LMVP: Video Predictor with Leaked Motion Information

Dong Wang [1], **Yitong Li** [1], **Wei Cao** [2], **Liqun Chen** [1], **Qi Wei** [3], **Lawrence Carin** [1]
[1] Duke University, [2] Tsinghua University, [3] JP Morgan & Chase
{dong.wang363, yitong.li, lcarin}@duke.edu

Abstract

We propose a Leaked Motion Video Predictor (LMVP) to predict future frames by capturing the spatial and temporal dependencies from given inputs. The motion is modeled by a newly proposed component, motion guider, which plays the role of both learner and teacher. Specifically, it *learns* the temporal features from real data and *guides* the generator to predict future frames. The spatial consistency in video is modeled by an adaptive filtering network. To further ensure the spatio-temporal consistency of the prediction, a discriminator is also adopted to distinguish the real and generated frames. Further, the discriminator leaks information to the motion guider and the generator to help the learning of motion. The proposed LMVP can effectively learn the static and temporal features in videos without the need for human labeling. Experiments on synthetic and real data demonstrate that LMVP can yield state-of-the-art results.

1 Introduction

Video combines structured spatial and temporal information in high dimensions. The strong spatio-temporal dependencies among consecutive frames in video greatly increases the difficulty of modeling. For instance, it is challenging to effectively separate moving objects from background, and predict a plausible future movement of the former [2, 4, 11, 14, 15, 21]. Though video is large in size and complex to model, video prediction is a task that can leverage the extensive online video data without the need of human labeling. Learning a good video predictor is an essential step toward understanding spatio-temporal modeling. These concepts can also be applied to various tasks, like weather forecasting, traffic-flow prediction, and disease control [16, 18, 17].

The recurrent neural network (RNN) is a widely used framework for spatio-temporal modeling. In most existing works, motion is estimated by the subtraction of two consecutive frames and the background is encoded by a convolutional neural network (CNN) [1, 9, 11, 12]. The CNN ensures spatial consistency, while temporal consistency is considered by the recurrent units, encouraging motion to smoothly progress through time. However, information in two consecutive frames is usually insufficient to learn the dynamics. Using 3D convolution to generate future frames can avoid these problems [13, 14], although generating videos by 3D-convolution usually lacks sharpness.

We propose a Leaked Motion Video Predictor (LMVP) for robust future-frame prediction. LMVP generates the prediction in an adversarial framework: we use a generative network to predict next video frames, and a discriminative network to judge the generated video clips. For the motion part, we propose to learn the dynamics by introducing a motion guider, connecting the generator and the discriminator. The motion guider learns the motion feature through training on real video clips, and guides the prediction process by providing possible motion features. At the same time, in contrast with estimating motions by subtracting two consecutive frames, we allow the discriminator to leak high-level extracted dynamic features to the motion guider to further help the prediction. Such dynamic features provide more informative guidance about dynamics to the generator. The spatial

Workshop on Modeling and Decision-Making in the Spatiotemporal Domain, 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

dependencies of video are imposed by a convolutional filter network conditioned on the current frame. This idea is inspired by a conventional signal processing technique named adaptive filter, which can increase the flexibility of the neural network [5]. It is assumed that, each pixel of the predicted frame is a nonlinear function of the neighborhood pixels of the current one, where the nonlinear function is implemented via LMVP as a deep neural network.

2 Models

The video frames are represented as $\boldsymbol{x} \in \mathbb{R}^{T\times H\times W\times C}$, where T is the total number of frames, H is the frame height, W is the width and C is the channel number. Given the first T_0 ($T_0 < T$) frames, the task is to predict the following $T - T_0$ frames. $\boldsymbol{x}_t$ and $\hat{\boldsymbol{x}}_t$ represent for real and predicted video frame at time t, respectively. The model framework is given in Figure 1. It mainly contains a generator G, a motion guider M, and a discriminator D. D distinguishes between the real and predicted video clips. M learns the temporal dependencies among the video through the features leaked from D, and generator G uses the output of motion guider M to predict the next frame based on the current.

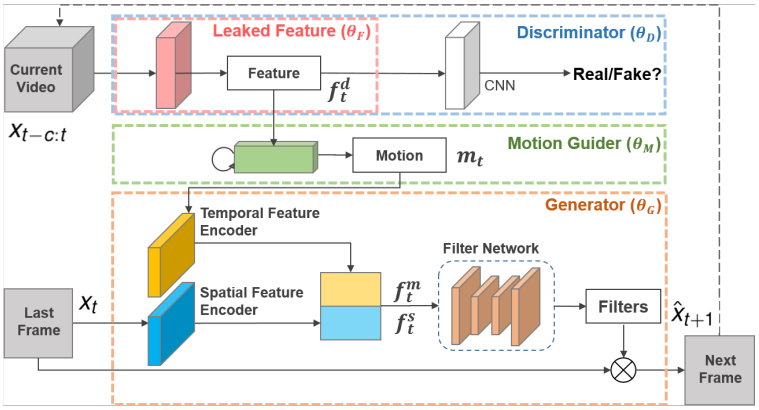

Figure 1: Model Framework.

2.1 Leaked Features from D as Motion Signals

The discriminator D (shown in top of Figure 1) is designed as both a discriminator and a motion feature extractor. The bottom layers of D is a feature extractor $F(\cdot;\boldsymbol{\theta}_F)$, followed by several convolutional and fully connected layers to classify real/fake samples, parameterized by $\boldsymbol{\theta}_C$. Mathematically, given input video clips $\boldsymbol{x}_{t-c:t}$, we have $D(\boldsymbol{x}_{t-c:t};\boldsymbol{\theta}_D) = \text{CNN}(F(\boldsymbol{x}_{t-c:t};\boldsymbol{\theta}_F);\boldsymbol{\theta}_C)$, where $\boldsymbol{\theta}_D = \{\boldsymbol{\theta}_F, \boldsymbol{\theta}_C\}$. The extracted motion feature from $\boldsymbol{x}_{t-c:t}$ is denoted as $\boldsymbol{f}_t^d = F(\boldsymbol{x}_{t-c:t};\boldsymbol{\theta}_F)$, which is the input of motion guider M.

The feature extractor F is implemented as a convolutional network. The output $\boldsymbol{f}_t^d$ is expected to capture motion from $\boldsymbol{x}_{t-c:t}$. The difference between $\boldsymbol{f}_{t+1}^d$ and $\boldsymbol{f}_t^d$ is treated as the dynamic motion feature between two consecutive frames, which is denoted as $\boldsymbol{m}_t$. In contrast of the direct subtraction of two consecutive frames [11], our dynamic motion feature is extracted from two consecutive video clips of length c. Since previous video frames are also included, it can still give reasonable output even if the model fails at previous time step. The discriminator loss can be written as

$$\mathcal{L}_{dis}(\boldsymbol{\theta}_D) = -\mathbb{E}_x[\log D(\boldsymbol{x};\boldsymbol{\theta}_D)] - \mathbb{E}_{\hat{x}}[\log(1 - D(\hat{\boldsymbol{x}};\boldsymbol{\theta}_D))]. \tag{1}$$

2.2 Learning and teaching game of M

To utilize the leaked motion information $\boldsymbol{f}_t^d$ from D, we introduce a motion guider module M, which is inspired by the leaky GAN model [3] for text generation task. The structure of M is displayed in the green dotted box in Figure 1. M has a recurrent structure that takes the extracted motion feature $\boldsymbol{f}_t^d$ as input at each time step t, and outputs a predicted motion feature $\hat{\boldsymbol{m}}_t$. Specifically, the motion guider plays two roles in the model: learner and teacher.

As a learner, M learns the motion in video via leaked feature from D from real video. At time step t, M receives the leaked information $\boldsymbol{f}_t^d$ exacted by D, and predicts the dynamic motion feature

between time t and $t+1$, by forcing $\hat{\boldsymbol{m}}_t = M(\boldsymbol{f}_t^d; \boldsymbol{\theta}_M)$ close to $\boldsymbol{m}_t = \boldsymbol{f}_{t+1}^d - \boldsymbol{f}_t^d$. Denoting the parameters of the motion guider as $\boldsymbol{\theta}_M$, the learner loss function can be written as

$$\mathcal{L}_M^L(\boldsymbol{\theta}_M) = \sum_t ||M(\boldsymbol{f}_t^d; \boldsymbol{\theta}_M) - (\boldsymbol{f}_{t+1}^d - \boldsymbol{f}_t^d)||_2^2. \tag{2}$$

Note that only real video samples are used to update $\boldsymbol{\theta}_M$. The superscript L means "Leaner".

As a teacher, M serves as a guider by providing predicted dynamic motion features to G. During this step, $\boldsymbol{\theta}_M$ is fixed while the generator is updated under the guidance of M. Given the leaked features $\hat{\boldsymbol{f}}_t^d = F(\hat{\boldsymbol{x}}_{t-c:t}; \theta_F)$ of predicted data $\hat{\boldsymbol{x}}_{t-c:t}$ at time t, the output $\hat{\boldsymbol{m}}_t = M(\hat{\boldsymbol{f}}_t^d; \boldsymbol{\theta}_M)$ serves as an input to the generator to predict the next frame $\hat{\boldsymbol{x}}_{t+1}$. This is detailed in the next section.

Since the dynamic motion feature is extracted from a real *video clip* instead of a single frame, M is robust against fail predictions at previous time step, i.e., even if the previous predicted frame diverges from the ground truth.

2.3 Generating the next frame under guide from M

The structure of the generator is shown at the bottom of Figure 1. It contains a spatial feature encoder, a temporal feature encoder, and a filter network. The spatial feature encoder is designed to learn the static background structure $\boldsymbol{f}_t^s$, while temporal feature $\boldsymbol{f}_t^m$ are computed from dynamic motion feature $\hat{\boldsymbol{m}}_t$. Note that only predicted samples have their motion guider output $\hat{\boldsymbol{m}}_t$ flow back to the generator. The spatio-temporal features $\boldsymbol{f}_t^s$ and $\boldsymbol{f}_t^m$ are concatenated and further fed into the filter network. The next frame is predicted by applying the generated adaptive filter from $[\boldsymbol{f}_t^s, \boldsymbol{f}_t^m]$ on the current frame. This technique is also known as visual transformation [14].

As mentioned in Section 2.2, M is updated during the learning step. When generating the next frame $\hat{\boldsymbol{x}}_{t+1}$ in the teaching step, M is fixed and outputs the motion guide $\hat{\boldsymbol{m}}_t$. Specifically, to ensure G generates the next frames following the guidance of M, the dynamic motion feature between the generated video clips at time $t+1$ and t, which is denoted as $\hat{\boldsymbol{f}}_{t+1}^d - \hat{\boldsymbol{f}}_t^d$, should be close to $\hat{\boldsymbol{m}}_t$ from M. Then, the generator is updated by minimizing the following loss function:

$$\mathcal{L}_M^T(\boldsymbol{\theta}_G) = \sum_t \| \left(F([\hat{\boldsymbol{x}}_{t-c+1:t}, G(\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{m}}_t; \boldsymbol{\theta}_G)]) - \hat{\boldsymbol{f}}_t^d \right) - \hat{\boldsymbol{m}}_t \|_2^2, \tag{3}$$

where $\boldsymbol{\theta}_G$ includes all parameters in the generator G. The gradient is taken w.r.t $\boldsymbol{\theta}_G$, while $\hat{\boldsymbol{f}}_t^d$ and $\hat{\boldsymbol{m}}_t$ are treated as inputs. Note that $\hat{\boldsymbol{x}}_t$ and $\hat{\boldsymbol{f}}_t^d$ are the predicted output and leaked motion feature from time step t, respectively. The superscript T in $\mathcal{L}_M^T$ indicates M as a "Teacher".

The total loss function for the generator is

$$\mathcal{L}_{gen}(\boldsymbol{\theta}_G) = \mathcal{L}_{recons}(\boldsymbol{\theta}_G) + \gamma \mathcal{L}_M^T(\boldsymbol{\theta}_G). \tag{4}$$

$\mathcal{L}_{recons}$ is the reconstruction loss function of $\boldsymbol{x}_{t+1}, \hat{\boldsymbol{x}}_{t+1}$, including a pixel-wise cross-entropy/MSE loss and the gradient difference loss (GDL) [8]. The whole model is updated iteratively for each component. A pseudo-algorithm is given in Alg. 1 in the Appendix. The discriminator loss (1) is first evaluated and $\boldsymbol{\theta}_D$ is updated. Then the motion guider parameters $\boldsymbol{\theta}_M$ are updated using only real samples of $\boldsymbol{x}$. The generated parameters $\boldsymbol{\theta}_G$ is updated using loss function (4). In practice, Adam [6] is used to perform the gradient descent optimization.

3 Experiments

Moving MNIST: Each video in the Moving MNIST dataset [10] has 20 frames in total, with two handwritten digits bouncing inside a 64×64 patch. Given 10 frames, the task is to predict the motion of the digits of the following 10 frames. We follow the same training and testing procedure as [10]. Evaluation metrics include Binary Cross Entropy (BCE), Peak Signal to Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM) [19] between the ground truth $\boldsymbol{x}$ and

	BCE	SSIM	PSNR
ConvLSTM [20]	8.96×10^{-2}	0.61	10.74
FPM [10]	8.33×10^{-2}	-	-
DFN [5]	6.89×10^{-2}	0.83	18.4
LMVP	6.13×10^{-2}	0.87	19.6

Table 1: Binary cross-entropy, SSIM and PSNR scores results on Moving MNIST dataset.

the prediction $\hat{x}$. Small values of BCE or large values of SSIM and PSNR indicate good prediction results. In this task, we need to keep the digit shape the same across time (spatial consistency) while giving them reasonable movements (temporal consistency).

Table 1 gives the comparison of LMVP and baseline models. The BCE of LMVP achieves 0.061 per pixel over 10 frames, which is better than state-of-the-art models [20, 10, 5, 11]. The predictions from LMVP and DFN [5] are shown in Figure 4. Input is given in the first row, followed by ground truth of the output, and results from DFN model and our LMVP model. To prove that our model has consistently good result in to the future, Figure 5 in the Appendix gives the SSIM and PSNR comparison over $(t = T_0 + 1, \cdots, T)$. LMVP achieves higher SSIM and PSNR scores than other baseline models through all time steps.

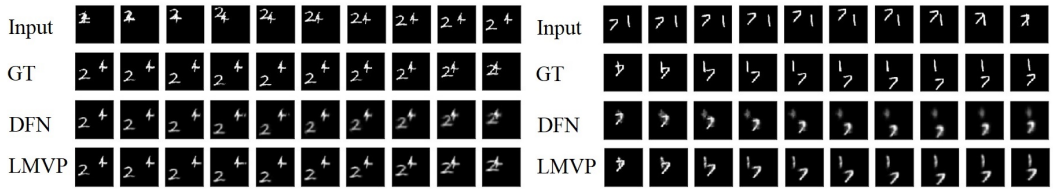

Figure 2: Two prediction examples for the Moving MNIST dataset. From top to down: concatenation of input and ground truth, prediction result of DFN, and prediction result of our model.

Highway Drive: The dataset contains videos were collected from a car-mounted camera during car driving on a highway. The videos contain rich temporal dynamics, including both self-motion of the car and the motion of other objects in the scene [7]. Following the setting used in [5], we split the approximately $20,000$ frames of the 30-minutes video into a training set of $16,000$ frames and a test set of $4,000$ frames. Each frame is of size 64×64. The task is to predict three frames in the future given the past three.

	MSE	SSIM	PSNR
Last Frame	10.34×10^{-3}	0.83	22.11
DFN [5]	3.08×10^{-3}	0.92	26.95
LMVP	2.67×10^{-3}	0.927	27.23

Table 2: MSE (per pixel), SSIM and PSNR scores results on highway driving video dataset.

The prediction results are compared in Table 2 and two samples from the test set are selected in Figure 3. In the prediction results of DFN, the rail of the guidepost becomes curving. However, in the prediction results of LMVP, the rail keeps straight in the first and second predicted frames. To help the visual comparison, this part has been highlighted by a red circle.

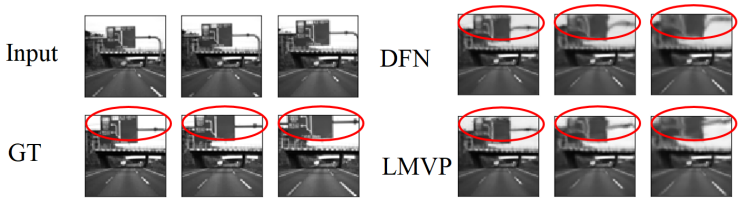

Figure 3: Qualitative prediction examples for the highway driving video dataset. From top to down, left to right: concatenation of input and ground truth, prediction result of DFN, and prediction result of our LMVP.

4 Conclusion

We have proposed the Leaked Motion Video Predictor (LMVP) to handle the spatio-temporal consistency in video prediction. For the dynamics in video, the motion guider learns motion features from real data and guides the prediction. Since the motion guider learns features from video sequences, it is more robust compared to using only single frames as input. For structures of the background, the adaptive filter generates input-aware filters when predicting the next frame, ensuring spatial consistency. Further, A discriminator is adopted to further improve the prediction result. On both synthetic and real datasets, LMVP shows superior results over the state-of-the-art approaches.

References

[1] H. Cai, C. Bai, Y.-W. Tai, and C.-K. Tang. Deep video generation, prediction and completion of human action sequences. *arXiv preprint arXiv:1711.08682*, 2017.

[2] Y.-W. Chao, J. Yang, B. Price, S. Cohen, and J. Deng. Forecasting human dynamics from static images. In *IEEE CVPR*, 2017.

[3] J. Guo, S. Lu, H. Cai, W. Zhang, Y. Yu, and J. Wang. Long text generation via adversarial training with leaked information. *AAAI*, 2018.

[4] M. Henaff, J. Zhao, and Y. LeCun. Prediction under uncertainty with error-encoding networks. *arXiv preprint arXiv:1711.04994*, 2017.

[5] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool. Dynamic filter networks. In *NIPS*, 2016.

[6] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] W. Lotter, G. Kreiman, and D. Cox. Deep predictive coding networks for video prediction and unsupervised learning. *ICLR*, 2017.

[8] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *ICLR*, 2017.

[9] V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. *ICLR*, 2016.

[10] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015.

[11] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee. Decomposing motion and content for natural video sequence prediction. *ICLR*, 2017.

[12] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee. Learning to generate long-term future via hierarchical prediction. *aICML*, 2017.

[13] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NIPS*, 2016.

[14] C. Vondrick and A. Torralba. Generating the future with adversarial transformers. In *CVPR*, 2017.

[15] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, 2016.

[16] D. Wang, W. Cao, J. Li, and J. Ye. Deepsd: supply-demand prediction for online car-hailing services using deep neural networks. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE, 2017.

[17] D. Wang, W. Cao, M. Xu, and J. Li. Etcps: An effective and scalable traffic condition prediction system. In *International Conference on Database Systems for Advanced Applications*. Springer, 2016.

[18] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng. When will you arrive? estimating travel time based on deep neural networks. AAAI, 2018.

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

[20] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*, 2015.

[21] T. Xue, J. Wu, K. Bouman, and B. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NIPS*, 2016.

A Model training

The model is first pre-trained by iteratively updating the parameters of D and G. In each iteration, we first update $\boldsymbol{\theta}_D$ by minimizing the loss $\mathcal{L}_{dis}$ in Equation (1); then, $\boldsymbol{\theta}_F$, $\boldsymbol{\theta}_M$, and $\boldsymbol{\theta}_G$ are jointly updated by minimizing the loss $\mathcal{L}_{gen}$ in Equation (4) with $\gamma = 0$. We found that the above pre-training technique can empirically stabilize the generation process and learn useful leaked information from discriminator.

In the main algorithm loop, D, M, and G are trained iteratively. The algorithm outline is given in Algorithm 1. Firstly, $\boldsymbol{\theta}_D$ is updated according to discriminator loss $\mathcal{L}_{dis}$ while $\boldsymbol{\theta}_M$ and $\boldsymbol{\theta}_G$ are kept fixed. Secondly, $\mathcal{L}_M(\boldsymbol{\theta}_M)$ defined in Equation (2) is evaluated to update $\boldsymbol{\theta}_M$ while $\boldsymbol{\theta}_D$ and $\boldsymbol{\theta}_G$ remain unchanged. The third step is to update $\boldsymbol{\theta}_G$ by minimizing loss $\mathcal{L}_{gen}$ in Equation (4) with $\gamma > 0$. Note that, in both pre-train and main algorithm loop, all the initial hidden states in recurrent architecture are set to zero. The gradient is updated by Adam [6].

Algorithm 1 Leaked Motion Video Prediction

Input: Training videos $\boldsymbol{V} = \{x_{1:T}\}$.
Output: Parameters $\boldsymbol{\theta}_G$, $\boldsymbol{\theta}_D$, $\boldsymbol{\theta}_F$ and $\boldsymbol{\theta}_M$.

Initialize the discriminator, generator, and motion guider with random weights. Initial hidden states in the model are set to zeros.
Pre-train $\boldsymbol{\theta}_D$ using videos in training dataset as positive samples and output from generator as negative samples.
Pre-train $\boldsymbol{\theta}_G$, $\boldsymbol{\theta}_F$ and $\boldsymbol{\theta}_M$ using leaked features from $\boldsymbol{\theta}_F$ according to loss $\mathcal{L}_{recons}$.
Repeat the two pre-train steps iteratively until convergence.
for $iter = 1$ to max_iter **do**
    Sample a mini-batch of real video clips $\{\boldsymbol{x}\}$ and generate fake video clips $\{\hat{\boldsymbol{x}}\}$ according to the input.
    // **Train Discriminator**
    Fix $\boldsymbol{\theta}_M$ and $\boldsymbol{\theta}_G$, update discriminator parameters $\boldsymbol{\theta}_D$ by $\frac{\partial \mathcal{L}_{dis}}{\partial \boldsymbol{\theta}_D}$.
    // **Train Motion Guider**
    Compute $\boldsymbol{f}_t^d$ and $\boldsymbol{m}_t$ using real data
    Compute motion guide (learner) loss $\mathcal{L}_M^L$ in Equation (2) using $\boldsymbol{m}_t$ and $\boldsymbol{f}_t^d$ computed above.
    Fix $\boldsymbol{\theta}_G$ and $\boldsymbol{\theta}_D$, update $\boldsymbol{\theta}_M$ by $\frac{\partial \mathcal{L}_M^L}{\partial \boldsymbol{\theta}_M}$.
    // **Train generator**
    Compute the prediction $\hat{\boldsymbol{x}}_{t+1} = G(\boldsymbol{x}_t, \boldsymbol{m}_t; \boldsymbol{\theta}_G)$ and loss $\mathcal{L}_M^T$ in (3) using $\hat{\boldsymbol{m}}_t$ and $\hat{\boldsymbol{f}}_t^d$ from generated samples in Equation (4).
    Fix $\boldsymbol{\theta}_M$ and $\boldsymbol{\theta}_D$, update $\boldsymbol{\theta}_G$ by $\frac{\partial \mathcal{L}_{gen}}{\partial \boldsymbol{\theta}_G}$.
end for

B Experiment Result on Moving MNIST Dataset

The predictions generated by a LMVP model and a DFN model [5] are displayed in Figure 4. Frames in the first line are input sequences and ground truth sequences. Frames generated by a DFN model and our LMVP model are shown in the second and the third line, respectively. Visually, the prediction of LMVP is better than DFN. This is further confirmed quantitatively in Table 3.

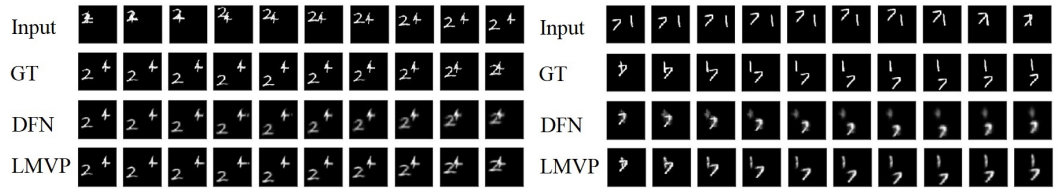

Figure 4: Two prediction examples for the Moving MNIST dataset. From top to down: concatenation of input and ground truth, prediction result of DFN, and prediction result of our model.

PSNR	2	4	6	8	10	2	4	6	8	10
DFN	23.42	21.19	19.01	18.21	18.46	18.70	18.38	18.32	17.91	17.76
LMVP	25.25	22.99	22.01	20.28	20.44	21.73	21.29	21.05	20.85	20.45

Table 3: PSNR scores of prediction frames for each even time step shown in Figure 4 on Moving MNIST dataset.

To demonstrate it more clearly, Figure 5 displays the prediction evaluation of DFN and our model over different time step $(t = T_0 + 1, \cdots, T)$. LMVP achieves higher SSIM and PSNR scores than other baseline models through all time steps.

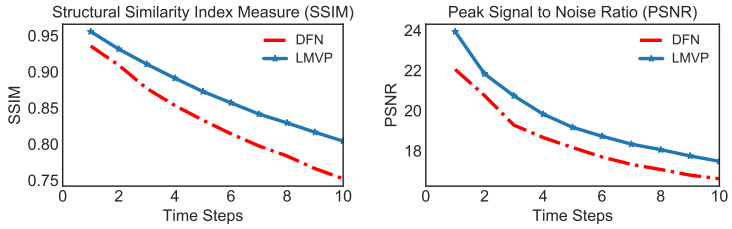

Figure 5: Evaluation result of DFN and ours over different time step (from 1^{st} frame prediction scores to 10^{th} frame prediction scores). The proposed model gets higher SSIM and PSNR scores than baselines through all time steps.