SimVP: Towards Simple yet Powerful Spatiotemporal Predictive Learning

Cheng Tan*, Zhangyang Gao*, and Stan Z. Li, *Fellow, IEEE*

Abstract—Recent years have witnessed remarkable advances in spatiotemporal predictive learning, incorporating auxiliary inputs, elaborate neural architectures, and sophisticated training strategies. Although impressive, the system complexity of mainstream methods is increasing as well, which may hinder the convenient applications. This paper proposes SimVP, a simple spatiotemporal predictive baseline model that is completely built upon convolutional networks without recurrent architectures and trained by common mean squared error loss in an end-to-end fashion. Without introducing any extra tricks and strategies, SimVP can achieve superior performance on various benchmark datasets. To further improve the performance, we derive variants with the gated spatiotemporal attention translator from SimVP that can achieve better performance. We demonstrate that SimVP has strong generalization and extensibility on real-world datasets through extensive experiments. The significant reduction in training cost makes it easier to scale to complex scenarios. We believe SimVP can serve as a solid baseline to benefit the spatiotemporal predictive learning community.

Index Terms—Spatiotemporal predictive learning, self-supervised learning, convolutional neural networks, computer vision

1 Introduction

A wise person can foresee the future, and so should an intelligent vision model. Due to spatiotemporal information implying the inner laws of the chaotic world, spatiotemporal predictive learning has recently attracted lots of attention [1], [2], [3], [4]. Struggling with its inherent complexity and randomness, many exciting works have emerged. These methods achieve impressive performance gain by introducing novel operators like typical recurrent units [1], [3], [5], [6], [7], [8] or transformers [9], [10], delicate architectures like autoregressive [1], [4], [11], [12], [13] or normalizing flow [14], and distinctive training strategies such as adversarial training [15], [16], [17], [18], [19], [20], [21], [22]. However, there is relatively little understanding of their necessity for outstanding performance since many methods use different metrics and datasets. Moreover, the increasing model complexity and the lack of a unified framework further aggravate this dilemma. A natural question arises: *Is it possible to develop a simple model to achieve comparable performance and provide a comprehensive understanding of the underlying dynamics?*

To clearly explore spatiotemporal predictive learning, we divide primary methods into four categories as shown in Fig. 1, i.e., (a) RNN-RNN-RNN, (b) CNN-RNN-CNN, (c) CNN-ViT-CNN, and (d) CNN-CNN-CNN. For the former three categories (a-c), models generate predictions frame by frame with the previous output for capturing temporal evolution. For the latter category (d), models generate predictions in a one-shot manner and potentially employ Unet connections between the convolutional layers.

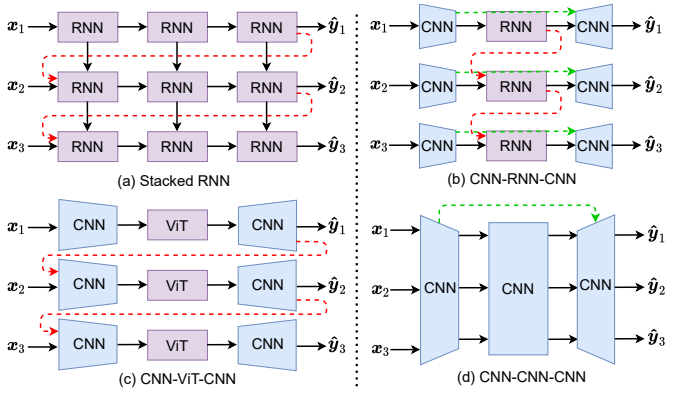


Fig. 1. Major categories of the architectures for spatiotemporal predictive learning. The red and blue dotted line are available to learn the temporal evolution and spatial dependency. Our proposed SimVP belongs to (d) CNN-CNN-CNN, which can outperform other state-of-the-art methods.

As shown in Table 1, we collect representative works from which we observe that recurrent architectures (Fig. 1 a-b) have been favored. In this context, numerous dedicated recurrent units are proposed. Inspired by the success of long short-term memory (LSTM) [51] in sequential modeling, ConvLSTM [1] is a seminal work on the topic of spatiotemporal predictive learning that extends fully connected LSTM to convolutional LSTM. PredRNN [6] 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 [3], [5], [7], [8], [52]. PhyDNet [47] introduces a two-branch architecture that involves physical-based PhyCells and ConvLSTMs for partial differential equation constraints. CrevNet [14] proposes an invertible

- * *Equal contribution.*
- *Cheng Tan and Zhangyang Gao are with Zhejiang University, Hangzhou, China, and aslo with the AI Lab, School of Engineering, Westlake University, Hangzhou, China. Email: {tancheng, gaozhangyang}@westlake.edu.cn.*
- *Stan Z. Li is with the AI Lab, School of Engineering, Westlake University, Hangzhou, China. Email: Stan.ZQ.Li@westlake.edu.cn.*

TABLE 1
Representative spatiotemporal predictive learning works since 2014.

	(a) RNN-RNN-RNN	(b) CNN-RNN-CNN	(c) CNN-ViT-CNN	(d) CNN-CNN-CNN
2014-2015	[1], [4], [23]	[24], [25]	-	[15]
2016-2017	[6], [26], [27], [28]	[2], [29], [30], [31], [32]	-	[33], [34], [35]
2018-2019	[3], [7], [36], [37], [38], [39]	[5], [14], [40], [41], [42]	[9]	[43], [44], [45]
2020-2022	[8]	[46], [47], [48]	[10]	[49], [50]

two-way autoencoder based on flow [53], [54] and a conditionally reversible architecture.

In contrast, purely CNN-based models (Fig. 1 d) are not as favored as the above RNN-based approaches (Fig. 1 a-b). Moreover, the existing methods usually require fancy techniques, e.g., adversarial training [44], teacher-student distilling [50], and optical flow [43]. We admire their significant advancements but expect to exploit how far a simple model can go. To this end, we start with the most common components, i.e., convolutional networks, shortcut connections, and train the model with the mean square error (MSE) loss in an end-to-end manner. Striving for simplicity, we propose a simple yet effective spatiotemporal predictive learning model, namely SimVP. Without extra tricks and complex strategies, SimVP can achieve state-of-the-art performance on various benchmark datasets. Its inherent simplicity provides an excellent potential for machine perception and understanding.

A preliminary version of this work was published in [55]. This journal paper extends it in the following aspects: 1) We develop variants of SimVP that can achieve even better performance. 2) We reproduce the mainstream spatiotemporal predictive learning methods into a unified framework and systematically evaluate performance on the common benchmark Moving MNIST dataset in consideration of computational cost and time complexity. 3) Additional experiments on climate prediction are conducted to analysis the robustness of SimVP and its variants. We release our code at github.com/chengtan9907/SimVPv2.

2 Related Work

2.1 Self-supervised learning

Recent years have witnessed tremendous progress in developing supervised learning methods that can learn from massive amounts of carefully labeled data. However, there is a limit to how far artificial intelligence can go with supervised learning alone. The bottleneck lies in traditional supervised learning is the limited labeled data compared to the abundant unlabeled. In contrast to supervised learning, self-supervised learning is a promising way to approximate human-level intelligence, enabling the model to take advantage of the ubiquitous unlabeled data.

Self-supervised learning obtains supervisory signals by designing pretext tasks and producing labels derived from the data itself. Through solving pretext tasks, the model leverages the underlying structure of the data and learns valuable representations. Early works on visual self-supervised learning design pretext tasks like colorization [56], inpainting [57], rotation [58], and jigsaw [59]. Contrastive self-supervised learning [60], [61], [62], [63], [64], [65] aims at a pretext task that grouping similar samples closer and diverse samples away from each other and has recently become a dominant manner in visual self-supervised learning. However, contrastive self-supervised learning suffers from making pairs by multiple images, which limits its ability on small-scale datasets. Some researchers also turn their attention to masked self-supervised learning [66], [67], [68], [69], [70], [71], which predicts the masked patches from the visible ones. Though masked pretraining has achieved great success in natural language processing, its applications in visual tasks are non-trivial.

Spatiotemporal predictive learning is another promising branch of self-supervised learning. 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 decouple the foreground and background easily. As shown in Fig. 2, masked self-supervised learning work on the space axis, and spatiotemporal predictive learning work on the time axis.

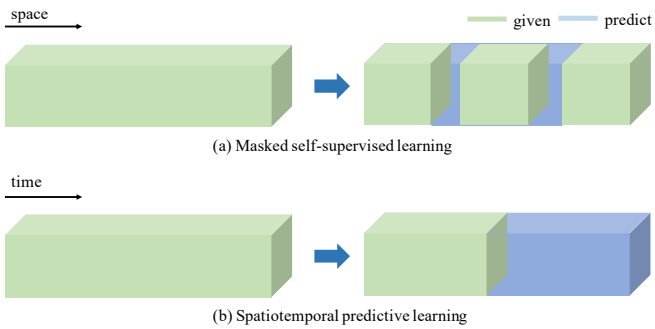


Fig. 2. Comparison of masked self-supervised learning and spatiotemporal predictive learning. We use green for the given information and blue for the information expected to be predicted. For masked self-supervised learning, images are corrupted. For spatiotemporal predictive learning, videos are divided into past and future frames.

2.2 Spatiotemporal predictive learning

2.2.1 RNN-RNN-RNN

As shown in Fig. 1 (a), this kind of method stacks RNN to make predictions. They usually design novel RNN modules (local) and overall architectures (global). Recurrent Grammar Cells [23] stacks multiple gated autoencoders in a recurrent pyramid structure. ConvLSTM [1] extends fully connected LSTMs to have convolutional computing structures to capture spatiotemporal correlations. PredRNN [6] suggests simultaneously extracting and memorizing spatial and temporal representations. PredRNN++ [7] proposes a gradient highway unit to alleviate the gradient propagation

difficulties for capturing long-term dependency. MIM [3] uses a self-renewed memory module to model both the non-stationary and stationary properties of the video. dGRU [38] shares state cells between encoder and decoder to reduce the computational and memory costs. Due to the excellent flexibility and accuracy, these methods play fundamental roles in spatiotemporal predictive learning. PredRNNv2 [8] extends PredRNN by introducing a decoupling loss and a reverse scheduled sampling method.

2.2.2 CNN-RNN-CNN

This framework projects video frames to the latent space and employs RNN to predict the future latent states, seeing Fig. 1 (b). In general, they focus on modifying the LSTM and encoding-decoding modules. Spatio-Temporal video autoencoder [24] incorporates ConvLSTM and an optical flow predictor to capture changes over time. Conditional VRNN [42] combines CNN encoder and RNN decoder in a variational generating framework. E3D-LSTM [5] applies 3D convolution for encoding and decoding and integrates it into latent RNNs for obtaining motion-aware and short-term features. CrevNet [14] proposes using CNN-based normalizing flow modules to encode and decode inputs for information-preserving feature transformations. PhyDNet [47] models physical dynamics with CNN-based PhyCells. Recently, this framework has attracted considerable attention because the CNN encoder can extract decent and compressed features for accurate and efficient prediction.

2.2.3 CNN-ViT-CNN

This framework introduces Vision Transformer (ViT) to model latent video dynamics. By extending language transformer [72] to ViT [69], a wave of research has been sparked recently. As to image transformers, DeiT [73] and Swin Transformer [71] have achieved state-of-the-art performance on various vision tasks. The great success of image transformers has inspired the investigation of video transformers. VTN [74] applies sliding window attention on temporal dimension following a 2D spatial feature extractor. TimeSformer and ViViT [75], [76] study different space-time attention strategies and suggest that separately applying temporal and spatial attention can achieve superb performance. MViT [77] extracts multiscale pyramid features to provide state-of-the-art results on SSv2. Video Swin Transformer [78] expands Swin Transformer from 2D to 3D, where the shiftable local attention schema leads to a better speed-accuracy trade-off. Most models above are designed for video classification; works about spatiotemporal predictive learning [9], [10] using ViT are still limited. More related works may emerge in the future.

2.2.4 CNN-CNN-CNN

The CNN-based framework is not as popular as the previous three because it is so simple that complex modules and training strategies are usually required to improve performance. DVF [34] suggests learning the voxel flow by CNN autoencoder to reconstruct a frame by borrowing voxels from nearby frames. PredCNN [45] combines cascade multiplicative units (CMU) with CNN to capture inter-frame dependencies. DPG [43] disentangles motion and background via a flow predictor and a context generator. [50] encodes RGB frames from the past and decodes the future semantic segmentation using CNN and teacher-student distilling. [49] uses a hierarchical neural model to make predictions at different spatial resolutions and train the model with adversarial and perceptual loss functions. While these approaches have progressed, we wonder what happens if the complexity is reduced. Is there a solution that is much simpler but can exceed or match the performance of state-of-the-art methods?

We have witnessed a wide range of terrific methods that can achieve outstanding performance. However, as the models become more complex, understanding their performance gain is an inevitable challenge, and scaling them into large datasets is intractable. In this work, we aim to build a simple model based on common convolutional modules and see how far the simple model can go in spatiotemporal predictive learning.

3 Preliminaries

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

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

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

4 Method

4.1 Motivation

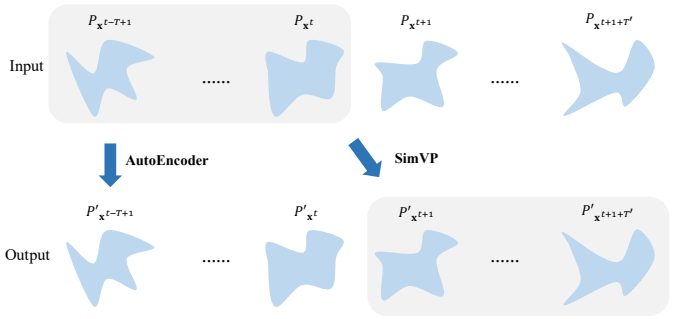


Fig. 3. The schematic diagram of the autoencoder and our proposed SimVP. While the autoencoder focuses on a single frame at a static time, SimVP concerns a sequence of frames at a dynamic time. The first row denotes the ground-truth frames, and the second denotes the predicted frames. From left to right, the data changes over time.

Inspired by the autoencoder that reconstructs a single frame image and captures spatial correlations, we aim to design an autoencoder-like architecture that inputs the past frames and outputs the future frames while preserving the temporal dependencies. As shown in Fig. 3, the traditional autoencoder focuses on single frame image reconstruction at a static time and learns a mapping $\mathcal{G}_\Phi : \mathbf{x} \mapsto \mathbf{x}$ to minimize the divergence between the decoded output probability distribution $P'_{\mathbf{x}} = \mathcal{G}_\Phi(\mathbf{x})$ and the encoded input probability distribution $P_{\mathbf{x}}$. Its optimal parameters Φ^* are:

$$\Phi^* = \arg\min_{\Phi} Div(P_{\mathbf{x}}, P'_{\mathbf{x}}), \tag{2}$$

where Div denotes a specific divergence measure. In practice, we usually minimize the MSE loss between $\mathbf{x}$ and $\mathcal{G}_\Phi(\mathbf{x})$ as follows:

$$\Phi^* = \arg\min_{\Phi} \|\mathbf{x} - \mathcal{G}_\Phi(\mathbf{x})\|^2. \tag{3}$$

Similar to the autoencoder, SimVP learns a mapping $\mathcal{F}_\Theta : \mathcal{X}^{t,T} \mapsto \mathcal{Y}^{t+1,T'}$ to encode the past frames $\mathcal{X}^{t,T}$ and decode the future frames $\mathcal{Y}^{t+1,T'}$ and thus extends the autoencoder-like framework along the time axis. The optimal parameters Θ^* are:

$$\Theta^* = \arg\min_{\Theta} \sum_{t+1}^{t+1+T'} Div(P_{\mathbf{x}^i}, P'_{\mathbf{x}^i}). \tag{4}$$

Analogous to the autoencoder, we minimize the MSE loss between $\mathbf{x}^i$ and $\mathcal{F}_\Theta(\mathbf{x}^i)$ in practice:

$$\Theta^* = \arg\min_{\Theta} \sum_{t+1}^{t+1+T'} \|\mathbf{x}^i - \mathcal{F}_\Theta(\mathbf{x}^i)\|^2. \tag{5}$$

4.2 Overview

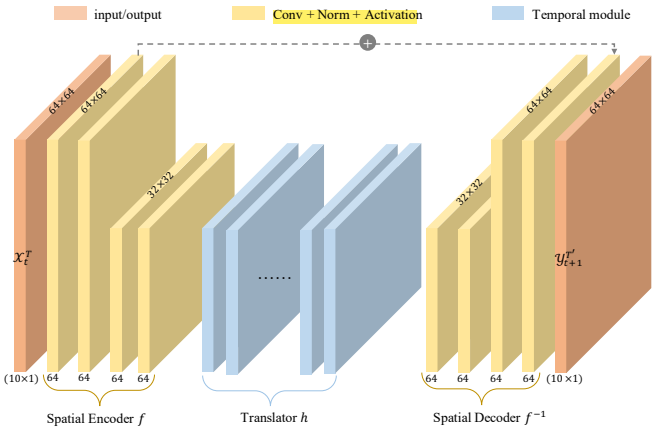


Fig. 4. The overall framework of SimVP.

Taking input Moving MNIST data as an example, we present the overview of our SimVP model as shown in Fig. 4. The spatial encoder is employed to encode the high-dimensional past frames into the low-dimensional latent space, and the translator learns both spatial dependencies and temporal variations from the latent space. The spatial decoder ultimately decodes the latent space into the predicted future frames.

Striving for simplicity, We implement the spatial encoder with N_s vanilla convolutional layers ('Conv2d' in PyTorch) and the spatial decoder with N_s upsampling layers ('ConvTranspose2d' or 'PixelShuffle' in PyTorch). The hidden representations in the spatial encoder f can be formalized as follows:

$$z_i = \sigma(\text{Norm2d}(\text{Conv2d}(z_{i-1}))), 1 \leq i \leq N_s, \tag{6}$$

where σ is a nonlinear activation, Norm2d is a normalization layer, z_0 is the input tensor. The strides of the convolutional layers are one, except downsampling, which has a stride of two. For every two convolutional layers, we perform downsampling once. The hidden representations in the spatial decoder f^{-1} can be formally described as:

$$\begin{aligned} z_k = \sigma(\text{Norm2d}(\text{unConv2d}(z_{i-1}))), \\ N_s + N_t < k \leq 2N_s + N_t, \end{aligned} \tag{7}$$

where unConv2d is a transposed convolutional layer or pixelshuffle layer if it needs upsampling. Otherwise, it is a convolutional layer with stride one.

The middle spatiotemporal translator of the model consists of N_t temporal modules, which we illustrate in detail in Section 4.3. The hidden representations in this part are:

$$z_j = \text{TemporalModule}(z_{i-1}), N_s < j \leq N_s + N_t, \tag{8}$$

where z_{N_s-1} is the output of the spatial encoder. A residual connection from the first layer in the spatial encoder to the last layer in the spatial decoder is introduced to preserve the spatial feature. The mapping $\mathcal{F}_\Theta$ is the composition of the above components:

$$\mathcal{F}_\Theta = f^{-1} \circ h \circ f \tag{9}$$

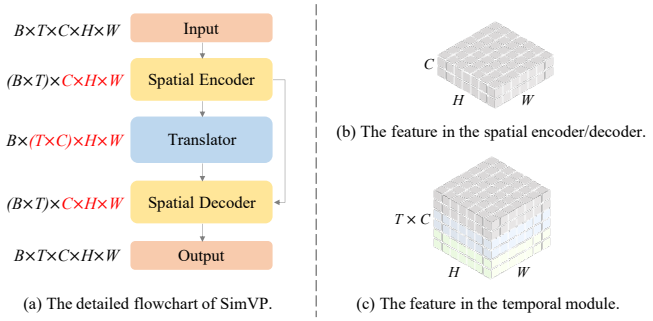
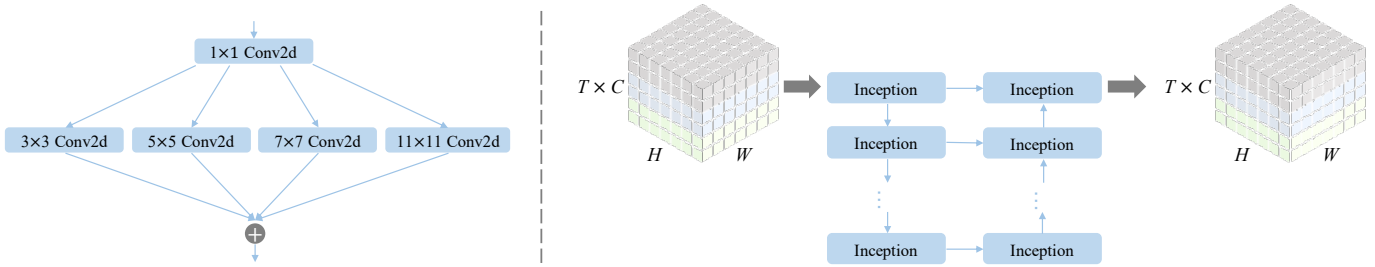


Fig. 5. The spatial encoder and decoder perform single-frame level spatial feature extraction and reconstruction. The translator learns from multi-frame level temporal dependencies.

Given a batch of input past frames $\mathcal{B} \in \mathbb{R}^{B\times T\times C\times H\times W}$ with the batch size of B. In the spatial encoder and decoder, we reshape the input tensors into tensors of shape $(B \times T) \times C \times H \times W$, as shown in Fig. 5 (b). Thus, the spatial encoder and decoder treat each frame as a single sample and focus on the single-frame level features regardless of the temporal variations. In the translator, we reshape the hidden representations from the spatial encoder into tensors of shape $B \times (T \times C) \times H \times W$ and stack multi-frame level features along the time axis, as shown in Fig. 5 (c). By forcing the designed temporal module built upon convolutional networks to learn from stacks of multi-frame features, our model can capture the intrinsic temporal evolutions inside the sequential data.



(a) Inception temporal module (b) Inception-Unet translator

Fig. 6. The proposed Inception temporal module and corresponding Inception-Unet translator architecture.

4.3 Spatiotemporal Translator

The spatiotemporal translator takes the encoded hidden representations of the spatial encoder f as input and outputs hidden spatiotemporal representations for the spatial decoder f^{-1} to decode. Here, we introduce two kinds of spatiotemporal translators built upon pure convolutional neural networks.

4.3.1 Inception-Unet Translator

In the conference version of SimVP, we design an Inception-like temporal module and build the middle spatiotemporal translator with blocks of this module.

As shown in Fig. 6 (a), our Inception temporal module is different from the original Inception module [79] in the following aspects: (1) We apply 1×1 convolution at the front instead of at the end for increasing the hidden dimension in advance. This operation is not responsible for better performance but convenience. (2) We employ larger kernels (e.g., 7×7 and 11×11) than the vanilla Inception module. Larger kernels are preferred for globally distributed information, while smaller kernels are preferred for locally distributed information. Spatiotemporal predictive learning usually faces the difficulty of considerable variations in the location of the valuable information along with time. By leveraging such a multi-branch architecture, the Inception temporal module can jointly obtain both local and global features from stacks of temporal dynamics. (3) The output features from convolutional layers with different kernel sizes are added up instead of concatenated as the simplicity of keeping the same dimension. Our Inception temporal module can be formally described as:

$$\hat{z}^j = \mathrm{Conv2d}_{1\times1}(z^j), \tag{10}$$

$$z^{j+1} = \sum_{k\in\{3,5,7,11\}} \mathrm{Conv2d}_{k\times k}(\hat{z}^j), \tag{11}$$

The middle spatiotemporal translator is built based on the above Inception modules with an Unet-like architecture. The input hidden representations are firstly passed through several Inception temporal modules from top to bottom and then go through a symmetric path from bottom to top. We have concatenation connections between the top-down and bottom-up paths for every Inception temporal module. Note that there is no contracting in the top-down path and expanding in the bottom-up path for simplicity, which is different from the vanilla Unet [80].

4.3.2 Gated Spatiotemporal Attention Translator

In this journal version of SimVP, we propose a gated spatiotemporal attention module and build the middle spatiotemporal translator by stacking such modules instead of using Unet architecture, which further simplifies the model in both time and space complexity. Though this module is still built on pure convolutional networks, it is efficient in capturing spatiotemporal dependencies.

Attention mechanism, which is a hotspot in visual transformers, can adaptively select discriminative features and ignore noisy responses according to the input features. We aim to design a spatiotemporal attention module that automatically captures features relying on temporal dependencies and spatial correlations. Recent research has revealed that large kernel convolutions share advantages with vision transformers in obtaining large effective receptive fields and higher shape bias rather than texture bias [81], [82], [83], [84]. Motivated by this observation, we leverage large kernel convolutions to imitate the attention mechanism and extract spatiotemporal attention from the input representations in the latent space.

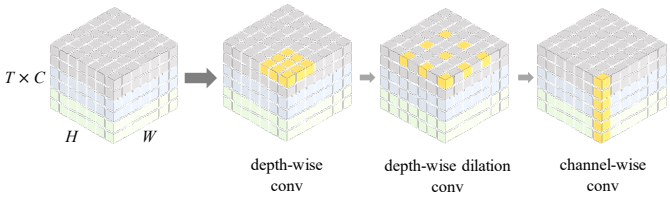
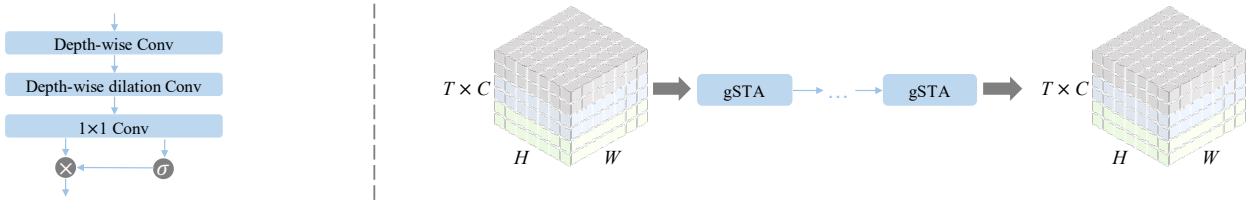


Fig. 7. The large kernel convolution in the gated spatiotemporal attention module. The yellow region denotes the receptive field.

However, directly utilizing large kernel convolutions suffers from inefficient computation and a huge amount of parameters. As an alternative, we decompose the large kernel convolution [81], [82], [84] into several components: (1) a depth-wise convolution that captures local receptive fields within a single channel, (2) a depth-wise dilation convolution that builds connections between distant receptive fields, (3) a 1×1 convolution that performs channel-wise interactions. A $(2d-1) \times (2d-1)$ depth-wise convolution and a $\frac{K}{d} \times \frac{K}{d}$ depth-wise dilation convolution with dilation d have a receptive field with a size of $K \times K$, and a channel-wise 1×1 further assist them in multi-channel connections. We use the above three components to simulate the large kernel convolution with a low computational overhead, as shown in Fig. 7.



(a) gated spatiotemporal attention (gSTA) module

(b) gSTA translator

Fig. 8. The proposed gSTA module and corresponding gSTA translator architecture.

The gated spatiotemporal attention (gSTA) module is illustrated in Fig. 8 (a). Benefited by the large receptive fields, we can capture long-range correlations in both spatial and temporal perspectives. We split the output of the above large kernel convolution operation into two parts and take one of them with a sigmoid function as an attention gate. The gated spatiotemporal attention module is formalized as:

$$\hat{z}^j = \text{Conv2d}_{1\times1}(\text{Conv}_{\text{Dw-d}}(\text{Conv}_{\text{Dw}}(z^j))), \quad (12)$$

$$g, \bar{z}^j = \text{split}(\hat{z}^j), \quad (13)$$

$$z^{j+1} = \sigma(g) \odot \bar{z}^j, \quad (14)$$

where Conv_{Dw} is the depth-wise convolution and $\text{Conv}_{\text{Dw-d}}$ is the depth-wise dilation convolution, g is the spatiotemporal attention coefficients, and $\odot$ denotes element-wise multiplication.

We show the gSTA translator in Fig. 8 (b). With the gSTA module in place, we can build the middle translator by simply stacking several gSTA modules without Unet architecture. The spatiotemporal attention coefficient g provides a dynamic mechanism that adaptively changes according to the input features. And the gated attention $\sigma(g)$ gate is used to adaptively select the informative features and filter unimportant features from a spatiotemporal perspective.

5 Experiments

In this section, we present detailed experimental results. The experiments are conducted on various datasets with different settings to evaluate our proposed method from the following aspects:

- Standard spatiotemporal predictive learning (Section 5.1). We regard the video prediction problem with 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 [4], TaxiBJ [85], and WeatherBench [86] datasets**.
- Generalization ability across different datasets (Section 5.2). Generalizing the learned knowledge to other domains is a challenge in unsupervised learning. We investigate such ability of our method by training the model on the **KITTI [87] dataset** and evaluating it on the **Caltech Pedestrian [88] dataset.**
- Predicting frames with flexible lengths (Section 5.3). One of the advantages of recurrent units is that they can easily handle flexible-length frames like the **KTH dataset** [89]. 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.

We summarize the statistics of the above datasets in Table 2, including the number of training samples N_{train} and the number of testing samples N_{test}.

TABLE 2
The statistics of datasets. The training or testing set has N_{train} or N_{test} samples, composed by T or T' images with the shape (C, H, W).

	N_{train}	N_{test}	(C, H, W)	T	T'
MMNIST	10,000	10,000	(1, 64, 64)	10	10
TaxiBJ	19,627	1,334	(2, 32, 32)	4	4
WeatherBench	324,311	17,495	(1, 32, 64)	12	12
Caltech	2042	1983	(3, 128, 160)	10	1
KTH	5200	3167	(1, 128, 128)	10	20 or 40

5.1 Standard spatiotemporal predictive learning

5.1.1 *Moving MNIST*

We first evaluate our model on the Moving MNIST [4] dataset and compare the results with state-of-the-art methods. The moving MNIST dataset is one of the fundamental benchmark datasets in spatiotemporal predictive learning and is widely used in the literature. In this dataset, each video is generated with 20 frames long and consists of two digits inside a 64×64 patch. The digits are randomly selected from the training set and placed initially at random locations. Each digit is assigned a velocity whose direction is chosen uniformly at random on the unit circle and whose size is chosen uniformly at random within a fixed range. The digits bounce off the edges of the 64×64 frame and overlap if they are in the same position.

As almost every spatiotemporal predictive learning methods evaluate them on this dataset and corresponding open-source codes, we reproduce state-of-the-art methods into a unified framework and evaluate them with the same protocol for fair comparisons. Those strong baselines include competitive recurrent-based models ConvLSTM [1], PredRNN [6], PredRNN++ [7], MIM [3], E3D-LSTM [5], PhyDNet [47], CrevNet [14], and MAU [90]. By using ConvLSTM with different sizes as the standard baselines, we divide these models into two groups according to their computational cost, i.e., small and large model groups, as

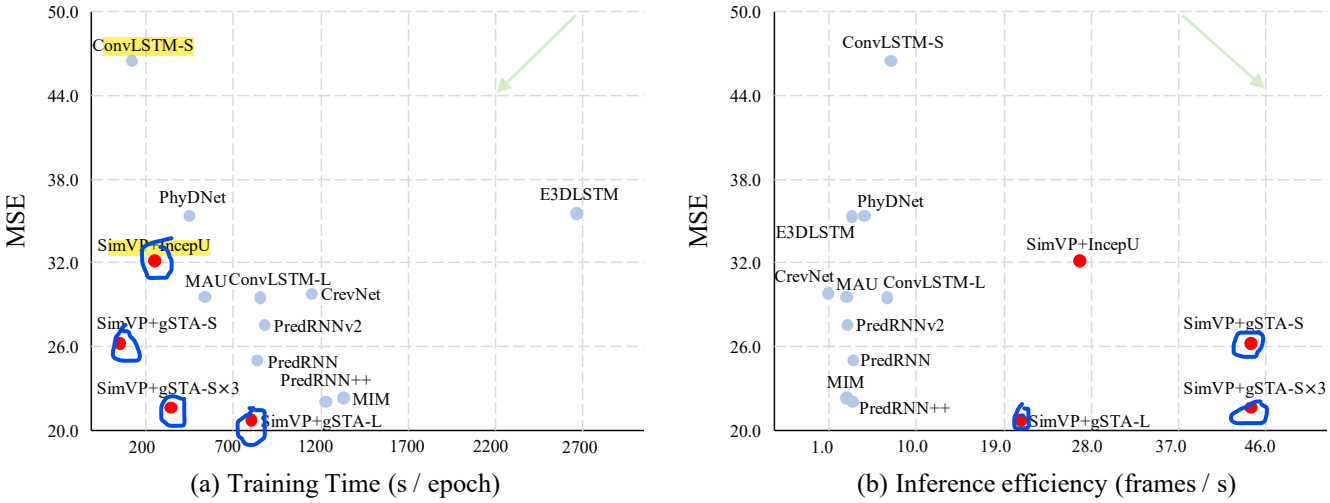


Fig. 9. The performance of SimVPs on the Moving MNIST dataset. The variants of SimVP are denoted in red color. For the training time, the less the better. For the inference efficiency (frames per second), the more the better. The light green arrow indicates the direction of model optimization.

TABLE 3
Quantitative results of different methods on the Moving MNIST dataset ($10 \rightarrow 10$ frames). Note that "-S" denotes the smaller model and "-L" denotes the larger model. We report "SimVP+gSTA-S×3" by training "SimVP+gSTA-s" with three times as much epoch, i.e., 600 epochs. Its training efficiency is reported by multiplying the original time by three.

Method	Flops (G) ↓	Training time ≈ (s) ↓	Inference efficiency ↑	MSE ↓	MAE ↓	SSIM ↑
ConvLSTM-S	**14.45**	190	7.50	46.26±0.26	142.18±0.61	0.878±0.001
PhyDNet	15.33	452	4.62	35.68±0.40	96.70±0.29	0.917±0.000
MAU	17.79	535	3.08	30.64±0.10	88.17±0.35	0.928±0.001
SimVP+IncepU	19.43	261	27.15	32.22±0.02	89.19±0.33	0.927±0.000
SimVP+gSTA-S	16.53	**156**	**44.09**	**26.60**±0.02	**77.32**±0.22	**0.940**±0.000
ConvLSTM-L	127.01	879	6.24	29.88±0.17	95.05±0.25	0.925±0.000
PredRNN	115.95	869	3.97	25.04±0.08	76.26±0.29	0.944±0.000
PredRNN++	171.73	1280	3.71	22.45±0.36	69.70±0.25	0.950±0.000
MIM	179.18	1388	3.08	23.66±0.20	74.37±0.46	0.946±0.000
E3D-LSTM	298.87	2693	3.73	36.19±0.20	78.64±0.35	0.932±0.000
CrevNet	270.68	1166	1.01	30.15±1.61	86.28±2.65	0.935±0.003
PredRNNv2	116.59	899	3.49	27.73±0.08	82.17±0.33	0.937±0.000
SimVP+gSTA-S×10	16.53	1560	**44.09**	**15.05**±0.03	**49.80**±0.10	**0.967**±0.000
SimVP+gSTA-S×5	16.53	780	**44.09**	16.47±0.02	53.24±0.04	0.964±0.000
SimVP+gSTA-S×3	16.53	468	**44.09**	22.37±0.06	67.52±0.03	0.951±0.000
SimVP+gSTA-L	152.20	796	21.23	21.81±0.03	66.43±0.04	0.952±0.000

shown in Table 3. In particular, ConvLSTM-S is the small model with four ConvLSTM layers with hidden size 64, and ConvLSTM-L is the large model with four ConvLSTM layers with hidden size 192.

We train the models using the Adam optimizer [91] with the OneCycle learning rate scheduler [92]. Following [55], We choose the optimal learning rate from $\{1e^{-2}, 1e^{-3}, 1e^{-4}\}$ under the premise of stable training. The batch size is set to 16 for all the models but 4 for E3D-LSTM for its large memory cost. We train the models for 200 epochs and evaluate the performance by mean square error (MSE), mean absolute error (MAE), and structural similarity index (SSIM) [93]. We repeat each experiment for three trials and provide the average results. Besides, we report Flops for every sample using fvcore [94] for accurate computation. The training time is reported by computing the average seconds for training an epoch. The inference efficiency is also reported by inferencing 10,000 test samples with a batch size of 1 and computing the average testing frames per second (FPS). Both the training time and the inference efficiency are tested on a single NVIDIA Tesla V100 GPU.

Table 3 shows the performance of SimVPs on the Moving MNIST dataset. Surprisingly, despite the simple architecture without recurrent units, SimVPs can still achieve highly competitive performance compared to state-of-the-art methods based on recurrent units. For the small model group (the first five rows in Table 3), SimVP+gSTA-S obtains the best prediction quality with the fastest training time and the highest inference efficiency. SimVP+IncepU also achieves competitive performance that is only weaker than SimVP+gSTA-S and MAU but has about nine times higher inference efficiency than MAU. For the large model group (from the sixth to the last row in Table 3), SimVP+gSTA-L, which uses larger hidden dimensions and a larger number of layers, achieves the best prediction quality compared with other oversized models. Furthermore, we report SimVP+gSTA-S×3 that simply trains SimVP+gSTA-S with three times epochs, i.e., 600 epochs. Surprisingly, SimVP+gSTA-S×3 achieves competitive performance as well as SimVP+gSTA-L. Though trained with three times epochs, SimVP+gSTA-S×3 still has the least training time compared with large models.

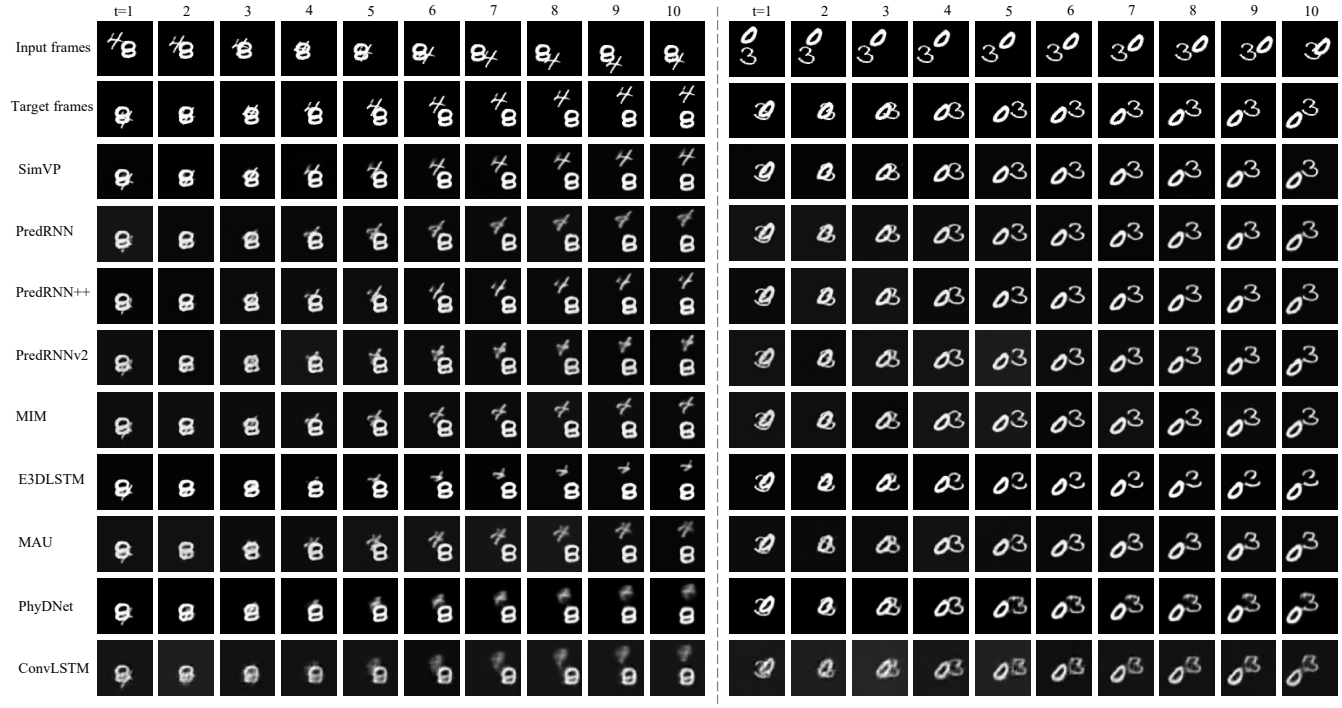


Fig. 10. Examples of predicted results on the Moving MNIST dataset. We denote SimVP+gSTA as SimVP for convenience here.

We plot the performance vs. training time and the performance vs. inference efficiency in Fig. 9(a) and Fig. 9(b), respectively. We choose MSE as the performance metric in these plots because it is synchronized with other metrics and thus enough to evaluate the Moving MNIST dataset. In Fig. 9(a), both the training time and MSE metric are the lower the better. We can see that The variants of SimVP are concentrated in the lower-left corner of this plot. SimVP+gSTA-S even takes about only one-sixteenth training time of E3D-LSTM and obtains significantly better performance. In Fig. 9(b), the inference efficiency is the higher the better. Thus SimVPs are concentrated in the lower-right corner. SimVPs significantly outperform other recurrent-based models and are the only model that can achieve more than 10 FPS. SimVP+gSTA-S has about six times inference efficiency compared to ConvLSTM-S and about forty times compared to CrevNet. Based on the above observations, we demonstrate that our proposed SimVPs outperform state-of-the-art methods in both training and inference efficiency.

Fig. 10 shows the qualitative comparison between SimVP and other state-of-the-art methods. It can be seen that SimVP predicts much clearer frames, especially when it comes to long-range predictions. For the first example, only SimVP shows clear and sharp digit '4' while other methods do not. When digit '4' and digit '8' are overlapped at $t = 4$, we still can infer these digits from the predicted frame of SimVP, but other methods fail to reconstruct the original digit '4'. PhyDNet and ConvLSTM even produce severely blurry frames from the beginning to the end. For the second example, most methods perform well except PhyDNet and ConvLSTM. PredRNN and its variants predict high-quality frames, but their predicted digits have some distortions, while SimVP keeps predicting almost the same frames as the ground-truth frames.

5.1.2 *TaxiBJ*

Traffic flow forecasting is of great importance to traffic management and public safety while being influenced by a variety of complex factors and is very challenging. We recognize traffic flow forecasting as a fundamental problem in standard spatiotemporal predictive learning. Due to the complex dependencies on road networks and non-linear temporal dynamics, previous traffic forecasting methods suffer from low prediction quality.

We use the TaxiBJ dataset [85] to evaluate the traffic forecasting ability of our proposed model. TaxiBJ contains the trajectory data in Beijing collected from taxicab GPS with two channels, i.e., inflow or outflow defined in [85]. Following [3], we transform the data into $[0, 1]$ via max-min normalization. Since the original data is between -1 and 1, the reported MSE and MAE are 1/4 and 1/2 of the original ones, consistent with previous literature [3], [47]. Models are trained to predict 4 subsequent frames by observing the prior 4 frames. We compare SimVP with ConvLSTM, PredRNN, PredRNN+, MIM, E3D-LSTM, and PhyDNet.

TABLE 4
Quantitative results of different methods on the TaxiBJ dataset ($4 \rightarrow 4$ frames).

Method	TaxiBJ MSE × 100↓	TaxiBJ MAE↓	TaxiBJ SSIM↑
ConvLSTM	48.5	17.7	0.978
PredRNN	46.4	17.1	0.971
PredRNN++	44.8	16.9	0.977
MIM	42.9	16.6	0.971
E3D-LSTM	43.2	16.9	0.979
PhyDNet	41.9	16.2	0.982
SimVP+IncepU	41.4	16.2	0.982
SimVP+gSTA	**34.8**	**15.6**	**0.984**

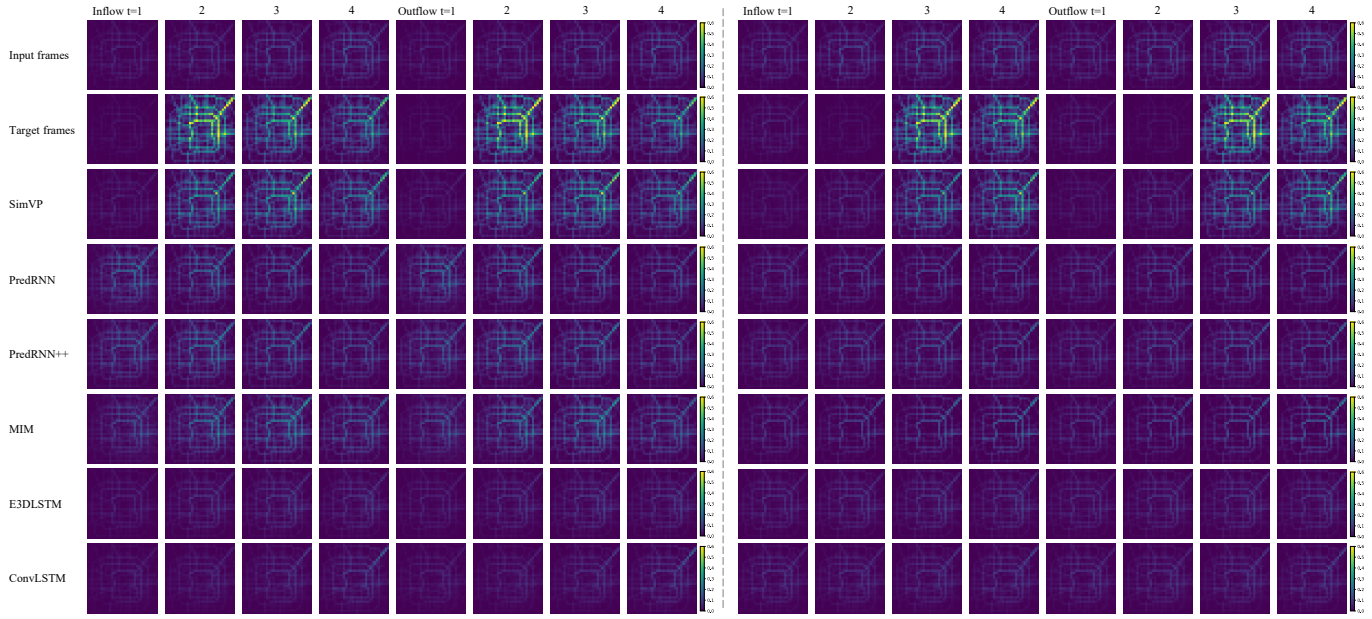


Fig. 11. Examples of predicted results on the TaxiBJ dataset. We denote SimVP+gSTA as SimVP for convenience here.

As shown in Table 4, SimVP+IncepU outperforms the previous recurrent-based state-of-the-art methods by a small margin among metrics like MSE, MAE, and SSIM. SimVP+gSTA that is introduced in this paper further stretches the margin between baseline models, quantitatively improving SimVP+IncepU by about 15.94% in the MSE metric and about 3.7% in the MAE metric. Benefiting from the gated spatiotemporal attention mechanism, SimVP+gSTA is able to achieve superior performance on such a complex traffic flow forecasting problem.

We also visualize two examples of predicted results on the TaxiBJ dataset. These two examples are exceptional cases that have very different target frames comparing to input frames. The first example has a sudden increase in traffic flow under both inflow and outflow channels from $t = 2$ in the target frames. The second example also performs a similar trend as the first example, but from $t = 3$ in the target frames. Such a sudden change of traffic flow may occur in real-life transportation, e.g., the traffic jam in the early morning hours after a period of idleness. It is extremely tough for deep learning models to detect sudden future traffic busy from this idle state. To prevent the model benefited from copying input frames, we elaborately select these two complex examples to qualitatively validate the real prediction ability of spatiotemporal predictive methods.

The predicted results of these two complex examples are impressive. While other recurrent-based methods fail to capture such different traffic variations, SimVP accurately predicts the future trend to a large extent and unexpectedly finds the sudden traffic jam from the observations of placid transportation. This phenomenon reveals the powerful perception of the long-range future of our proposed SimVP model. SimVP learns the spatiotemporal dynamics in a way that is consistent with the real-world situation. In contrast, recurrent-based methods seem to be over-depending on the previous frames, and they are not able to directly capture the long-range dependencies in such complex traffic flows.

5.1.3 *WeatherBench*

Climate prediction is another fundamental task in spatiotemporal predictive learning. Current purely physical computational methods that solve the governing equations on a discrete numerical grid suffer from large amounts of computing power, especially for creating probabilistic forecasts with ensemble members. Thus, a robust data-driven spatiotemporal predictive learning model has been urged in this field for a considerable period. However, capturing global weather patterns from climate data over the years remains challenging.

In this context, we employ our SimVP model in the climate prediction problem on the WeatherBench [86] dataset. This dataset contains various types of climatic data from 1979 to 2018. The raw data is regrid to low resolutions, we here choose 5.625° (32×64 grid points) resolution for our data. Since the complete data is very large that includes massive climatic attributes like geopotential, temperature, and other variables, we specifically choose the temperature prediction task to evaluate our model. Following the original protocol from [86], we train the model using data from 1979 to 2015 and validate the model using data from 2016. The final evaluation is done for the years 2017 and 2018. We use the global temperature from the past 12 hours to predict that in the future 12 hours. The unit of global temperature is K. The results are evaluated by RMSE and MAE metrics.

We compare our model with other strong climate prediction baselines, i.e., TGCN [95], STGCN [96], MSTGCN [97], ASTGCN [97], GCGRU [98], DCRNN [99], AGCRN [100], CLCSTN [101], and CLCRN [101]. Additional comparisons with spatiotemporal predictive learning methods such as ConvLSTM [1], PredRNN [6], and PredRNN++ [7] are included. While some of them are designed especially for this task, we do not employ any tricks for SimVP. We train the SimVP model for 50 epochs and optimize the training using Adam optimizer with learning 0.01. The MSE loss function is consistent with the previous tasks.

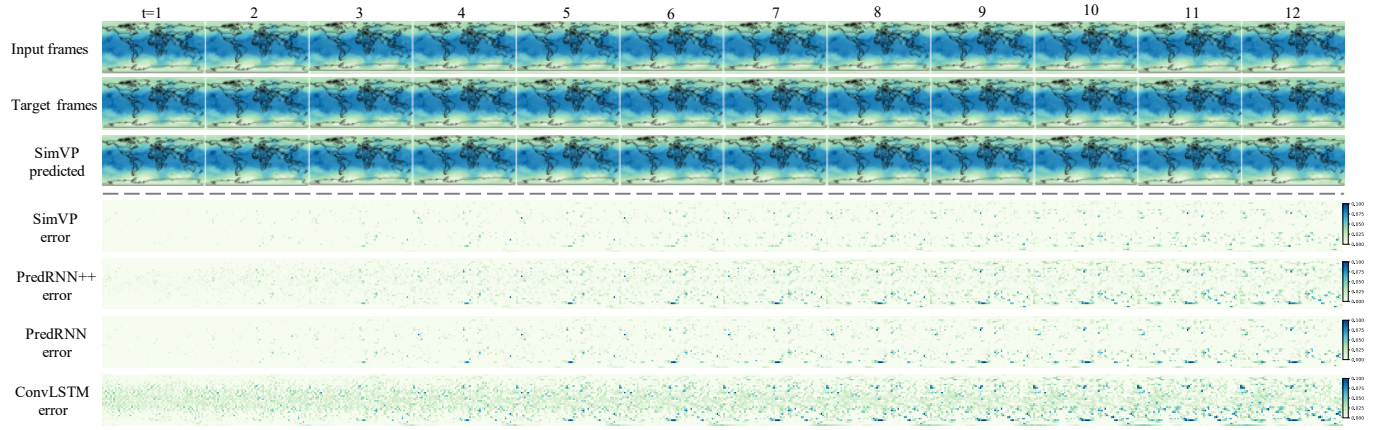


Fig. 12. Examples of predicted results on the WeatherBench dataset. We denote SimVP+gSTA as SimVP for convenience here.

We report the quantitative results in Table 5. As the input and target frames are similar, we also report the results by copying the input frames as the predicted frames to evaluate the actual results, which is denoted as 'Copying' in Table 5. Unexpectedly, some climate prediction methods like TGCN and STGCN fail to predict reasonable results and even perform worse than the Copying baseline. It can be seen that SimVP outperforms current state-of-the-art climate prediction models by relatively large margins. Specifically, SimVP improves the state-of-the-art meteorological forecasting model CLCRN by about 36.04% in the MAE metric and about 42.71% in the RMSE metric. Moreover, other common spatiotemporal predictive learning methods also show superior performance on this global temperature prediction task, while SimVP still holds the lead.

TABLE 5
Quantitative results of different methods on the WeatherBench dataset ($12 \rightarrow 12$ frames).

Method	WeatherBench	
	MAE↓	RMSE↓
Copying	1.6906	2.4838
TGCN	3.8638	5.8554
STGCN	4.3525	6.8600
MSTCN	1.2199	1.9203
ASTGCN	1.4896	2.4622
GCGRU	1.3256	2.1721
DCRNN	1.3232	2.1874
AGCRN	1.2551	1.9314
CLCSTN	1.3325	2.1239
CLCRN	1.1688	1.8825
ConvLSTM	1.0529	1.4606
PredRNN	0.8268	1.2119
PredRNN++	0.8054	1.1776
SimVP+gSTA	**0.7475**	**1.0785**

The qualitative results are visualized in Fig. 12. For convenience, we normalize the data into $[0, 1]$ to obtain a decent visualization effect. It can be seen that the predicted frames of SimVP are highly similar to the ground-truth frames. We also plot the error comparison in the last four rows in Fig. 12. The error is calculated as the absolute values of the differences between the input and predicted frames, i.e., $|predicted - target|$. SimVP has the most sparse error plot in almost every predicted frame. PredRNN++ and PredRNN perform well at the beginning but fail to accurately predict the global temperature. ConvLSTM, however, obtains the messy error plot from the beginning and predicts worse and worse along the time axis. Based on the above observations, We demonstrate the strong ability of SimVP to capture the weather patterns in the global weather prediction task.

5.2 Generalization ability across different datasets

Generalizing the knowledge across different datasets, especially in an unsupervised setting, is the core research point of machine learning and artificial intelligence. To investigate the generalization ability of SimVP, we train the model for 50 epochs on KITTI and evaluate it on Caltech Pedestrian. Both KITTI and Caltech datasets are captured from road traffic scenarios but in different environments. It is reasonable to assume that generalizing the knowledge from a specific road traffic scenario to another is possible for robust spatiotemporal predictive learning.

KITTI [87] is one of the most popular datasets for mobile robotics and autonomous driving. It includes hours of traffic scenarios recorded with high-resolution RGB images. CalTech Pedestrian [102] is a driving dataset focused on detecting pedestrians. It is conformed of approximately 10 hours of 640×480 30 FPS video taken from a vehicle driving through regular traffic in an urban environment. Models are trained on the KITTI dataset to predict the next frame after a 10-frame warm-up and are evaluated on Caltech Pedestrian. We aim to force our SimVP to predict the next frame by previously observed 10 frames.

Compared with the previous experiments, the car-mounted camera videos dataset and the distinct training-evaluating data present another level of difficulty for spatiotemporal predictive learning as it describes various nonlinear three-dimensional dynamics of multiple moving objects including backgrounds. Such an experimental setting requires the robust generalization ability of spatiotemporal predictive learning models. Following [14], [103], [104], several strong baselines are selected for comparison, including BeyondMSE [15], MCnet [29], DVF [34], DualGAN [31], CtrlGen [105], PredNet [103], ContextVP [52], SDC-Net [106], rCycleGan [44], DPG [43], CrevNet [14] and STMFANet [107]. SSIM [93], PSNR, and LPIPS [108] metrics are used in the evaluation phase.

As shown in Table 6, SimVP+IncepU has achieved better performance than the baseline models by a large margin. Specifically, SimVP+IncepU outperforms STMFANet by about 1.04% in the SSIM metric, approximately 13.74% in the PSNR metric, and approximately 35.31% in the LPIPS metric. SimVP+gSTA further improves the SimVP+IncepU and obtains the best performance among SSIM, PSNR, and LPIPS metrics.

TABLE 6
Quantitative results of different methods on the Caltech Pedestrian dataset ($10 \rightarrow 1$ frame).

Method	Caltech Pedestrian		
	SSIM↑	PSNR↑	LPIPS↓
BeyondMSE	0.847	-	-
MCnet	0.879	-	-
DVF	0.897	26.2	5.57
Dual-GAN	0.899	-	-
CtrlGen	0.900	26.5	6.38
PredNet	0.905	27.6	7.47
ContextVP	0.921	28.7	6.03
SDC-Net	0.918	-	-
rCycleGan	0.919	29.2	-
DPG	0.923	28.2	5.04
CrevNet	0.925	29.3	-
STMFANet	0.927	29.1	5.89
SimVP+IncepU	0.940	33.1	3.81
SimVP+gSTA	**0.949**	**33.2**	**3.11**

We also visualize several examples of predicted results in Fig. 13. It can be seen that SimVP accurately predicts the directions of lane lines in the first three examples based on the previously observed frames. SimVP shows its strong prediction ability though there are obvious differences between the last frame and the predicted frame. From the last column, we can see that the error is relatively large when the scene changes dramatically but remains low in most cases. Based on the above observations, we demonstrate the robust generalization ability of SimVP.

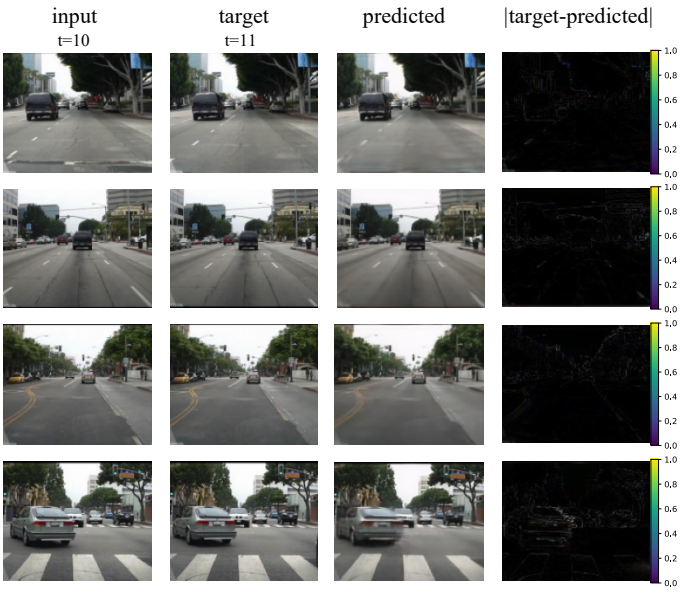


Fig. 13. Examples of predicted results on the Caltech dataset. We denote SimVP+gSTA as SimVP for convenience here.

5.3 Predicting frames with flexible lengths

A possible limitation of CNN-based methods is that it may be challenging to scale to prediction with flexible length. Though we can easily change the length of output predicted frames by expanding the temporal channels, we prefer an alternative solution without involving more parameters and calculations. Thus, we handle this flexible length problem by imitating RNN, which takes previous predictions as current inputs to recursively produce long-term predictions. We choose the KTH dataset [89] for evaluating the flexibility of SimVP. It contains 25 individuals performing 6 types of actions, i.e., walking, jogging, running, boxing, hand waving, and hand clapping. The complication of human motion lies in the randomness of different people performing different actions. However, the motion contained in the KTH dataset is relatively regular. By observing the previous frames, the spatiotemporal predictive learning model is supposed to learn the dynamics of human motion and is able to predict posture changes in the long-term future.

Following [5], [104], we compare the PSNR and SSIM of SimVP with other baselines on KTH. We train our model for 100 epochs and evaluate the results by SSIM and PSNR metrics. Following [5], [29], we use persons 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. Sixteen strong baselines are included, such as MCnet [29], ConvLSTM [1], SAVP, SAVP-VAE [109], VPN [12], DFN [110], fRNN [38], Znet [36], SV2P [2], PredRNN [6], VarNet [111], PredRNN++ [7], MSNET [112], E3d-LSTM [5], and STMFANet [107]. We compare the predicted quality with these recent state-of-the-art baselines under both $10 \rightarrow 20$ frames and $10 \rightarrow 40$ frames cases.

We show the quantitative results on the KTH dataset with output frame lengths of 20 and 40 in Table 7. It can be seen that SimVPs are superior to those baselines in both PSNR and SSIM metrics. Moreover, SimVPs even accurately predict the future frames under the extremely long-range case like $10 \rightarrow 40$ frames. We demonstrate that SimVP is able to predict future frames with flexible length.

TABLE 7
Quantitative results of different methods on the KTH dataset ($10 \rightarrow 20$ and $10 \rightarrow 40$ frames).

Method	KTH ($10 \rightarrow 20$)		KTH ($10 \rightarrow 40$)	
	SSIM↑	PSNR↑	SSIM↑	PSNR↑
MCnet	0.804	25.95	0.73	23.89
ConvLSTM	0.712	23.58	0.639	22.85
SAVP	0.746	25.38	0.701	23.97
VPN	0.746	23.76	–	–
DFN	0.794	27.26	0.652	23.01
fRNN	0.771	26.12	0.678	23.77
Znet	0.817	27.58	–	–
SV2Pv	0.838	27.79	0.789	26.12
PredRNN	0.839	27.55	0.703	24.16
VarNet	0.843	28.48	0.739	25.37
SAVP-VAE	0.852	27.77	0.811	26.18
PredRNN++	0.865	28.47	0.741	25.21
MSNET	0.876	27.08	–	–
E3d-LSTM	0.879	29.31	0.810	27.24
STMFANet	0.893	29.85	0.851	27.56
SimVP+IncepU	0.905	33.72	0.886	32.93
SimVP+gSTA	**0.913**	**34.24**	**0.895**	**33.35**

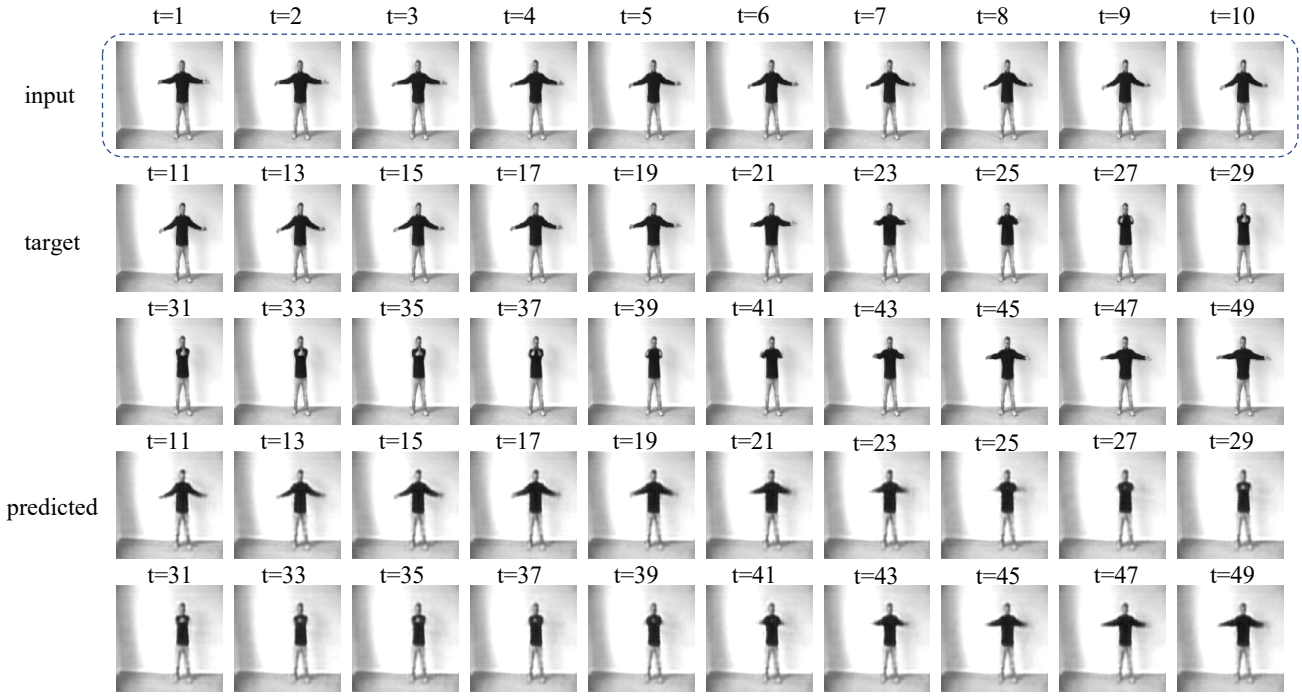


Fig. 14. An example of predicted results on the KTH dataset.

We show the qualitative results on the KTH dataset with output frame lengths of 40 in Fig. 14. In general, SimVP can predict the overall posture at almost every frame, albeit with a slight blur. Although the static contour is not very detailed, the movements of the human body are learned based on observations and related motions from training data, thus enabling the accurate prediction of future postures.

5.4 Ablation study

To explore the roles of spatiotemporal translator, spatial encoder, and decoder, we perform an ablation study on the Moving MNIST dataset, as shown in Fig. 15.

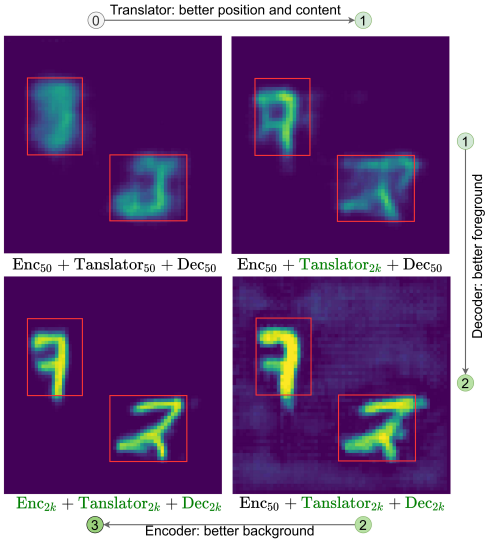


Fig. 15. The role of the Translator, Spatial Encoder and Decoder.

We represent submodules trained with n epochs as $\mathrm{Enc}_n, \mathrm{Translator}_n, \mathrm{Dec}_n$. Given a model trained with 50 epochs, we replace its submodules with maturer ones trained with $2,000$ epochs. It can be seen that the spatiotemporal translator mainly focuses on predicting the position and content of the objects. The spatial encoder focuses on the background portrayed, and the spatial decoder is responsible for optimizing the shape of the foreground objects.

6 Conclusion

In this paper, we propose SimVP for the simple yet powerful spatiotemporal predictive learning. Without recurrent architectures, SimVP challenges the common sense that pure convolution networks are weak in capturing spatiotemporal correlations. Our model is composed of a spatial encoder, a spatiotemporal translator, and a spatial decoder. The spatial encoder encodes the observed frames into the latent space, the spatiotemporal translator learns both spatial and temporal variations from the latent space, and the spatial decoder further reconstructs the predicted future frames. Through extensive experiments on the synthetic moving digits, traffic flow forecasting, climate prediction, road driving, and human motion prediction, we demonstrate the superior performance of SimVP under various settings like standard spatiotemporal predictive learning, generalization across similar scenarios, and prediction with flexible lengths. We believe SimVP can serve as a strong baseline and provide a new perspective for future research.

Acknowledgments

This work is supported by the Science and Technology Innovation 2030 - Major Project (No. 2021ZD0150100) and National Natural Science Foundation of China (No. U21A20427).

REFERENCES

[1] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[2] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, "Stochastic variational video prediction," *arXiv preprint arXiv:1710.11252*, 2017.

[3] Y. Wang, J. Zhang, H. Zhu, M. Long, J. Wang, and P. 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*, 2019, pp. 9154–9162.

[4] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International conference on machine learning*. PMLR, 2015, pp. 843–852.

[5] Y. Wang, L. Jiang, M.-H. Yang, L.-J. Li, M. Long, and L. Fei-Fei, "Eidetic 3d lstm: A model for video prediction and beyond," in *International conference on learning representations*, 2018.

[6] Y. Wang, M. Long, J. Wang, Z. Gao, and P. S. Yu, "Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[7] Y. Wang, Z. Gao, M. Long, J. Wang, and S. Y. Philip, "Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5123–5132.

[8] Y. Wang, H. Wu, J. Zhang, Z. Gao, J. Wang, P. S. Yu, and M. Long, "Predrnn: A recurrent neural network for spatiotemporal predictive learning," *arXiv preprint arXiv:2103.09504*, 2021.

[9] D. Weissenborn, O. Täckström, and J. Uszkoreit, "Scaling autoregressive video models," *arXiv preprint arXiv:1906.02634*, 2019.

[10] R. Rakhimov, D. Volkhonskiy, A. Artemov, D. Zorin, and E. Burnaev, "Latent video transformer," *arXiv preprint arXiv:2006.10704*, 2020.

[11] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra, "Video (language) modeling: a baseline for generative models of natural videos," *arXiv preprint arXiv:1412.6604*, 2014.

[12] N. Kalchbrenner, A. Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu, "Video pixel networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1771–1779.

[13] J. Ho, N. Kalchbrenner, D. Weissenborn, and T. Salimans, "Axial attention in multidimensional transformers," *arXiv preprint arXiv:1912.12180*, 2019.

[14] W. Yu, Y. Lu, S. Easterbrook, and S. Fidler, "Efficient and information-preserving future frame prediction and beyond," in *International Conference on Learning Representations*, 2019.

[15] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," *arXiv preprint arXiv:1511.05440*, 2015.

[16] M. Saito, E. Matsumoto, and S. Saito, "Temporal generative adversarial nets with singular value clipping," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2830–2839.

[17] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1526–1535.

[18] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," *Advances in neural information processing systems*, vol. 29, pp. 613–621, 2016.

[19] M. Saito and S. Saito, "Tganv2: Efficient training of large models for video generation with multiple subsampling layers," 2018.

[20] P. Luc, A. Clark, S. Dieleman, D. d. L. Casas, Y. Doron, A. Cassirer, and K. Simonyan, "Transformation-based adversarial video prediction on large-scale data," *arXiv preprint arXiv:2003.04035*, 2020.

[21] A. Clark, J. Donahue, and K. Simonyan, "Adversarial video generation on complex datasets," *arXiv preprint arXiv:1907.06571*, 2019.

[22] D. Acharya, Z. Huang, D. P. Paudel, and L. Van Gool, "Towards high resolution video generation with progressive growing of sliced wasserstein gans," *arXiv preprint arXiv:1810.02419*, 2018.

[23] V. Michalski, R. Memisevic, and K. Konda, "Modeling deep temporal dependencies with recurrent grammar cells"",," *Advances in neural information processing systems*, vol. 27, pp. 1925–1933, 2014.

[24] V. Patraucean, A. Handa, and R. Cipolla, "Spatio-temporal video autoencoder with differentiable memory," *arXiv preprint arXiv:1511.06309*, 2015.

[25] W. Lotter, G. Kreiman, and D. Cox, "Unsupervised learning of visual structure using predictive generative networks," *arXiv preprint arXiv:1511.06380*, 2015.

[26] C. Lu, M. Hirsch, and B. Scholkopf, "Flexible spatio-temporal networks for video prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6523–6531.

[27] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," in *international conference on machine learning*. PMLR, 2017, pp. 3560–3569.

[28] I. Prémont-Schwarz, A. Ilin, T. H. Hao, A. Rasmus, R. Boney, and H. Valpola, "Recurrent ladder networks," *arXiv preprint arXiv:1707.09219*, 2017.

[29] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing motion and content for natural video sequence prediction," in *International Conference on Learning Representations*, 2017.

[30] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[31] X. Liang, L. Lee, W. Dai, and E. P. Xing, "Dual motion gan for future-flow embedded video prediction," in *proceedings of the IEEE international conference on computer vision*, 2017, pp. 1744–1752.

[32] E. L. Denton *et al.*, "Unsupervised learning of disentangled representations from video," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[33] J. Van Amersfoort, A. Kannan, M. Ranzato, A. Szlam, D. Tran, and S. Chintala, "Transformation-based models of video sequences," *arXiv preprint arXiv:1701.08435*, 2017.

[34] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4463–4471.

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

[36] J. Zhang, Y. Wang, M. Long, W. Jianmin, and S. Y. Philip, "Z-order recurrent neural networks for video prediction," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2019, pp. 230–235.

[37] J. Sun, J. Xie, J.-F. Hu, Z. Lin, J. Lai, W. Zeng, and W.-S. Zheng, "Predicting future instance segmentation with contextual pyramid convlstms," in *Proceedings of the 27th acm international conference on multimedia*, 2019, pp. 2043–2051.

[38] M. Oliu, J. Selva, and S. Escalera, "Folded recurrent neural networks for future video prediction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 716–731.

[39] J.-T. Hsieh, B. Liu, D.-A. Huang, L. Fei-Fei, and J. C. Niebles, "Learning to decompose and disentangle representations for video prediction," *arXiv preprint arXiv:1806.04166*, 2018.

[40] R. Villegas, D. Erhan, H. Lee *et al.*, "Hierarchical long-term video prediction without supervision," in *International Conference on Machine Learning*. PMLR, 2018, pp. 6038–6046.

[41] E. Denton and R. Fergus, "Stochastic video generation with a learned prior," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1174–1183.

[42] L. Castrejon, N. Ballas, and A. Courville, "Improved conditional vrnns for video prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7608–7617.

[43] H. Gao, H. Xu, Q.-Z. Cai, R. Wang, F. Yu, and T. Darrell, "Disentangling propagation and generation for video prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9006–9015.

[44] Y.-H. Kwon and M.-G. Park, "Predicting future frames using retrospective cycle gan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1811–1820.

[45] Z. Xu, Y. Wang, M. Long, J. Wang, and M. KLiss, "Predcnn: Predictive learning with cascade convolutions." in *IJCAI*, 2018, pp. 2940–2947.

[46] A. Hu, F. Cotter, N. Mohan, C. Gurau, and A. Kendall, "Probabilistic future prediction for video scene understanding," in *European Conference on Computer Vision*. Springer, 2020, pp. 767–785.

[47] V. L. Guen and N. Thome, "Disentangling physical dynamics from unknown factors for unsupervised video prediction," in

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11 474–11 484.
[48] M. Babaeizadeh, M. T. Saffar, S. Nair, S. Levine, C. Finn, and D. Erhan, "Fitvid: Overfitting in pixel-level video prediction," *arXiv preprint arXiv:2106.13195*, 2021.
[49] O. Shouno, "Photo-realistic video prediction on natural videos of largely changing frames," *arXiv preprint arXiv:2003.08635*, 2020.
[50] H.-k. Chiu, E. Adeli, and J. C. Niebles, "Segmenting the future," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4202–4209, 2020.
[51] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[52] W. Byeon, Q. Wang, R. K. Srivastava, and P. Koumoutsakos, "Contextvp: Fully context-aware video prediction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 753–769.
[53] L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation," in *International Conference on Learning Representations Workshop*, 2015.
[54] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," in *International Conference on Learning Representations*, 2017.
[55] Z. Gao, C. Tan, and S. Z. Li, "Simvp: Simpler yet better video prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3170–3180.
[56] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European conference on computer vision*. Springer, 2016, pp. 649–666.
[57] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544.
[58] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *International Conference on Learning Representations*, 2018.
[59] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European conference on computer vision*. Springer, 2016, pp. 69–84.
[60] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.
[61] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1597–1607.
[62] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent - a new approach to self-supervised learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 21 271–21 284.
[63] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 22 243–22 255, 2020.
[64] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9929–9939.
[65] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," *arXiv preprint arXiv:2103.03230*, 2021.
[66] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.
[67] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
[68] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880.
[69] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.
[70] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.
[71] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.
[72] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
[73] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 347–10 357.
[74] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, "Video transformer network," *arXiv preprint arXiv:2102.00719*, 2021.
[75] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" *arXiv preprint arXiv:2102.05095*, 2021.
[76] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," *arXiv preprint arXiv:2103.15691*, 2021.
[77] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, "Multiscale vision transformers," *arXiv preprint arXiv:2104.11227*, 2021.
[78] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," *arXiv preprint arXiv:2106.13230*, 2021.
[79] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
[80] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
[81] X. Ding, X. Zhang, Y. Zhou, J. Han, G. Ding, and J. Sun, "Scaling up your kernels to 31x31: Revisiting large kernel design in cnns," *arXiv preprint arXiv:2203.06717*, 2022.
[82] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," *arXiv preprint arXiv:2201.03545*, 2022.
[83] H. Liu, Z. Dai, D. So, and Q. V. Le, "Pay attention to mlps," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9204–9215, 2021.
[84] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng, and S.-M. Hu, "Visual attention network," *arXiv preprint arXiv:2202.09741*, 2022.
[85] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Thirty-first AAAI conference on artificial intelligence*, 2017.
[86] S. Rasp, P. D. Dueben, S. Scher, J. A. Weyn, S. Mouatadid, and N. Thuerey, "Weatherbench: a benchmark data set for data-driven weather forecasting," *Journal of Advances in Modeling Earth Systems*, vol. 12, no. 11, p. e2020MS002203, 2020.
[87] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
[88] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 304–311.
[89] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 3. IEEE, 2004, pp. 32–36.
[90] Z. Chang, X. Zhang, S. Wang, S. Ma, Y. Ye, X. Xinguang, and W. Gao, "Mau: A motion-aware unit for video prediction and beyond," *Advances in Neural Information Processing Systems*, vol. 34, 2021.
[91] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[92] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Artificial intelligence and machine learning for multi-domain operations applications*, vol. 11006. International Society for Optics and Photonics, 2019, p. 1100612.
[93] 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*, vol. 13, no. 4, pp. 600–612, 2004.
[94] F. Research, "fvcore," https://github.com/facebookresearch/fvcore, 2021.
[95] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2019.
[96] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3634–3640.
[97] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 922–929.
[98] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *International conference on neural information processing*. Springer, 2018, pp. 362–373.
[99] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018.
[100] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *Advances in neural information processing systems*, vol. 33, pp. 17 804–17 815, 2020.
[101] H. Lin, Z. Gao, Y. Xu, L. Wu, L. Li, and S. Z. Li, "Conditional local convolution for spatio-temporal meteorological forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, 2022, pp. 7470–7478.
[102] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *CVPR*, June 2009.
[103] W. Lotter, G. Kreiman, and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," in *International Conference on Learning Representations*, 2016.
[104] S. Oprea, P. Martinez-Gonzalez, A. Garcia-Garcia, J. A. Castro-Vargas, S. Orts-Escolano, J. Garcia-Rodriguez, and A. Argyros, "A review on deep learning techniques for video prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
[105] Z. Hao, X. Huang, and S. Belongie, "Controllable video generation with sparse trajectories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7854–7863.
[106] F. A. Reda, G. Liu, K. J. Shih, R. Kirby, J. Barker, D. Tarjan, A. Tao, and B. Catanzaro, "Sdc-net: Video prediction using spatially-displaced convolution," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 718–733.
[107] B. Jin, Y. Hu, Q. Tang, J. Niu, Z. Shi, Y. Han, and X. 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*, 2020, pp. 4554–4563.
[108] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.
[109] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic adversarial video prediction," *arXiv preprint arXiv:1804.01523*, 2018.
[110] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," *Advances in neural information processing systems*, vol. 29, 2016.
[111] B. Jin, Y. Hu, Y. Zeng, Q. Tang, S. Liu, and J. Ye, "Varnet: Exploring variations for unsupervised video prediction," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 5801–5806.
[112] J. Lee, J. Lee, S. Lee, and S. Yoon, "Mutual suppression network for video prediction using disentangled features," *arXiv preprint arXiv:1804.04810*, 2018.

Cheng Tan received the B.S. degree from the the College of Information Engineering, Northwest A&F University, China, in 2021. He is currently pursuing the Ph.D. degree with the School of Engineering, Westlake University, Hangzhou, China. His main research interests include self-supervised learning and spatiotemporal learning.

Zhangyang Gao received the B.S. degree from the School of Automation, Central South University, Changsha, China, in 2020. He is currently a Ph.D. candidate at the School of Engineering, Westlake University, Hangzhou, China. His main research interests include clustering, unsupervised video tasks, and deep learning.

Stan Z. Li (Fellow, IEEE) received the B.Eng. degree from Hunan University, China, the M.Eng. degree from the National University of Defense Technology, China, and the Ph.D. degree from the University of Surrey, U.K. He is currently a Professor and the Director of the Center for Biometrics and Security Research (CBSR), Institute of Automation, Chinese Academy of Sciences (CASIA). From 2000 to 2004, he worked at Microsoft Research Asia, as a Researcher. Prior to that, he was an Associate Professor with Nanyang Technological University, Singapore. He has published over 200 papers in international journals and conferences and authored and edited eight books. His research interests include pattern recognition and machine learning, image and vision processing, face recognition, biometrics, and intelligent video surveillance. He was elevated to a fellow of the IEEE, for his contributions to the fields of face recognition, pattern recognition, and computer vision. He served as the Program Co-Chair for the International Conference on Biometrics in 2007 and 2009, and has been involved in organizing other international conferences and workshops in the fields of his research interest.