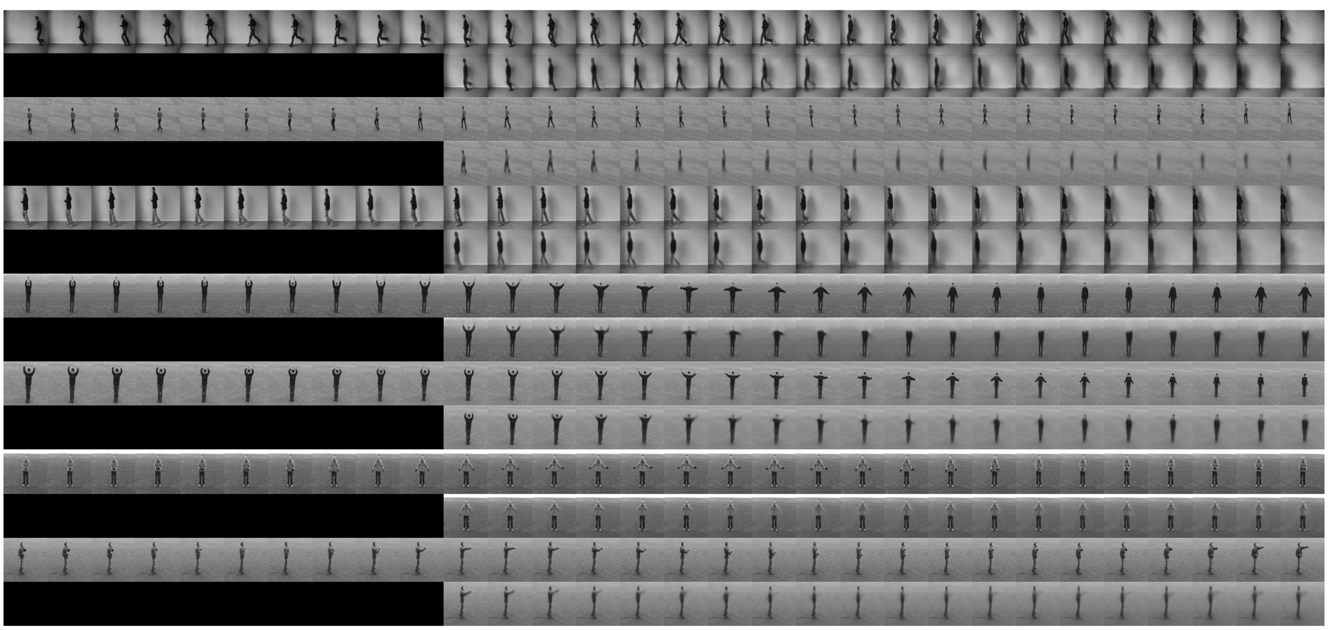

Figure 7: Qualitative results for KTH. The upper row of each sample shows the ground truth for 20 future frames and the lower row is the output of MMVP.