Temporal Attention Unit: Towards Efficient Spatiotemporal Predictive Learning

Cheng Tan*, Zhangyang Gao†, Lirong Wu, Yongjie Xu, Jun Xia, Siyuan Li, Stan Z. Li†
AI Lab, Research Center for Industries of the Future, Westlake University
{tancheng,gaozhangyang,wulirong,xuyongjie,xiajun,lisiyuan,stan.zq.li}@westlake.edu.cn

Abstract

Spatiotemporal predictive learning aims to generate future frames by learning from historical frames. In this paper, we investigate existing methods and present a general framework of spatiotemporal predictive learning, in which the spatial encoder and decoder capture intra-frame features and the middle temporal module catches inter-frame correlations. While the mainstream methods employ recurrent units to capture long-term temporal dependencies, they suffer from low computational efficiency due to their unparallelizable architectures. To parallelize the temporal module, we propose the Temporal Attention Unit (TAU), which decomposes temporal attention into intra-frame statical attention and inter-frame dynamical attention. Moreover, while the mean squared error loss focuses on intra-frame errors, we introduce a novel differential divergence regularization to take inter-frame variations into account. Extensive experiments demonstrate that the proposed method enables the derived model to achieve competitive performance on various spatiotemporal prediction benchmarks.

1. Introduction

The last decade has witnessed revolutionary advances in deep learning across various supervised learning tasks such as image classification [34,52,87], object detection [73,75], computational biology [21, 22, 43, 83, 84], and etc. Despite significant breakthroughs in supervised learning, which relies on large-scale labeled datasets, the potential of unsupervised learning remains largely untapped. Self-supervised learning that designs pretext tasks to produce labels derived from the data itself is recognized as a subset of unsupervised learning. In the context of self-supervised learning, contrastive self-supervised learning [7,8,28,33,85,86,92,104] predicts the noise contrastive estimation from predefined positive or negative pairs, and masked self-supervised learning [17, 32, 45, 51, 57, 102] predicts the masked patches from the visible patches. Unlike these image-level self-supervised learning, spatiotemporal predictive learning that predicts future frames from past frames at the video-level [6, 19, 27, 46, 60, 63].

Figure 1. Comparison of model architectures among common spatiotemporal predictive learning methods. Note that we denote 2D convolutional neural networks as Conv while 3D Conv means 3D convolutional neural networks.

Accurate spatiotemporal predictive learning can benefit broad practical applications in climate change [74, 77], human motion forecasting [91, 107], traffic flow prediction [18, 97], and representation learning [39, 71]. The significance of spatiotemporal predictive learning primarily lies in its potential of exploring both spatial correlation and temporal evolution in the physical world. Moreover, the self-supervised nature of spatiotemporal predictive learning aligns well with human learning styles without a large amount of labeled data. Massive videos can provide a rich source of visual information, enabling spatiotemporal pre-
predictive learning to serve as a generative pre-training strategy [35, 60] for feature representation learning towards diverse downstream visual supervised tasks.

Most of the existing methods [2,3,9,11,12,19,24,29,41,44,65,78,79,82,88,89,93–96,98,100,103] in spatiotemporal predictive learning employ hybrid architectures of convolutional neural networks and recurrent units in which spatial correlation and time evolution can be learned, respectively. Inspired by the success of long short-term memory (LSTM) [36] in sequential modeling, ConvLSTM [78] is a seminal work on the topic of spatiotemporal predictive learning that extends fully connected LSTM to convolutional LSTM towards accurate precipitation nowcasting. PredRNN [95] is an admirable work that proposes Spatiotemporal LSTM (ST-LSTM) units to model spatial appearances and temporal variations in a unified memory pool. This work provides insights on designing typical recurrent units for spatiotemporal predictive learning and inspires a series of subsequent works [4,93,96,98]. E3D-LSTM [94] integrates 3D convolutional neural networks into recurrent units towards good representations with both short-term frame dependencies and long-term high-level relations. PhyDNet [29] introduces a two-branch architecture that involves physical-based PhyCells and ConvLSTMs for performing partial differential equation constraints in latent space. CrevNet [103] proposes an invertible two-way autoencoder based on flow [14,15] and a conditionally reversible architecture for spatiotemporal predictive learning. As shown in Figure 1 (a), we present a general framework consisting of the spatial encoder/decoder and the middle temporal module, abstracted from these methods. Though these spatiotemporal predictive learning methods have different temporal modules and spatial encoders/decoders, they basically share a similar framework.

Based on the general framework, we argue that the temporal module plays an essential role in spatiotemporal predictive learning. While the common choice of the temporal module is the recurrent-based units, we explore a novel parallelizable attention module named Temporal Attention Unit (TAU) to capture time evolution. The proposed temporal attention is decomposed into intra-frame statical attention and inter-frame dynamical attention. Furthermore, we argue that the mean square error loss only focuses on intra-frame differences, and we propose a differential divergence regularization that also cares about the inter-frame variations. Keeping the spatial encoder and decoder as simple as 2D convolutional neural networks, we deliberately implement our proposed TAU modules and surprisingly find the derived model achieves competitive performance as those recurrent-based models. This observation provides a new perspective to improve spatiotemporal predictive learning by parallelizable attention networks instead of commonly used recurrent units.

We conduct experiments on various datasets with different experimental settings: (1) Standard spatiotemporal predictive learning. Quantitative results on various datasets demonstrate our proposed method can achieve competitive performance on standard spatiotemporal predictive learning. (2) Generalization ability. To verify the generalization ability, we train our model on KITTI and test it on the Caltech Pedestrian dataset with different domains. (3) Predict future frames with flexible lengths. We tackle the long-length frames by feeding the predicted frames as the input and find the performance is consistently well. Through the superior performance in the above three experimental settings, we demonstrate that our proposed model can provide a novel manner that learns temporal dependencies without recurrent units.

2. Related works

2.1. Self-supervised learning

Deep learning has been well developed and applied in various fields [5,55,56,109,110]. Learning from massive data enables tremendous progress in supervised learning. By designing pretext tasks and generating labels from the data itself, self-supervised learning obtains supervisory signals. The model learns valuable representations by solving pretext tasks that leverage the underlying structure of the data. Early works on visual self-supervised learning design pretext tasks such as colorization [108], inpainting [69], rotation [26], jigsaw [66]. Contrastive self-supervised learning [7,8,28,33,92,104] is a dominant manner in visual self-supervised learning that aims at a pretext task of grouping similar samples closer and diverse samples away from each other. However, contrastive self-supervised learning is limited by making pairs by multiple images, which affects its ability on small-scale datasets. Masked self-supervised learning [17,32,45,51,57,102], which predicts the masked patches from the visible ones, is another research direction. Although masked pretraining has great success in natural language processing, its applications in visual tasks are challenging. Spatiotemporal predictive learning is another promising branch of self-supervised learning that focuses on video-level information and predicts future frames conditioned on past frames.

In contrast to the above image-level methods, spatiotemporal predictive learning focus on video-level information and predicts future frames conditioned on past frames. By learning the intrinsic motion dynamics, the model is enabled to easily decouple the foreground and background.

2.2. Spatiotemporal predictive learning

Recurrent models have achieved remarkable advances in spatiotemporal predictive learning. Inspired by recurrent neural networks, VideoModeling [62] adopts language

Instead of using recurrent-based methods that are computationally expensive for spatiotemporal predictive learning, we introduce TAU, a model that uses visual attention mechanism to parallelize the temporal evolution without the recurrent structure. There are prior arts that have some similarities with our proposed model. PredCNN [101] and TrajectoryCNN [54] implement pure convolutional neural networks as the temporal module. SimVP [23] is a seminal work that applies blocks of Inception modules with a UNet architecture to learn the temporal evolution. Though their temporal modules are parallelizable, we argue that convolutions alone cannot capture long-term dependencies. Moreover, SimVP provides a simple baseline with minor complex attachment but a large space for further improvements. In general, SimVP first downsamples video sequences to reduce the computation, then uses Inception-UNet to learn essential spatiotemporal relationships, and upsamples the representations to predict future frames. Our work aims to replace the pivotal Inception-UNet with efficient attention modules that promote prediction performance. In this paper, we employ a simple yet effective attention mechanism to enable the temporal module not only to be parallelizable but also to capture long-term time evolution.

3. Methods

3.1. Preliminaries

We formally define the spatiotemporal predictive learning problem as follows. Given a video sequence $\mathcal{X}^{t,T} = \{x_t\}_{t=T+1}$ at time $t$ with the past $T$ frames, we aim to predict the subsequent $T'$ frames $\mathcal{Y}^{t+1,T'} = \{y_{t+1:t+T'}\}$ from time $t+1$, where $x_t \in \mathbb{R}^{C \times H \times W}$ is usually an image with channels $C$, height $H$, and width $W$. In practice, we represent the video sequences as tensors, i.e., $\mathcal{X}^{t,T} \in \mathbb{R}^{T \times C \times H \times W}$ and $\mathcal{Y}^{t+1,T'} \in \mathbb{R}^{T' \times C \times H \times W}$.

The model with learnable parameters $\Theta$ learns a mapping $\mathcal{F}_{\Theta}: \mathcal{X}^{t,T} \mapsto \mathcal{Y}^{t+1,T'}$ by exploring both spatial and temporal dependencies. In our case, the mapping $\mathcal{F}_{\Theta}$ is a neural network model trained to minimize the difference between the predicted future frames and the ground-truth future frames. The optimal parameters $\Theta^*$ are:

$$\Theta^* = \arg\min_{\Theta} \mathcal{L}(\mathcal{F}_{\Theta}(\mathcal{X}^{t,T}), \mathcal{Y}^{t+1,T'})$$

where $\mathcal{L}$ is a loss function that evaluates such differences.

3.2. Overview

We illustrate the overview model in Figure 2 using the input Moving MNIST [81] data as an example.

![Figure 2. The overview architecture of our proposed model.](image)

Striving for simplicity, the model follows the general framework in Figure 1, while the spatial encoder consists of four vanilla 2D convolutional layers, and the spatial decoder consists of four 2D transposed convolutional layers (‘Conv2d’ and ‘ConvTranspose2d’ in PyTorch respectively). We add a residual connection from the first convolutional layer to the last transposed convolutional layer for preserving the spatial-dependent features. Stacks of TAU modules are in the middle of the spatial encoder and decoder to extract temporal-dependent features. Though our model is simple, it can efficiently learn both spatial-dependent and temporal-dependent features without recurrent architectures.

3.3. Temporal Attention Unit

Suppose a batch of input video tensors $\mathcal{B} \in \mathbb{R}^{B \times T \times C \times H \times W}$ with the number of videos $B = |\mathcal{B}|$ is given. In the spatial encoder and decoder, we reshape the sequential input data $B \times T \times C \times H \times W$ as $(B \times T) \times C \times H \times W$ so that only spatial correlations are taken into account. In the temporal module, we reshape the feature $B \times T \times C \times H \times W$ as $B \times (T \times C) \times H \times W$ so that frames are arranged in order on the channel dimension.
We decompose the temporal attention into the intra-frame statical attention and the inter-frame dynamical attention, as shown in Figure 3. Inspired by the recent progress of vision Transformers (ViTs) \cite{17,57} and large kernel convolutions \cite{13,30,58}, we propose to employ small kernel depth-wise convolutions (DW Conv), depth-wise convolutions with dilations (DW-D Conv), and $1 \times 1$ convolutions to model the large kernel convolutions. Through the obtained large receptive field on intra-frames, the statical attention is able to capture long-range dependencies. However, the statical attention alone is not enough for learning temporal evolutions along the timeline. Thus, we employ the dynamical attention that learns the attention weights of channels in a squeeze-and-excitation manner \cite{38}. The final attention is the product of dynamical attention and statical attention.

![Figure 3. The intra-frame statical attention and the inter-frame dynamical attention.](image)

We show the detailed scheme of our model in Figure 4. The proposed TAU module can be formally expressed as:

\begin{align}
    SA &= \text{Conv}_{1 \times 1}(\text{DW-D Conv}(\text{DW Conv}(H))), \\
    DA &= \text{FC}(\text{AvgPool}(H)), \\
    H' &= (SA \odot DA) \odot H, 
\end{align}

where $H \in \mathbb{R}^{B \times (T \times C') \times H \times W}$ is the hidden feature that will be fed into the TAU module, $SA \in \mathbb{R}^{B \times (T \times C') \times H \times W}, DA \in \mathbb{R}^{B \times (T \times C') \times 1 \times 1}$ denote the statical and dynamical attention, FC and AvgPool are fully connected layers and the average pooling. We represent the Kronecker product by $\odot$ and the Hadamard product by $\odot$.

### 3.4. Differential Divergence Regularization

To improve the prediction of our model, we further propose a differential divergence regularization that forces the model to learn the differences between consecutive frames and be aware of the inherent variation.

Given the predicted frames $\hat{Y} = F_{\Theta}(X') \in \mathbb{R}^{T' \times C \times H \times W}$ and its corresponding ground-truth frames $Y$, we first calculate their forward difference $\Delta \hat{Y}, \Delta Y \in \mathbb{R}^{(T'-1) \times C \times H \times W}$, where:

\begin{align}
    \Delta \hat{Y}_i &= \hat{Y}_{i+1} - \hat{Y}_i, \\
    \Delta Y_i &= Y_{i+1} - Y_i. 
\end{align}

Then, we transform the differences into probabilities by the softmax function on the channel, height, and width dimension and obtain $\sigma(\Delta \hat{Y}), \sigma(\Delta Y)$, where:

\begin{align}
    \sigma(\Delta \hat{Y})_{i,j,k,l} &= \frac{\exp(\Delta \hat{Y}_{i,j,k,l}/\tau)}{\sum_{j'=1}^{C} \sum_{k'=1}^{H} \sum_{l'=1}^{W} \exp(\Delta \hat{Y}_{i,j',k',l'}/\tau)}, \\
    \sigma(\Delta Y)_{i,j,k,l} &= \frac{\exp(\Delta Y_{i,j,k,l}/\tau)}{\sum_{j'=1}^{C} \sum_{k'=1}^{H} \sum_{l'=1}^{W} \exp(\Delta Y_{i,j',k',l'}/\tau)},
\end{align}

and $\tau$ represents the temperature parameter which we empirically set as 0.1 to sharpen the probability distribution. Through the competition mechanism of the softmax function \cite{10,70,99}, the high difference frames are penalized.

Thus, the differential divergence regularization $L_{\text{reg}}$ is defined as the Kullback-Leibler divergence between the probability distributions $\sigma(\Delta \hat{Y})$ and $\sigma(\Delta Y)$:

\begin{align}
    L_{\text{reg}}(\hat{Y}, Y) &= D_{KL}(\sigma(\Delta \hat{Y}) \mid \mid \sigma(\Delta Y)) \\
    &= \sum_{i=1}^{T'-1} \sigma(\Delta \hat{Y}_i) \log \frac{\sigma(\Delta \hat{Y}_i)}{\sigma(\Delta Y_i)}. \quad (5)
\end{align}

Our model is trained end-to-end in a fully unsupervised manner, and the overall objective function consists of the mean square error loss and the differential divergence regularization weighted by a constant $\alpha$:

\begin{align}
    L &= \sum_{i=1}^{T'} ||\hat{Y}_i - Y_i||^2 + \alpha L_{\text{reg}}(\hat{Y}, Y), \quad (6)
\end{align}

where the first term focuses on intra-frame-level differences, and the second regularization term focuses on inter-frame-level variations.
4. Experiments

In this section, we present experiments that demonstrate the effectiveness of our proposed method. The experiments are conducted on various datasets with different settings to validate our proposed model from three aspects:

- Standard spatiotemporal predictive learning (Section 4.2). We recognize the prediction problem of the same number of input and output frames as the standard spatiotemporal predictive learning. We evaluate the performance on standard spatiotemporal predictive learning and compare our model with state-of-the-art methods with Moving MNIST [81] and TaxiBJ [106] datasets.

- Generalization ability across different datasets (Section 4.3). Generalizing the learned knowledge to other domains is a challenge in unsupervised learning. We investigate the such ability of our method by training the model on the KITTI [25] dataset and evaluating it on the Caltech Pedestrian [16] dataset.

- Predicting frames with flexible lengths (Section 4.4). One of the advantages of recurrent units is that they can easily handle flexible-length frames like the KTH dataset [76]. Our work tackles the long-length frame prediction by imitating recurrent units that feed predicted frames as the input and recursively produce long-term predictions.

4.1. Experimental Setups

Datasets We quantitatively evaluate our model on the following datasets for both synthetic and real-world scenarios:

- Moving MNIST [81] is a synthetic dataset consisting of two digits independently moving within the 64 × 64 grid and bouncing off the boundary. It is a standard benchmark in spatiotemporal predictive learning.

- TaxiBJ contains the trajectory data in Beijing collected from taxicab GPS with two channels, i.e., inflow or outflow defined in [106]. Following the previous works [29,98], we normalize the data into [0, 1].

- KTH [76] contains 25 individuals performing six types of actions. Following [90, 94], we use person 1-16 for training and 17-25 for testing. Models are trained to predict the next 20 or 40 frames from the previous 10 observations.

- Caltech Pedestrian is a driving dataset focusing on detecting pedestrians. It consists of approximately 10 hours of 640 × 480 videos taken from vehicles driving through regular traffic in an urban environment. We follow the same protocol of PredNet [61] and CrevNet [103] for pre-processing, training, and evaluation.

We summarize the statistics of the above datasets in Table 1, including the number of training samples \( N_{\text{train}} \) and the number of testing samples \( N_{\text{test}} \).

<table>
<thead>
<tr>
<th>( N_{\text{train}} )</th>
<th>( N_{\text{test}} )</th>
<th>( (C, H, W) )</th>
<th>( T )</th>
<th>( T' )</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMNIST 10000</td>
<td>10000</td>
<td>(1, 64, 64)</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>TaxiBJ 19627</td>
<td>1334</td>
<td>(2, 32, 32)</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>KTH 5200</td>
<td>3167</td>
<td>(1, 128, 128)</td>
<td>10</td>
<td>20 or 40</td>
</tr>
<tr>
<td>Caltech 2042</td>
<td>1983</td>
<td>(3, 128, 160)</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

Measurement Following [29, 103], we employ Mean Squared Error (MSE), Mean Absolute Error (MAE), Structure Similarity Index Measure (SSIM), and Peak Signal to Noise Ratio (PSNR) to evaluate the quality of predictions. MSE and MAE estimate the absolute pixel-wise errors, SSIM measures the similarity of structural information within the spatial neighborhoods, and PSNR is an expression for the ratio between the maximum possible power of a signal and the power of distorted noise.

Implementation details We implement the proposed method with the Pytorch framework and conduct experiments on a single NVIDIA-V100 GPU. The model is trained with a mini-batch of 16 video sequences while the AdamW optimizer is utilized with a learning rate of 0.01 and a weight decay of 0.05.

4.2. Standard spatiotemporal predictive learning

4.2.1 Moving MNIST

This dataset is a standard benchmark in spatiotemporal predictive learning. We evaluate our proposed method against strong recent baselines, including competitive recurrent architectures: ConvLSTM [78], PredRNN [95], PredRNN++ [93], MIM [98], LMC [50], E3D-LSTM [94], Conv-TT-LSTM [82], and CrevNet [103]. We also compare the method DDPAE [37] that is specifically designed for this dataset. The quantitative results are reported in Table 2, and qualitative visualizations of the predicted results are shown in Figure 5.

Our proposed method significantly outperforms all the baselines above under three different metrics. The performance gain is large with respect to state-of-the-art recurrent methods.
Table 2. Quantitative results of different methods on the Moving MNIST dataset (10 → 10 frames).

<table>
<thead>
<tr>
<th>Method</th>
<th>Moving MNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MSE↓</td>
</tr>
<tr>
<td>ConvLSTM [78]</td>
<td>103.3</td>
</tr>
<tr>
<td>VPN [44]</td>
<td>64.1</td>
</tr>
<tr>
<td>PredRNN [95]</td>
<td>56.8</td>
</tr>
<tr>
<td>PredRNN++ [93]</td>
<td>46.5</td>
</tr>
<tr>
<td>MIM [98]</td>
<td>44.2</td>
</tr>
<tr>
<td>LMC [50]</td>
<td>41.5</td>
</tr>
<tr>
<td>E3D-LSTM [94]</td>
<td>41.3</td>
</tr>
<tr>
<td>Conv-TT-LSTM [82]</td>
<td>53.0</td>
</tr>
<tr>
<td>DDPAE [37]</td>
<td>38.9</td>
</tr>
<tr>
<td>PhyDNet [29]</td>
<td>24.4</td>
</tr>
<tr>
<td>SimVP [23]</td>
<td>23.8</td>
</tr>
<tr>
<td>Crevnet [103]</td>
<td>22.3</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>19.8</strong></td>
</tr>
</tbody>
</table>

4.2.2 TaxiBJ

We evaluate our proposed model on a complicated real-world dataset, TaxiBJ [106]. Driven by human consciousness, the complex real-world traffic flows requires modeling transport phenomena and traffic diffusion for prediction. Due to the spatiotemporal nature of the traffic forecasting task, we straightforwardly implement our model for it.

The qualitative visualizations of the predicted results are shown in Figure 6, and the quantitative results are reported in Table 3. Though the given frames are quite different from the future frames, our model can still accurately produce reliable frames. The difference between target frames and predicted frames is mainly located in central spots, but the overall trend is approximating the ground truth. It can also be observed that the quantitative results are consistently well, which demonstrates the practical application value in real-world scenarios.

4.3. Generalization across different datasets

The generalization ability is one of the fundamental problems in artificial intelligence technology. Traditional supervised learning suffers from its poor generalization of
Table 3. Quantitative results of different methods on the TaxiBJ dataset (4 → 4 frames).

<table>
<thead>
<tr>
<th>Method</th>
<th>MSE × 100↓</th>
<th>MAE↓</th>
<th>SSIM↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConvLSTM [78]</td>
<td>48.5</td>
<td>17.7</td>
<td>0.978</td>
</tr>
<tr>
<td>PredRNN [95]</td>
<td>46.4</td>
<td>17.1</td>
<td>0.971</td>
</tr>
<tr>
<td>PredRNN++ [93]</td>
<td>44.8</td>
<td>16.9</td>
<td>0.977</td>
</tr>
<tr>
<td>MIM [98]</td>
<td>42.9</td>
<td>16.6</td>
<td>0.971</td>
</tr>
<tr>
<td>E3D-LSTM [94]</td>
<td>43.2</td>
<td>16.9</td>
<td>0.979</td>
</tr>
<tr>
<td>PhyDNet [29]</td>
<td>41.9</td>
<td>16.2</td>
<td>0.982</td>
</tr>
<tr>
<td>SimVP [23]</td>
<td>41.4</td>
<td>16.2</td>
<td>0.982</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>34.4</strong></td>
<td><strong>15.6</strong></td>
<td><strong>0.983</strong></td>
</tr>
</tbody>
</table>

Self-supervised learning aims to learn robust representations based on unlabeled data and evaluates the generalization ability of the learned model. While contrastive self-supervised learning and masked self-supervised learning in visual tasks usually evaluate such generalization ability by downstream tasks, we evaluate it by the prediction results across different datasets in spatiotemporal predictive learning.

4.4. Predicting frames with flexible lengths

Though recurrent units are adept at handling flexible-length frames, our model can also easily tackle such problems by imitating recurrent units that feed predicted frames as the input and recursively produce predictions. For this KTH dataset, our model is trained to predict the next 20 or 40 frames from the given 10 observations. Moreover, this dataset contains six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) performed several times by 25 subjects in four different scenarios: outdoors, outdoors with scale variations, outdoors with different clothes and indoors. The difficulty of this human motion prediction task not only lies in its flexible lengths of predicted frames but also in its complex dynamics involving the randomness of human consciousness. Following [95], we use the Peak Signal Noise Ratio (PSNR) and the Structural Similarity Index Measure (SSIM) as evaluation metrics to measure the framewise prediction quality from the perceptive view. The detailed quantitative results are shown in Table 5.

Table 4. Quantitative results of different methods on the Caltech Pedestrian dataset (10 → 1 frame).

<table>
<thead>
<tr>
<th>Method</th>
<th>SSIM↑</th>
<th>PSNR↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>BeyondMSE [64]</td>
<td>0.847</td>
<td>-</td>
</tr>
<tr>
<td>MCnet [90]</td>
<td>0.879</td>
<td>-</td>
</tr>
<tr>
<td>DVF [59]</td>
<td>0.897</td>
<td>26.2</td>
</tr>
<tr>
<td>Dual-GAN [53]</td>
<td>0.899</td>
<td>-</td>
</tr>
<tr>
<td>CtrlGen [31]</td>
<td>0.900</td>
<td>26.5</td>
</tr>
<tr>
<td>PredNet [61]</td>
<td>0.905</td>
<td>27.6</td>
</tr>
<tr>
<td>ContextVP [4]</td>
<td>0.921</td>
<td>28.7</td>
</tr>
<tr>
<td>GAN-VGG [80]</td>
<td>0.916</td>
<td>-</td>
</tr>
<tr>
<td>G-VGG [80]</td>
<td>0.917</td>
<td>-</td>
</tr>
<tr>
<td>SDC-Net [72]</td>
<td>0.918</td>
<td>-</td>
</tr>
<tr>
<td>rCycleGan [47]</td>
<td>0.919</td>
<td>29.2</td>
</tr>
<tr>
<td>DPG [20]</td>
<td>0.923</td>
<td>28.2</td>
</tr>
<tr>
<td>G-MAE [80]</td>
<td>0.923</td>
<td>-</td>
</tr>
<tr>
<td>GAN-MAE [80]</td>
<td>0.923</td>
<td>-</td>
</tr>
<tr>
<td>CrevNet [103]</td>
<td>0.925</td>
<td>29.3</td>
</tr>
<tr>
<td>STMFANet [41]</td>
<td>0.927</td>
<td>29.1</td>
</tr>
<tr>
<td>SimVP [23]</td>
<td>0.940</td>
<td>33.1</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>0.946</strong></td>
<td><strong>33.7</strong></td>
</tr>
</tbody>
</table>

Following the previous works [53, 61, 103], we train the model using the raw video sequences from the KITTI dataset and evaluate the model by Caltech Pedestrian dataset that is made to match the frame rate and image size (128 × 160) of the KTTI dataset. The final prediction is made for the next frame after a 10-frame warm-up.

We show the qualitative visualization in Figure 7 and report the quantitative results in Table 4. Note that some of the baseline results are copied from [68]. It can be seen that our proposed method achieves state-of-the-art performance under both SSIM and PSNR metrics in this generalization evaluation task. Moreover, our model shows significantly robust predictions in terms of the variation of illumination and lane line, suggesting its practical applications in autonomous vehicles.

Figure 7. Qualitative visualization of predicted results on Caltech dataset. The differences between the ground truth and the predicted frames are visualized in the last row.
Table 5. Quantitative results of different methods on the KTH dataset (10 → 20 and 10 → 40 frames).

<table>
<thead>
<tr>
<th>Method</th>
<th>KTH (10 → 20)</th>
<th>KTH (10 → 40)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSIM↑ PSNR↑</td>
<td>SSIM↑ PSNR↑</td>
<td></td>
</tr>
<tr>
<td>MCnet [90]</td>
<td>0.804 25.95</td>
<td>0.73 23.89</td>
</tr>
<tr>
<td>ConvLSTM [78]</td>
<td>0.712 23.58</td>
<td>0.639 22.85</td>
</tr>
<tr>
<td>SAVP [48]</td>
<td>0.746 25.38</td>
<td>0.701 23.97</td>
</tr>
<tr>
<td>VPN [44]</td>
<td>0.746 23.76</td>
<td>– –</td>
</tr>
<tr>
<td>DFN [40]</td>
<td>0.794 27.26</td>
<td>0.652 23.01</td>
</tr>
<tr>
<td>fRNN [67]</td>
<td>0.771 26.12</td>
<td>0.678 23.77</td>
</tr>
<tr>
<td>Znet [105]</td>
<td>0.817 27.58</td>
<td>– –</td>
</tr>
<tr>
<td>SV2Pv [1]</td>
<td>0.838 27.79</td>
<td>0.789 26.12</td>
</tr>
<tr>
<td>PredRNN [95]</td>
<td>0.839 27.55</td>
<td>0.703 24.16</td>
</tr>
<tr>
<td>VarNet [42]</td>
<td>0.843 28.48</td>
<td>0.739 25.37</td>
</tr>
<tr>
<td>SAVP-VAE [48]</td>
<td>0.852 27.77</td>
<td>0.811 26.18</td>
</tr>
<tr>
<td>PredRNN++ [93]</td>
<td>0.865 28.47</td>
<td>0.741 25.21</td>
</tr>
<tr>
<td>MSNET [49]</td>
<td>0.876 27.08</td>
<td>– –</td>
</tr>
<tr>
<td>E3d-LSTM [94]</td>
<td>0.879 29.31</td>
<td>0.810 27.24</td>
</tr>
<tr>
<td>STMFA Net [41]</td>
<td>0.893 29.85</td>
<td>0.851 27.56</td>
</tr>
<tr>
<td>SimVP [23]</td>
<td>0.905 33.72</td>
<td>0.886 32.93</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>0.911 34.13</strong></td>
<td><strong>0.897 33.01</strong></td>
</tr>
</tbody>
</table>

4.5. Empirical Running Time

TAU benefits from the parallelizable computation architecture, which leads to fast convergence and high training speed. We evaluate our efficiency by measuring the running time against state-of-the-art spatiotemporal predictive learning methods.

![Learning curve comparison](image)

Figure 8. The learning curve comparison between state-of-the-art methods and ours (evaluated by MSE). The red dotted line denotes MSE 35.0, and only the first 50 epochs are shown.

The experiments are conducted on a single Tesla V100 GPU, and the batch size is set as 16. For the Moving MNIST dataset, CrevNet [103] needs about 30 minutes per epoch, and PhyDNet [29] needs about 7 minutes per epoch. Our model only requires 2.5 minutes per epoch. Furthermore, our method is able to converge at a rapid rate. As shown in Figure 8, on the Moving MNIST dataset, our model can achieve MSE 35.0 with only 50 epochs, while CrevNet and PhyDNet are far from this performance.

4.6. Computational Cost and Ablation Study

We compare the performance and computational cost with state-of-the-art methods in the first several rows in Table 6. Our model achieves superior results with better performance and much lower Flops. We also conduct ablation studies and summarize the results in Table 6. It can be seen that replacing TAU with the same number of convolutional blocks with vanilla 3 × 3 convolutions (Conv Baseline) significantly degrades the performance. Training our model without differential divergence regularization (w/o DDR) will also weaken the prediction results. Both the TAU module and differential divergence regularization are useful. We also find that SA and DA of the TAU module play important roles in better performance.

<table>
<thead>
<tr>
<th>Method</th>
<th>M-MNIST</th>
<th>TaxiBJ (×100)</th>
<th>Flops(G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PredRNN</td>
<td>56.8</td>
<td>46.4</td>
<td>115.94</td>
</tr>
<tr>
<td>PredRNN++</td>
<td>46.4</td>
<td>44.8</td>
<td>171.73</td>
</tr>
<tr>
<td>MIM</td>
<td>44.2</td>
<td>42.9</td>
<td>179.17</td>
</tr>
<tr>
<td>E3DLSTM</td>
<td>41.3</td>
<td>43.2</td>
<td>298.85</td>
</tr>
<tr>
<td>SimVP</td>
<td>23.8</td>
<td>41.4</td>
<td>19.43</td>
</tr>
<tr>
<td><strong>Conv Baseline</strong></td>
<td>58.9</td>
<td>43.5</td>
<td>6.11</td>
</tr>
<tr>
<td>Ours w/o SA</td>
<td>23.2</td>
<td>40.8</td>
<td>15.30</td>
</tr>
<tr>
<td>Ours w/o DA</td>
<td>22.4</td>
<td>38.4</td>
<td>15.96</td>
</tr>
<tr>
<td>Ours w/o DDR</td>
<td>21.1</td>
<td>37.7</td>
<td>15.96</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>19.8</strong></td>
<td><strong>34.4</strong></td>
<td><strong>15.96</strong></td>
</tr>
</tbody>
</table>

5. Conclusion

In this paper, we present a general framework of spatiotemporal predictive learning and propose an attention-based temporal module to replace the common-used recurrent units. By decomposing the temporal module into the intra-frame statical attention and the inter-frame dynamical attention, our proposed TAU module can achieve competitive performance across various experimental settings and datasets. Moreover, a novel differential divergence regularization is proposed to overcome the drawback of MSE loss that only considers the intra-frame error. In summary, our work highlights the importance of both intra-frame and inter-frame variations that enable the model to capture long-term relations and provide a new paradigm of efficient spatiotemporal predictive learning.

**Acknowledgement.** We thank the anonymous reviewers for their constructive and helpful reviews. This work was supported by the National Key R&D Program of China (2022ZD0115100), the National Natural Science Foundation of China (U21A20427), the Competitive Research Fund (WU2022A009) from the Westlake Center for Synthetic Biology and Integrated Bioengineering.
References


[27] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative varia-
[32] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 1, 2


