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 [14, 11, 2, 18, 4, 15]. 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.

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) [11, 1, 12, 9]. 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 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]. Mathematically, 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 $x \in \mathbb{R}^{T \times H \times W \times C}$, 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. $x_t$ and $\hat{x}_t$ represent 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 $D$, while generator $G$ uses the current video frame and the output of motion guider $M$ to predict the next frame.

![Figure 1: Model Framework.](image)

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 $F(\cdot)$, followed by a classifier between real/fake samples, parameterized by $\theta_C$. Mathematically, given input video clips $x_{t-c:t}$, we have $D(x_{t-c:t}; \theta_D) = \text{CNN}(F(x_{t-c:t}; \theta_F); \theta_C)$, where $\theta_D = (\theta_F, \theta_C)$. The extracted motion feature from $x_{t-c:t}$ is denoted $f^d_t = F(x_{t-c:t}; \theta_F)$, which is the input of motion guider $M$.

The feature extractor $F$ is implemented as a convolutional network. The output $f^d_t$ is expected to capture motion from $x_{t-c:t}$. The difference $f^d_{t+1} - f^d_t$ is treated as the motion feature between two consecutive frames. In contrast of the direct subtraction of two consecutive frames [11], our motion feature is extracted from two video clips of length $c$. Since it also uses previous video frames, it provides a much more informative guidance signal about dynamics, as well as alleviating the error even if the current prediction fails. Mathematically, the discriminator loss can be written as

$$L_{dis}(\theta_D) = -\mathbb{E}_x[\log D(x; \theta_D)] - \mathbb{E}_x[\log(1 - D(\hat{x}; \theta_D))].$$ (1)

2.2 Learning and teaching game of $M$

To utilize the leaked motion information $f^d_t$ from $D$, we introduce a motion guider module $M$, which is inspired by the leaky GAN model [3]. The structure of $M$ is displayed in the green dotted box in Figure 1. $M$ has a recurrent structure that takes the extracted feature vector $f^d_t$ as input at each time step $t$, and outputs a predicted motion feature $m_t$. 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 $D$ with real video. At time step $t$, $M$ receives the leaked information $f^d_t$ exacted by $D$, and predicts the motion difference between
the video clip of time $t$ and $t+1$, by forcing $\mathbf{m}_t = M(f_t^d; \theta_M)$ close to $f_{t+1}^d - f_t^d$. Denoting the parameters of the motion guider as $\theta_M$, the learner loss function can be written as

$$L^T_M(f_t^d, f_{t+1}^d) = \sum_t \|f_{t+1}^d - (f_t^d + M(f_t^d; \theta_M))\|^2_2. \hspace{1cm} (2)$$

Note that only real video samples are used to update $\theta_M$. The superscript $L$ means “Leaner”.

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

Since the dynamic motion feature is extracted from a real video clip instead of a single frame, $M$ can help alleviate error accumulation, 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 $f_t^s$, while temporal feature $f_t^m$ are computed from dynamic motion features $\mathbf{m}_t$. Note that only predicted samples have their motion guider output $\mathbf{m}_t$ flow back to the generator. The spatio-temporal features $f_t^s$ and $f_t^m$ are concatenated and further fed into the filter network. The next frame is predicted by applying the generated adaptive filter on the current frame. This technique is also known as visual transformation in [14].

As mentioned in Section 2.2, the $M$ is updated during the learning step while fixed during teaching step. When generating the next frame $\hat{x}_{t+1}$, $M$ is fixed and outputs the motion guider $\hat{\mathbf{m}}_t$. Specifically, to ensure $G$ generates the next frames following the guidance of $M$, the feature $f_{t+1}^d$ of the generated frame $\hat{x}_{t+1} = G(\hat{x}_t; \theta_G)$ should be close to the feature taught by the guider $\hat{\mathbf{m}}_t + f_t^d$. The generator is updated by minimizing the following loss function:

$$L^T_M(\theta_G, \theta_F) = \sum_t \| F ([\hat{x}_{t-c+1:t}, G(\hat{x}_t; \theta_G)]; \theta_F) - (f_t^d + \hat{\mathbf{m}}_t) \|^2_2, \hspace{1cm} (3)$$

where $\theta_G$ includes all parameters in the generator $G$. Note that $\hat{x}_t$ and $f_t^d$ are the predicted output and leaked motion feature from step $t-1$. The superscript $T$ in $L^T_M$ indicates $M$ as a “Teacher”.

The total loss function for the generator is

$$L_{\text{gen}}(x_{t+1}, \hat{x}_{t+1}) = L_{\text{recons}}(x_{t+1}, \hat{x}_{t+1}) + \gamma L^T_M. \hspace{1cm} (4)$$

$L_{\text{recons}}$ is the reconstruction loss, 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 $\theta_D$ is updated. Then the motion guider parameters $\theta_M$ are updated using only real samples of $x$. The generated parameters $\theta_G$, together with the leaked feature extractor parameters $\theta_F$, are 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] is 20 frames long, with two handwritten digits bouncing inside a $64 \times 64$ 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) [16] between the ground truth $x$ and the prediction $\hat{x}$. Small values of BCE or large

<table>
<thead>
<tr>
<th>BCE</th>
<th>SSIM</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConvLSTM [17]</td>
<td>$8.96 \times 10^{-2}$</td>
<td>0.61</td>
</tr>
<tr>
<td>FPM [10]</td>
<td>$8.43 \times 10^{-2}$</td>
<td>-</td>
</tr>
<tr>
<td>DFN [5]</td>
<td>$6.89 \times 10^{-2}$</td>
<td>0.83</td>
</tr>
<tr>
<td>LMVP</td>
<td>$6.13 \times 10^{-2}$</td>
<td>0.87</td>
</tr>
</tbody>
</table>

Table 1: Binary cross-entropy, SSIM and PSNR scores results on Moving MNIST dataset.
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 [17, 10, 5, 11]. The predictions from LMVP and DFN [5] are shown in Figure 2. 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 4 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.

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

<table>
<thead>
<tr>
<th></th>
<th>MSE</th>
<th>SSIM</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Frame</td>
<td>$10.34 \times 10^{-3}$</td>
<td>0.83</td>
<td>22.11</td>
</tr>
<tr>
<td>DFN [5]</td>
<td>$3.08 \times 10^{-3}$</td>
<td>0.92</td>
<td>26.95</td>
</tr>
<tr>
<td>LMVP</td>
<td>$2.67 \times 10^{-3}$</td>
<td>0.927</td>
<td>27.23</td>
</tr>
</tbody>
</table>

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 can also alleviate the error accumulated in a standard RNN video predictor. This is more robust against frame-level feature extraction. For structures in video 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 state-of-the-art approaches.
References


