SimVPv2: Towards Simple yet Powerful Spatiotemporal Predictive Learning

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

***Abstract*—Recent years have witnessed remarkable advances in spatiotemporal predictive learning, with methods incorporating auxiliary inputs, complex neural architectures, and sophisticated training strategies. While SimVP has introduced a simpler, CNN-based baseline for this task, it still relies on heavy Unet-like architectures for spatial and temporal modeling, which still suffers from high complexity and computational overhead. In this paper, we propose SimVPv2, a streamlined model that eliminates the need for Unet architectures and demonstrates that plain stacks of convolutional layers, enhanced with an efficient Gated Spatiotemporal Attention mechanism, can deliver state-of-the-art performance. SimVPv2 not only simplifies the model architecture but also improves both performance and computational efficiency. On the standard Moving MNIST benchmark, SimVPv2 achieves superior performance compared to SimVP, with fewer FLOPs, about half the training time, and 60% faster inference efficiency. Extensive experiments across eight diverse datasets, including real-world tasks such as traffic forecasting and climate prediction, further demonstrate that SimVPv2 offers a powerful yet straightforward solution, achieving robust generalization across various spatiotemporal learning scenarios. We believe the proposed SimVPv2 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**

I. Introduction

A wise person can foresee the future, and so should an intelligent vision model. In recent years, spatiotemporal predictive learning has gained significant attention due to its ability to infer the future by leveraging the underlying patterns embedded in spatiotemporal data, which reflect the complex and often chaotic dynamics of the real world [1]–[12]. Despite its vast potential, this task presents substantial challenges due to the inherent complexity and randomness in the data, such as non-linear dynamics, long-term dependencies, and high-dimensionality. To address these challenges, numerous methods have emerged, introducing a variety of novel operators, sophisticated neural architectures, and advanced training strategies aimed at improving the accuracy and efficiency of predictions. Many of these methods achieve impressive performance gains by incorporating recurrent networks [1], [3], [13], transformer-based models [14], [15], and complex autoregressive or normalizing flow models [16]. Additionally, various training techniques, such as adversarial learning [17], have been adopted to improve the fidelity of generated future frames. However, these advances come at a cost: increasing model complexity. As models become more intricate, they require more computational resources, are more challenging to train and scale. This raises a fundamental question: Can a simple model without recurrent units achieve comparable performance, while offering better scalability and interpretability?

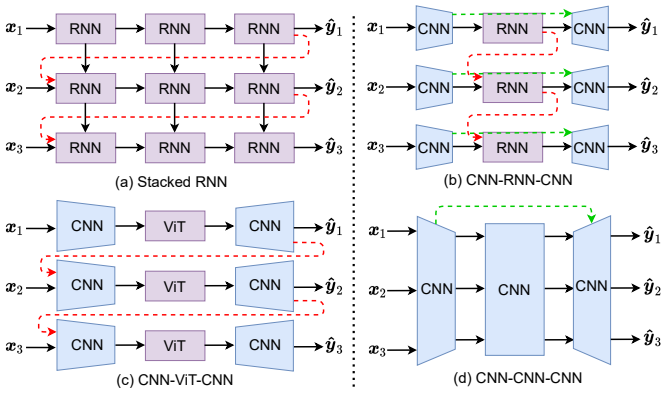


Fig. 1. Major categories of the architectures for spatiotemporal predictive learning. The red and blue dotted line are available to learn the temporal evolution and spatial dependency. Our proposed SimVP and SimVPv2 belong to (d) CNN-CNN-CNN, which can outperform other state-of-the-art methods.

Our previous work, SimVP, answers this question by introducing a pure convolution neural network (CNN) model that demonstrates the possibility of removing recurrent units without sacrificing predictive performance. To provide a clearer understanding of existing works in spatiotemporal predictive learning, we categorize current approaches into four broad groups, as illustrated in Fig. 1. The first category consists of stacked recurrent neural network (RNN) models, which rely solely on recurrent units for both spatial and temporal modeling. The second group includes CNN-RNN hybrid models, which combine CNN for spatial feature extraction with RNNs for modeling temporal dynamics. The third group comprises CNN-ViT models, which integrate vision Transformer (ViT) to capture global temporal relationships using attention mechanisms, while CNNs are responsible for spatial feature extraction and reconstruction. The final category includes fully convolutional models that use CNNs for both spatial and temporal modeling and thus eliminate the need for recurrent units. SimVP belongs to this category, offering a simpler alternative by relying entirely on convolutional layers.

* Equal contribution.

Cheng Tan, Zhangyang Gao and Siyuan Li are with Zhejiang University, Hangzhou, China, and also with the AI Lab, Research Center for Industries of the Future, Westlake University. Email: {tancheng, gaozhangyang, lisiyuan}@westlake.edu.cn.

Stan Z. Li is with the AI Lab, Research Center for Industries of the Future, Westlake University. Email: Stan.ZQ.Li@westlake.edu.cn.

TABLE I
REPRESENTATIVE SPATIOTEMPORAL PREDICTIVE LEARNING WORKS SINCE 2014.

	(a) Stacked RNN	(b) CNN-RNN-CNN	(c) CNN-ViT-CNN	(d) CNN-CNN-CNN
2014-2015	ConvLSTM [1], Composite LSTM [4], PGP [18]	AE-ConvLSTM-flow [19], PGN [20]	-	GDL [17]
2016-2017	FSTN [21], PredRNN [13], Hierarchical ConvLSTM [22], RLadder [23]	MCnet [24], CDNA [25], SV2P [2], Dual motion GAN [26], DrNet [27]	-	Amersfoort et al. [28], DVF [29], EEN [30]
2018-2019	Znet [31], ConvLSTM-DTD [32], dGRU [33], DDPAE [34], PredRNN++ [35], MIM [3]	E3D-LSTM [36], EPVA [37], SVG-LP [38], CrevNet [16], hierarchical-VRNN [39]	Weissenborn et al. [14]	DPG [40], PredCNN [41], Retrospective Cycle GAN [42]
2020-2024	PredRNNv2 [43], MAU [44], SwinLSTM [45]	Hu et al. [46], PhyDNet [47], FitVid [48], STAM [49], STRPM [50]	LVT [15], VMRNN [51]	G-VGG [52], Chiu et al. [53], DMVFN [54], MMVP [55]

As shown in Table I, we have gathered a range of representative works in spatiotemporal predictive learning, from which we observe that recurrent-based architectures (Fig.1 a-b) have historically been the dominant choice. These architectures, particularly those based on recurrent units, have led the field for years due to their success in handling sequential data. Inspired by the success of long short-term memory (LSTM) [56] 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 [13] 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], [35], [36], [43], [57]. PhyDNet [47] combined ConvLSTM with a two-branch architecture involving PhyCells, which incorporated physical dynamics through partial differential equations, helping guide the model with physics-based constraints. Similarly, CrevNet [16] proposed an invertible two-way autoencoder based on normalizing flow, introducing a conditionally reversible architecture that allowed better information preservation across time steps. Despite the success of these recurrent-based architectures, they face several inherent limitations. The sequential nature of RNNs poses significant computational challenges, particularly for long-term predictions. As each time step is processed sequentially, computations cannot be fully parallelized, leading to inefficiencies in training and inference. These limitations motivate the exploration of simpler and more efficient architectures that can handle spatiotemporal dynamics without the computational overhead of recurrence.

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 [42], teacher-student distilling [53], and optical flow [40]. In an effort to simplify the landscape of spatiotemporal predictive learning, SimVP was introduced as a fully convolutional architecture that used common components such as convolutional networks for both spatial and temporal modeling, simple shortcut connections for efficient feature propagation and was trained end-to-end with mean squared error loss. SimVP proved to be successful in demonstrating that pure CNN with Unet-insipred multi-scale processing could achieve comparable performance.

Despite the success of SimVP, however, the reliance on Unet architectures introduced its own set of challenges. These multi-scale processing frameworks, with their top-down and bottom-up paths and skip connections, are still computationally expensive and complex. While SimVP streamlined many aspects of spatiotemporal predictive learning, the complexity of the Unet structure limited its efficiency, particularly in terms of computation and model simplicity.

Building on the foundation laid by SimVP, we propose SimVPv2, a further simplified model that completely eliminates the need for Unet architectures. Instead of relying on complex multi-scale and skip-connection frameworks, SimVPv2 introduces a novel and efficient Gated Spatiotemporal Attention (gSTA) mechanism. The gSTA module captures both spatial and temporal dependencies without the overhead of hierarchical or multi-branch processing, achieving a streamlined architecture that is more computationally efficient. As a result, SimVPv2 significantly reduces the number of parameters and FLOPs, leading to shorter training times and increased inference efficiency, while maintaining or improving predictive accuracy. Extensive experiments across diverse datasets demonstrate that SimVPv2 not only surpasses SimVP in terms of performance but also generalizes effectively across a wide range of spatiotemporal predictive tasks.

A preliminary version of this work was published in [58]. This journal paper extends it in the following aspects:

- We introduce SimVPv2, a more streamlined and efficient architecture that completely removes the need for Unet structures, replacing them with gSTA modules to capture both spatial and temporal dependencies with greater computational efficiency.
- We reproduce the mainstream spatiotemporal predictive learning methods into a unified framework and systematically evaluate performance on the standard benchmark Moving MNIST dataset in consideration of computational cost and time complexity.
- We conduct a comprehensive evaluation across a broader set of eight benchmark datasets, including both synthetic and real-world tasks. This comprehensive assessment demonstrates the robustness and generalization capabilities of the proposed approach across various spatiotemporal predictive learning challenges.

We release our code at github.com/chengtan9907/SimVPv2.

II. Related work

A. Stacked RNN

As shown in Fig. 1 (a), stacked RNN architectures have been widely adopted for spatiotemporal predictive tasks. These methods typically involve the design of novel recurrent units (local) and sophisticated architectural frameworks (global) to model temporal dependencies in sequential data. Recurrent Grammar Cells [18] 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 [13] suggests simultaneously extracting and memorizing spatial and temporal representations. PredRNN++ [35] 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 [33] 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 [43] extends PredRNN by introducing a decoupling loss and a reverse scheduled sampling method. MAU [44] leverages an attention mechanism to capture the correlations between the current spatial state and historical states, aggregating motion and appearance to improve predictions. SwinLSTM [45] merges the strengths of Swin Transformer blocks with a simplified LSTM architecture and replaces the convolutional structure of ConvLSTM with the self-attention mechanism from transformers.

B. 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 [19] incorporates ConvLSTM and an optical flow predictor to capture changes over time. Conditional VRNN [39] combines CNN encoder and RNN decoder in a variational generating framework. E3D-LSTM [36] applies 3D convolution for encoding and decoding and integrates it into latent RNNs for obtaining motion-aware and short-term features. CrevNet [16] 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 compressed features for accurate and efficient prediction. FitVid [48] employs a conditional variational model, using LSTMs to predict hidden states in a probabilistic manner. STAM [49] utilizes 3D convolutional layers in recurrent units to jointly learn high-level semantic features and low-level texture representations. STRPM [50] introduces Residual Predictive Memory (RPM), which focuses on modeling the spatiotemporal residuals between consecutive frames. Additionally, STRPM is trained with generative adversarial networks and incorporates a learned perceptual loss, improving the perceptual quality of the generated frames.

C. CNN-ViT-CNN

This framework introduces Vision Transformer (ViT) to model latent dynamics. By extending language transformer [59] to ViT [60], a wave of research has been sparked recently. DeiT [61] and Swin Transformer [62] have achieved superior performance on various computer vision tasks. The great success of image transformers has inspired the investigation of video transformers. VTN [63] applies sliding window attention on temporal dimension following a 2D spatial feature extractor. TimeSformer [64] and ViViT [65] explore various strategies for space-time attention. MViT [66] introduces a multiscale pyramid feature extractor, capturing both fine-grained and high-level temporal patterns across varying resolutions. Video Swin Transformer [67] extend Swin Transformer from 2D to 3D, utilizing shiftable local attention windows to balance speed and accuracy. VMRNN [51] proposes the VMRNN cell, a recurrent unit that integrates the strengths of Vision Mamba [68] blocks with LSTM. Many of these works focus on video generation using ViTs [14], [15]. For instance, MAGViT [10] introduces a 3D tokenizer that quantizes video into spatiotemporal tokens and utilizes non-autoregressive decoding to generate tokens. OmniTokenizer [11] trains on fixed-resolution image data to develop strong spatial encoding capabilities and then jointly trains on both image and video data at multiple resolutions to learn temporal dynamics.

D. CNN-CNN-CNN

This framework is not as popular as the previous one because it is so simple that complex modules and training strategies are required to improve performance. DVF [29] learns the voxel flow by an autoencoder to reconstruct a frame by borrowing voxels from nearby frames. DMVFN [54] proposes a routing module to dynamically select a sub-network. PredCNN [41] combines cascade multiplicative units with CNN to capture inter-frame dependencies. DPG [40] disentangles motion and background via a flow predictor and a context generator. G-VGG [52] uses a hierarchical model to make predictions at different scales and train the model with adversarial and perceptual loss. Chiu et al. [53] encodes RGB frames from the past and decodes the future semantic segmentation using teacher-student distilling. MMVP [55] decouples motion and appearance information by constructing appearance-agnostic motion matrices.

E. Summary

Although mainstream reucrrent-based approaches have made significant progress, the reliance on recurrent units still imposes considerable computational overhead, and their architectures face challenges in parallelization. ViT-based models, while powerful, demand substantial computational resources. Fully CNN-based models offer a more efficient solution but depend on complex techniques. SimVP represents an important step in this direction, demonstrating the potential of fully convolutional architectures. However, its reliance on Unet-like structures remains a source of computational complexity. SimVPv2 builds on the simplicity of fully CNN-based models, taking it a step further by removing the need for complex components such as Unet, recurrent units, or transformers.

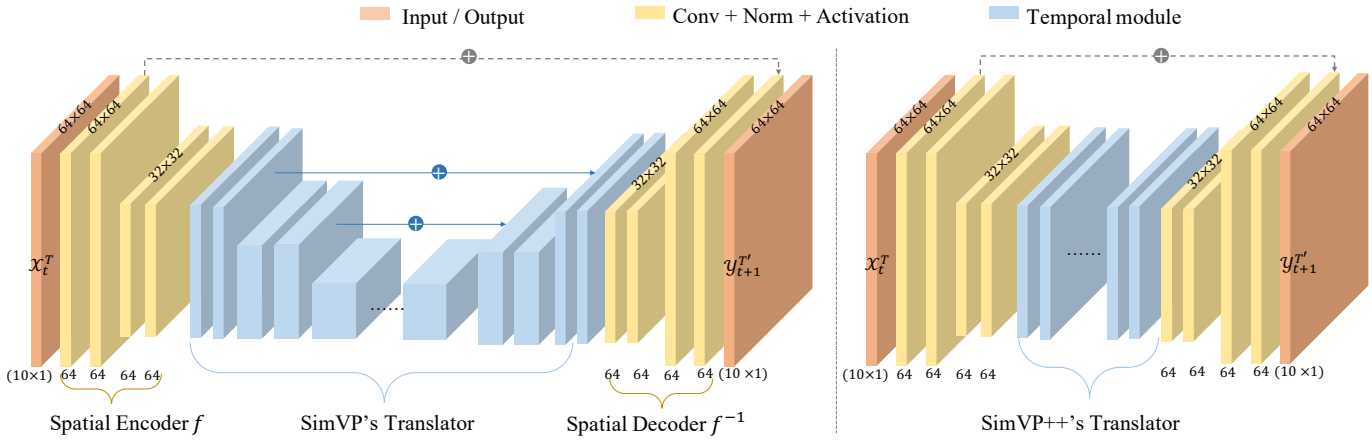


Fig. 2. The overall framework of SimVP and SimVPv2.

III. 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.

IV. Method

A. Motivation

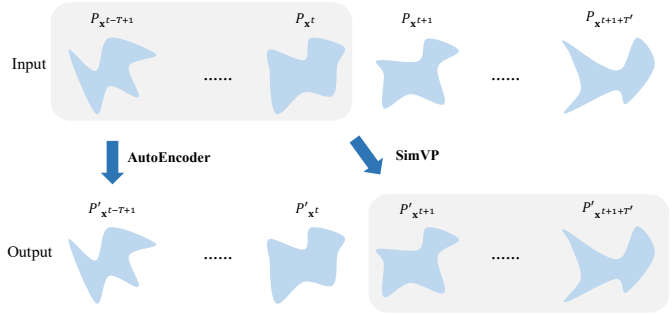


Fig. 3. The schematic diagram of the autoencoder and our proposed SimVP. While the autoencoder focuses on a single frame at a static time, SimVP concerns a sequence of frames at a dynamic time. The first row denotes the ground-truth frames, and the second denotes the predicted frames. From left to right,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

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

B. Overview

Taking input Moving MNIST data as an example, we provide an overview of SimVP and SimVPv2 models, as illustrated in Fig. 2. They share a similar architecture but SimVPv2 introduces a significantly streamlined design which eliminates the Unet multi-scale processing and complex hierarchical structures. 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 decodes the latent space into the predicted 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 are as follows:

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

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:

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

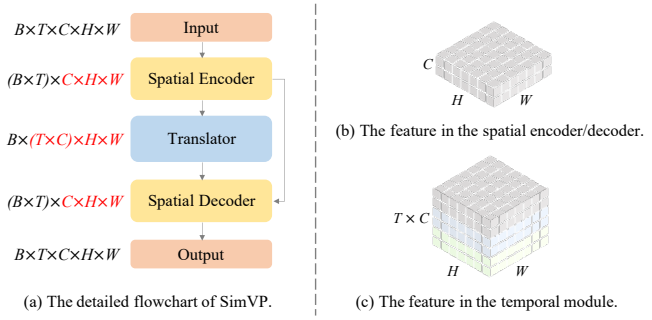
where unConv2d is a transposed convolutional layer or pixelshuffle layer if it needs upsampling. Otherwise, it is a convolutional layer with stride one.

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

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

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:

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



(a) The detailed flowchart of SimVP. (b) The feature in the spatial encoder/decoder. (c) The feature in the temporal module.

Fig. 4. 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. 4 (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. 4 (c). By forcing the designed temporal module built upon convolutional networks to learn from stacks of multi-frame features, SimVP can capture the intrinsic temporal evolutions inside the sequential data.

C. 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.

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. 5 (a), our Inception temporal module is different from the original Inception module [69] 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:

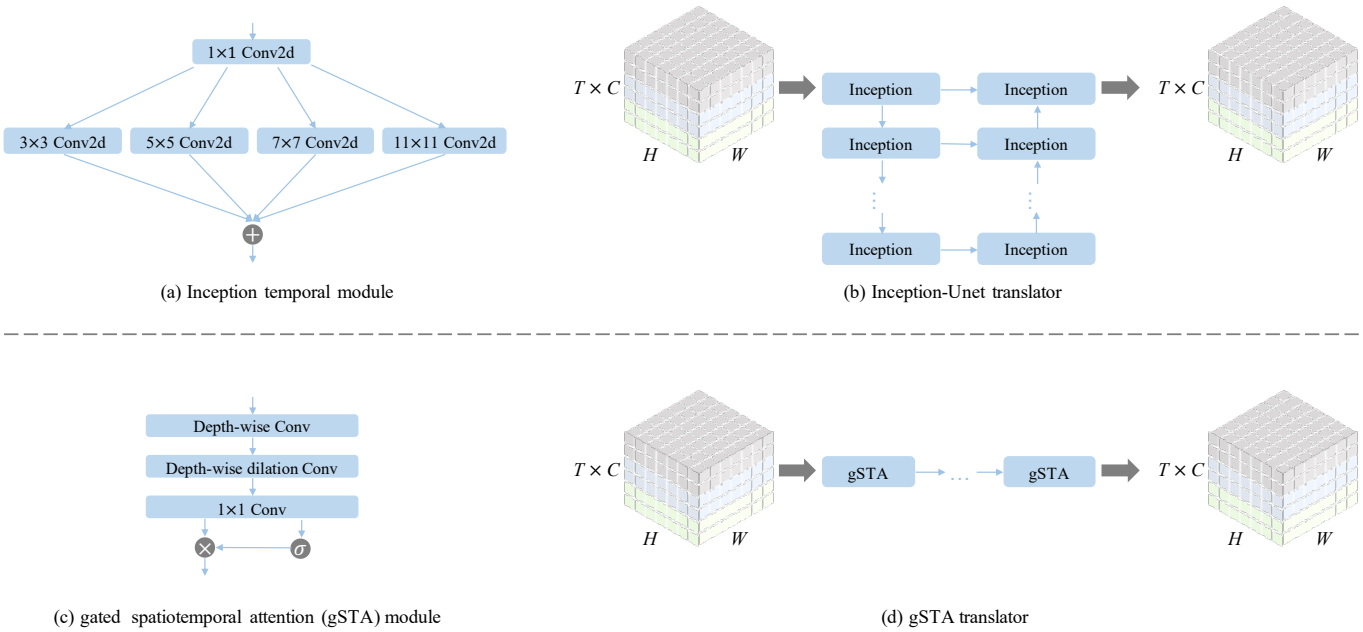
$$\hat{z}^j = \text{Conv2d}_{1\times1}(z^j), \quad (10)$$

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

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 [70].

2) Gated Spatiotemporal Attention Translator: In this journal version, 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



(a) Inception temporal module

(b) Inception-Unet translator

(c) gated spatiotemporal attention (gSTA) module

(d) gSTA translator

Fig. 5. (a-b) The Inception temporal module and corresponding Inception-Unet translator. (c-d) The gSTA module and corresponding gSTA translator.

obtaining large effective receptive fields and higher shape bias rather than texture bias [71]–[74]. 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.

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 [71], [72], [74] 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. 6.

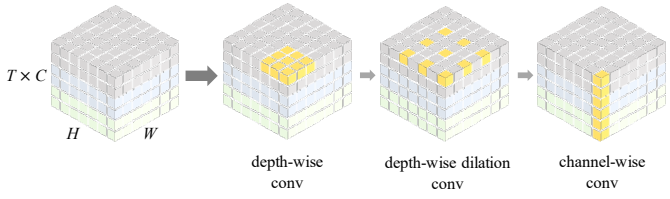


Fig. 6. The large kernel convolution in the gated spatiotemporal attention module. The yellow region denotes the receptive field.

The gated spatiotemporal attention (gSTA) module is illustrated in Fig. 5 (c). 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:

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

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

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

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

We show the gSTA translator in Fig. 5 (d). 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. The gated attention $\sigma(g)$ gate is used to adaptively select the informative features and filter unimportant features from a spatiotemporal perspective.

D. Advantages of gSTA over IncepU

One key advantage of gSTA is the **enhanced receptive field coverage** achieved through the use of large kernel convolutions and dilated convolutions, enabling the module to capture long-range dependencies more effectively. Another important benefit of gSTA is its ability to **dynamically select informative features through the gating mechanism**, which acts as an adaptive filter that enhances significant features while suppressing less relevant ones, allowing the model to focus on the most salient spatiotemporal patterns. Moreover, gSTA offers **reduced computational complexity and improved parameter efficiency**. Unlike IncepU's multi-branch architecture, which introduces multiple convolutional pathways and increases the computational burden, gSTA simplifies the architecture by employing a single unified path with large kernel convolutions and gating mechanisms.

TABLE II
QUANTITATIVE RESULTS OF DIFFERENT METHODS ON THE MOVING MNIST DATASET (10 → 10 FRAMES). NOTE THAT "-S" DENOTES THE SMALLER MODEL AND "-L" DENOTES THE LARGER MODEL. WE REPORT "SIMVPV2-S×3" BY TRAINING "SIMVPV2-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	19.43	261	27.15	32.22±0.02	89.19±0.33	0.927±0.000
SimVPv2-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
SwinLSTM	69.87	820	6.51	27.44±0.08	78.69±0.29	0.938±0.000
MMVP	93.55	402	21.33	33.29±0.02	89.61±0.23	0.926±0.000
SimVPv2-S×10	16.53	1560	**44.09**	**15.05**±0.03	**49.80**±0.10	**0.967**±0.000
SimVPv2-S×5	16.53	780	**44.09**	16.47±0.02	53.24±0.04	0.964±0.000
SimVPv2-S×3	16.53	468	**44.09**	22.37±0.06	67.52±0.03	0.951±0.000
SimVPv2-L	152.20	796	21.23	21.81±0.03	66.43±0.04	0.952±0.000

V. EXPERIMENTS

The experiments are conducted on various datasets with different settings to evaluate from the following aspects:

- Standard spatiotemporal predictive learning (Section V-A). 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 **Moving MNIST [4], TaxiBJ [75], and WeatherBench [76]**.
- Generalization ability across different datasets (Section V-B). Generalizing the learned knowledge to other domains is a challenge. We investigate such ability by training the model on the **KITTI [77]** and evaluating it on the **Caltech Pedestrian [78].**
- Predicting frames with flexible lengths (Section V-C). One of the advantages of recurrent units is that they can easily handle flexible-length frames like the **KTH** [79]. 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.
- Challenging multi-domain evaluation (Section V-D). **RoboNet** [80] is designed for robotic action planning, containing a diverse collection of robot interactions across various environments. **BridgeData** [81], [82] is a multi-domain dataset including 71 distinct tasks across 10 different scenes. They pose significant challenges due to their need for adapting to various environments.

A. Standard spatiotemporal predictive learning

1) Moving MNIST: In this dataset, each video is generated 20 frames long and consists of two digits inside a 64×64 patch. The digits are randomly selected 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.

We evaluate state-of-the-art methods with the same protocol for fair comparisons, including ConvLSTM [1], PredRNN [13], PredRNN++ [35], MIM [3], E3D-LSTM [36], PhyDNet [47], CrevNet [16], MAU [44], SwinLSTM [45], and MMVP [55]. By using ConvLSTM with different sizes as the standard baselines, we divide these models into two groups according to their computational cost, as shown in Table II. ConvLSTM-S is the small model with four layers with a hidden size of 64, and ConvLSTM-L is the large model with four layers with a hidden size of 192. Models are trained using the Adam optimizer [83] with the OneCycle learning rate scheduler [84] for 200 epochs. Following [58], 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 evaluate the performance by mean square error (MSE), mean absolute error (MAE), and structural similarity index (SSIM) [85]. We repeat each experiment for three trials and report the average. FLOPs are reported using fvcore [86]. The training time is reported by computing the average seconds for training an epoch and the inference efficiency is reported by inferencing 10,000 test samples with a batch size of 1 and computing the average testing frames per second (FPS). Experiments are conducted on a single NVIDIA V100 GPU.

Table II shows the performance on the Moving MNIST dataset. SimVPs achieve highly competitive performance compared to state-of-the-art. For the small model group (the first five rows in Table II), SimVPv2-S obtains the best prediction quality with the fastest training time and the highest inference efficiency. For the large model group (from the sixth to the last row in Table II), SimVPv2-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 SimVPv2-S×3 that simply trains SimVPv2-S with three times epochs, i.e., 600 epochs. Surprisingly, SimVPv2-S×3 achieves competitive performance as well as SimVPv2-L. Morover, SimVPv2-S×3 still has the least training time compared with large models.

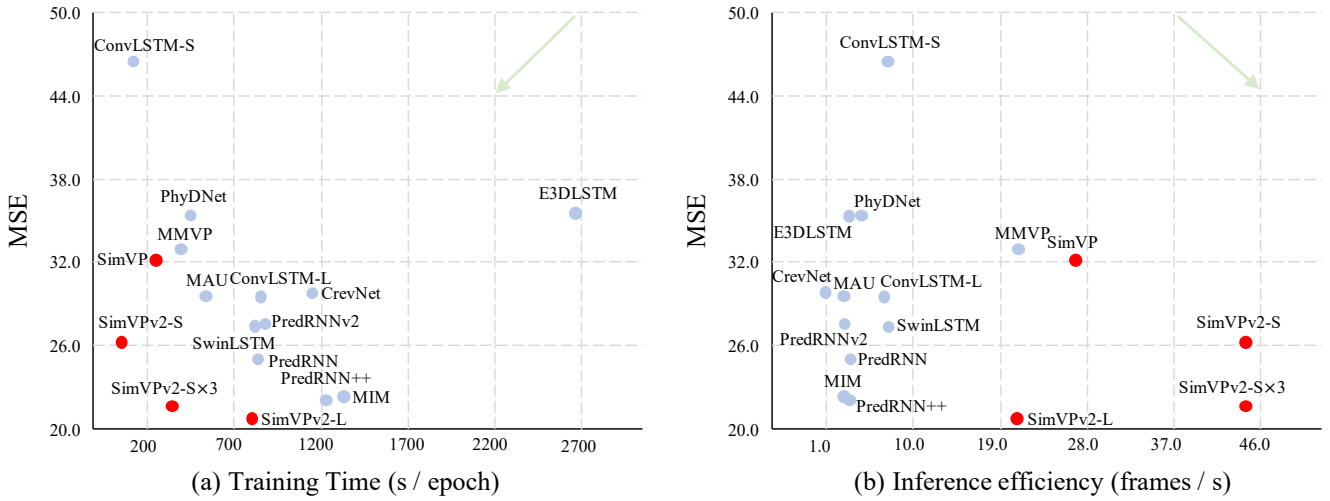


Fig. 7. The performance of SimVPs on the Moving MNIST dataset. The variants of SimVP are denoted in red color. For the training time, the less the better. For the inference efficiency (frames per second), the more the better. The light green arrow indicates the direction of model optimization.

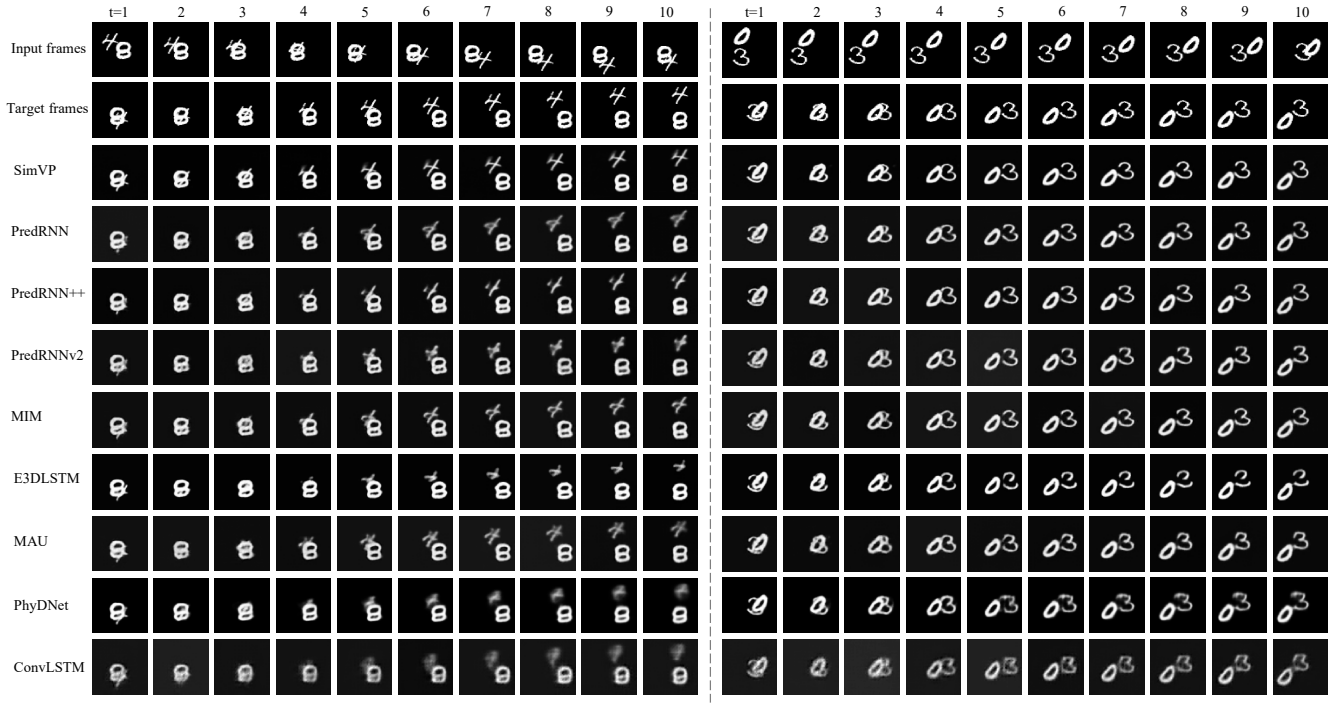


Fig. 8. Examples of predicted results on the Moving MNIST dataset. We denote SimVPv2 as SimVP for convenience here.

We plot the performance vs. training time and the performance vs. inference efficiency in Fig. 7. In Fig. 7(a), both the training time and MSE are the lower the better. We can see that SimVPs are concentrated in the lower-left corner of this plot. SimVPv2-S even takes about only one-sixteenth training time of E3D-LSTM and obtains significantly better performance. In Fig. 7(b), the inference efficiency is the higher the better. SimVPs are concentrated in the lower-right corner. SimVPs outperform other models and are the only model that can achieve more than 10 FPS. SimVPv2-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 SimVPs outperform state-of-the-art methods in both training and inference efficiency.

Fig. 8 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.

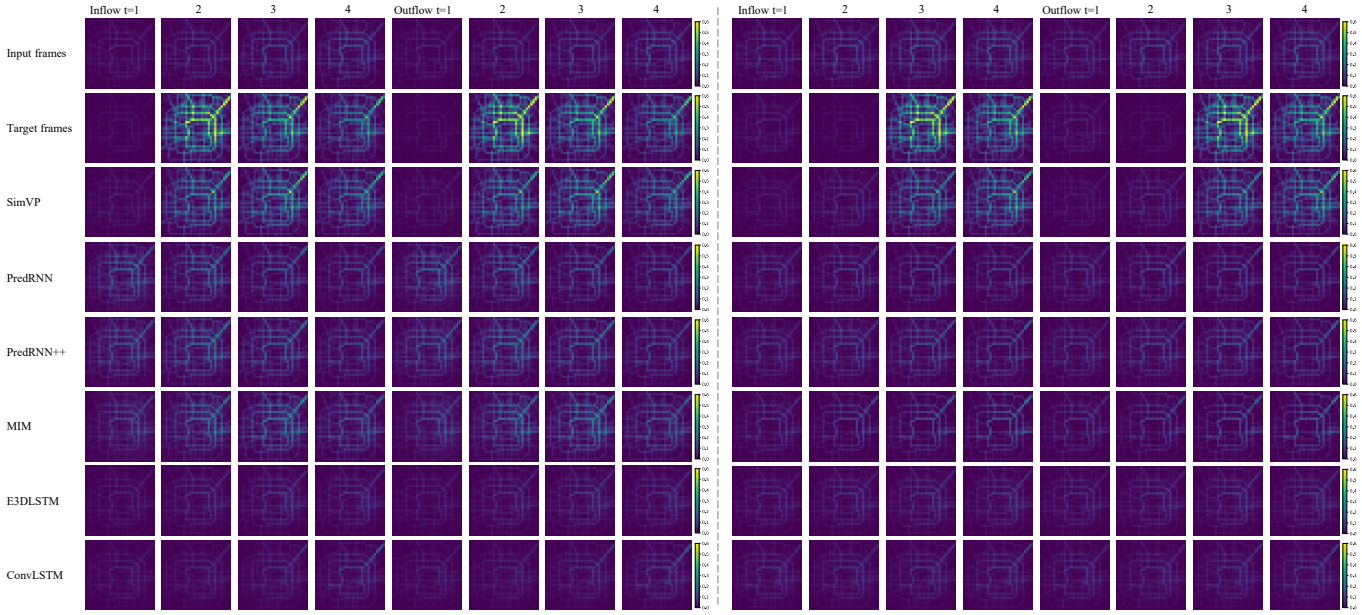


Fig. 9. Examples of predicted results on the TaxiBJ dataset. We denote SimVPv2 as SimVP for convenience here.

2) TaxiBJ: We use the TaxiBJ dataset [75] to evaluate the traffic forecasting ability. TaxiBJ contains the trajectory data in Beijing collected from taxicab GPS with two channels, i.e., inflow or outflow defined in [75]. 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 III
QUANTITATIVE RESULTS ON THE TAXIBJ DATASET ($4 \rightarrow 4$ FRAMES).

Method	TaxiBJ		
	MSE $\times$ 100↓	MAE↓	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	41.4	16.2	0.982
SimVPv2	**34.8**	**15.6**	**0.984**

As shown in Table III, SimVP outperforms the previous recurrent-based state-of-the-art methods by a small margin. SimVPv2 which is introduced in this paper further stretches the margin between baseline models, quantitatively improving SimVP by about 15.94% in the MSE metric and about 3.7% in the MAE metric. Benefiting from the gated spatiotemporal attention mechanism, SimVPv2 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 in Fig. 9. 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. 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 SimVP. SimVP learns the spatiotemporal dynamics in a way that is consistent with the real-world situation. In contrast, recurrent-based methods over-depend on the previous frames, and they are not able to directly capture the long-range dependencies in such complex traffic flows.

3) WeatherBench: We employ our model in the climate prediction on the WeatherBench [76]. 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. We choose the temperature prediction task to evaluate our model. Following the protocol from [76], we train the model using data from 1979 to 2015 and validate the model using data from 2016. The 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 [87], STGCN [88], MSTGCN [89], ASTGCN [89], GCGRU [90], DCRNN [91], AGCRN [92], CLCSTN [93], and CLCRN [93]. Additional comparisons with spatiotemporal predictive learning methods such as ConvLSTM [1], PredRNN [13], and PredRNN++ [35] are included.

We report the quantitative results in Table IV. 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 IV. It can be seen that SimVP outperforms baselines 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.

TABLE IV
QUANTITATIVE RESULTS ON THE WEATHERBENCH DATASET (12 → 12 FRAMES).

Method	WeatherBench MAE↓	WeatherBench 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	0.7882	1.1483
SimVPv2	**0.7475**	**1.0785**

B. 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.

TABLE V
QUANTITATIVE RESULTS ON THE CALTECH PEDESTRIAN DATASET (10 → 1 FRAME).

Method	Caltech Pedestrian SSIM↑	Caltech Pedestrian PSNR↑	Caltech Pedestrian 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	0.940	33.1	3.81
SimVPv2	**0.949**	**33.2**	**3.11**

Following [16], [94], [95], several strong baselines are selected for comparison, including BeyondMSE [17], MCnet [24], DVF [29], Dual-GAN [26], CtrlGen [96], PredNet [94], ContextVP [57], SDC-Net [97], rCycleGan [42], DPG [40], CrevNet [16] and STMFANet [98]. SSIM [85], PSNR, and LPIPS [99] metrics are used in the evaluation phase. As shown in Table V, SimVP has achieved better performance than baseline models by a large margin. Specifically, SimVP outperforms STMFANet by about 1.04% in the SSIM, approximately 13.74% in the PSNR, and about 35.31% in the LPIPS. SimVPv2 further improves the SimVP and obtains the best performance among SSIM, PSNR, and LPIPS.

C. Predicting frames with flexible lengths

We choose the KTH dataset [79] for evaluating the flexibility. It contains 25 individuals performing 6 types of actions, i.e., walking, jogging, running, boxing, hand waving, and hand clapping. Following [36], [95], 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. Models are trained to predict the next 20 or 40 frames from the previous 10 observations. Strong baselines are included, such as MCnet [24], ConvLSTM [1], SAVP, SAVP-VAE [100], VPN [101], DFN [102], fRNN [33], Znet [31], SV2P [2], PredRNN [13], VarNet [103], PredRNN++ [35], MSNET [104], E3d-LSTM [36], and STMFANet [98]. We compare the predicted quality with these state-of-the-art baselines under both 10 → 20 frames and 10 → 40 frames cases.

TABLE VI
QUANTITATIVE RESULTS ON THE KTH DATASET (10 → 20/40 FRAMES).

Method	KTH (10 → 20) SSIM↑	KTH (10 → 20) PSNR↑	KTH (10 → 40) SSIM↑	KTH (10 → 40) 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	0.905	33.72	0.886	32.93
SimVPv2	**0.913**	**34.24**	**0.895**	**33.35**

We show the quantitative results in Table VI. It can be seen that SimVPs are superior to those baselines. Moreover, SimVPs even accurately predict the future frames under the extremely long-range case like 10 → 40 frames. We show the qualitative results on the KTH dataset with output frame lengths of 40 in Fig. 10. In general, SimVP can predict the overall posture in almost every frame, albeit with a slight blur. The performance comes to the recursive prediction strategy that mitigates the issue of error accumulation. In typical recurrent models, predictions are generated frame by frame, leading to sequential error accumulation at each time step, which can severely degrade the quality of long-term forecasts. In contrast, our recursive approach reduces the frequency of error propagation by accumulating errors only over four iterations, as opposed to forty iterations in recurrent methods when predicting 40 frames ahead. Furthermore, gSTA modules dynamically adjusts its attention weights based on the evolving spatiotemporal patterns, enabling the model to adapt to changes in motion dynamics throughout the prediction sequence. Additionally, SimVP's architecture facilitates the seamless integration of spatiotemporal context, allowing the model to jointly learn spatial and temporal features without treating them as separate components.

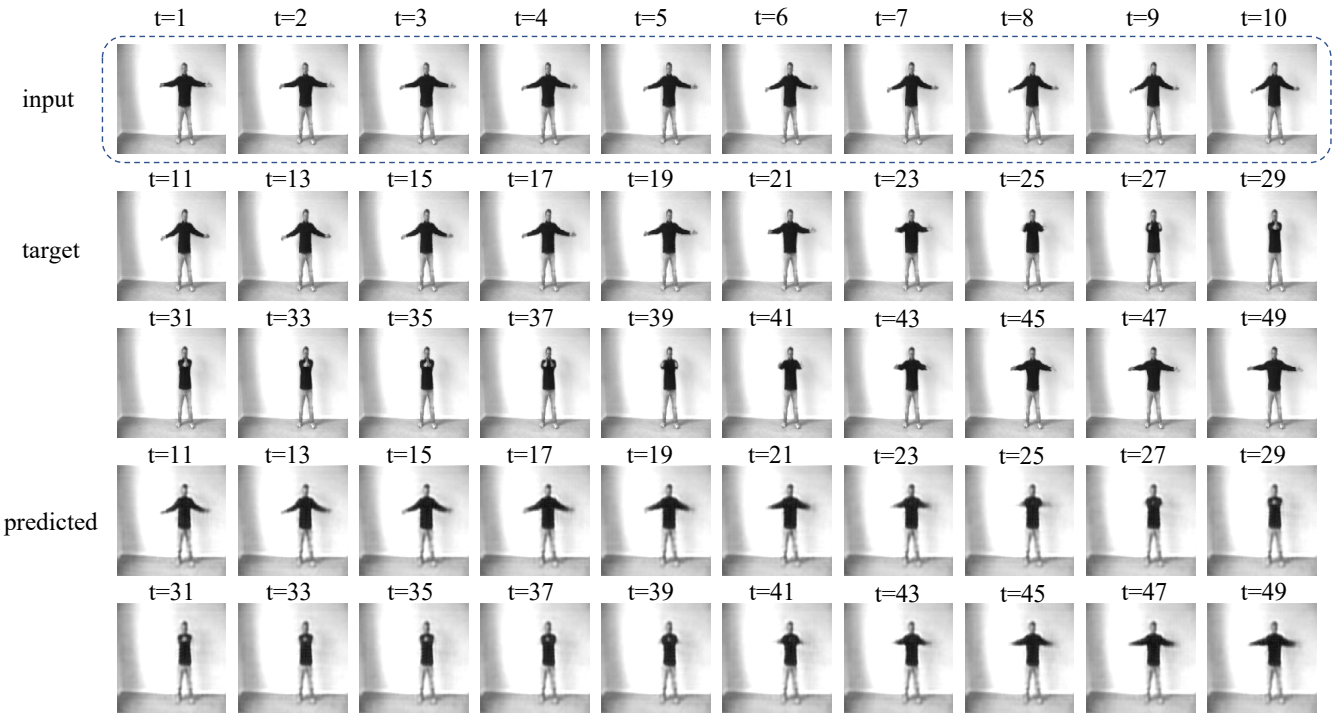


Fig. 10. An example of predicted results on the KTH dataset.

D. Challenging multi-domain evaluation

To thoroughly assess the robustness and adaptability of models, we conduct evaluations on two challenging multi-domain datasets: RoboNet and BridgeData. These datasets are specifically chosen due to their diverse range of tasks and environments, which present significant challenges. Both datasets involve predicting the next 10 frames given the first 2 input frames. We compare SimVPv2 with ConvLSTM, E3D-LSTM, MAU, PhyDNet, PredRNN, PredRNN++, and PredRNNv2 using SSIM and PSNR metrics. Table VII presents the quantitative results for the RoboNet and BridgeData datasets. The results demonstrate that SimVPv2 consistently outperforms other baseline models across both datasets, achieving the highest SSIM and PSNR values.

On the RoboNet dataset, SimVPv2 achieves an SSIM of 0.856 and a PSNR of 22.78, surpassing the performance of recurrent-based models such as PredRNN++ (SSIM 0.849, PSNR 22.66) and PredRNNv2 (SSIM 0.847, PSNR 22.52). On the BridgeData dataset, SimVPv2 attains an SSIM of 0.865 and a PSNR of 22.62, outperforming all other methods. Notably, it outperforms PredRNN++ (SSIM 0.856, PSNR 22.34) and ConvLSTM (SSIM 0.832, PSNR 21.36), indicating that the model can effectively generalize across a variety of domains and tasks with different visual characteristics. The superior performance of SimVPv2 on the challenging multi-domain evaluation illustrates the versatility and robustness of SimVPv2, establishing it as a strong baseline for future research in spatiotemporal predictive learning. The results suggest that even in complex, multi-domain settings, a streamlined model like SimVPv2 can outperform more complex architectures, reinforcing the benefits of simplicity and efficiency.

TABLE VII
QUANTITATIVE RESULTS ON THE ROBONET ($2 \rightarrow 10$ FRAMES) AND BRIDGEDATA DATASETS ($2 \rightarrow 10$ FRAMES).

Method	RoboNet ($2 \rightarrow 10$)		BridgeData ($2 \rightarrow 10$)	
	SSIM↑	PSNR↑	SSIM↑	PSNR↑
ConvLSTM	0.836	22.15	0.832	21.36
E3D-LSTM	0.827	21.82	0.784	20.36
MAU	0.843	22.40	0.821	21.14
PhyDNet	0.797	20.92	0.762	19.61
PredRNN	0.850	22.63	0.853	22.19
PredRNN++	0.849	22.66	0.856	22.34
PredRNNv2	0.847	22.52	0.850	22.01
SimVP	0.854	22.73	0.863	22.60
SimVPv2	**0.856**	**22.78**	**0.865**	**22.62**

E. Ablation study

1) The quantitative analysis of spatiotemporal translator: The flexibility of our framework allows for the seamless integration of various spatiotemporal translators, enabling a comprehensive comparison of different architectures. We explore the adaptability of our model by replacing the spatiotemporal translator with different architectures, including vanilla ViT [60], Swin Transformer [62], Poolformer [105], MLPMixer [106], and ConvMixer [107]. This analysis helps to evaluate the impact of spatiotemporal translator choices. The results indicate that the gSTA module consistently outperforms other architectures across all evaluation metrics, achieving the lowest MSE and MAE values while also obtaining the highest SSIM score. Notably, gSTA achieves these results with comparable FLOPs and parameter count relative to other competitive architectures, such as Swin Transformer and MLPMixer. This

suggests that gSTA effectively captures spatiotemporal dependencies with greater accuracy and efficiency. In comparison, ConvMixer shows the lowest FLOPs and parameter, making it the most lightweight option; however, its predictive performance is lower than that of gSTA and some other architectures. Swin Transformer and MLPMixer also demonstrate strong predictive capabilities, with MLPMixer achieving an MSE close to gSTA but requiring more computational resources. IncepU, on the other hand, exhibits higher computational complexity and does not match the performance with gSTA.

Overall, these results underscore the effectiveness of the gSTA module in capturing complex spatiotemporal relationships while maintaining computational efficiency, making it a suitable choice for high-performance temporal modeling.

TABLE VIII
ABLATION STUDY ON THE MOVING MNIST DATASET.

Method	Params (M)	FLOPs (G) ↓	MSE ↓	MAE ↓	SSIM ↑
ViT	46.1	16.9	35.15	95.87	0.914
Swin Transformer	46.1	16.4	29.70	84.05	0.933
Poolformer	37.1	14.1	31.79	88.48	0.927
MLPMixer	38.2	14.7	29.52	83.36	0.934
ConvMixer	**3.9**	**5.5**	32.09	88.93	0.926
IncepU	58.0	19.4	32.22	89.19	0.927
gSTA	46.8	16.5	**26.60**	**77.32**	**0.940**

2) The qualitative analysis of modules: 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. 11. 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 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.

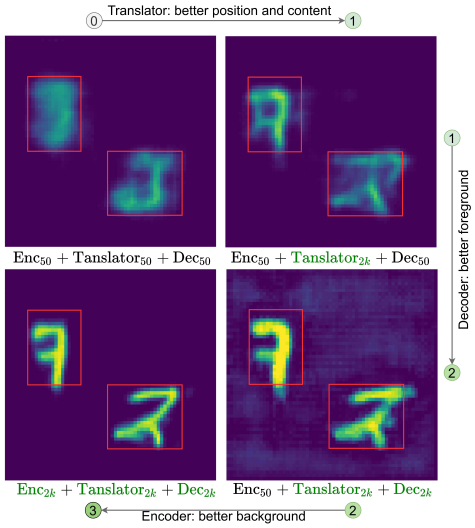


Fig. 11. The role of the Translator, Spatial Encoder and Decoder.

VI. CONCLUSION

In this paper, we introduce SimVPv2, aimed at pushing the boundaries of existing spatiotemporal predictive learning models. Building on the success of SimVP, which demonstrated the feasibility of using a fully convolutional architecture to enable parallel processing and reduce dependence on complex recurrent structures, SimVPv2 goes a step further by completely removing Unet-like multi-scale architectures. Instead, it incorporates an efficient gSTA mechanism, allowing SimVPv2 to achieve state-of-the-art performance with a more streamlined and computationally efficient design. Through extensive experiments on the synthetic moving digits, traffic flow forecasting, climate prediction, road driving, human motion prediction, and robo action planning, we demonstrate the superior performance of SimVPv2 under various settings like standard spatiotemporal predictive learning, generalization across similar scenarios, prediction with flexible lengths, and multi-domain evaluation. We believe SimVPv2 establishes a strong baseline that will benefit future research in spatiotemporal predictive learning.

ACKNOWLEDGMENT

This work was supported by National Science and Technology Major Project (No. 2022ZD0115101), National Natural Science Foundation of China Project (No. 624B2115, No. U21A20427), Project (No. WU2022A009) from the Center of Synthetic Biology and Integrated Bioengineering of Westlake University and Integrated Bioengineering of Westlake University and Project (No. WU2023C019) from the Westlake University Industries of the Future Research Funding.

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. Zhang, T. Zhang, C. Wu, and R. Tao, "Multi-scale spatiotemporal feature fusion network for video saliency prediction," *IEEE Transactions on Multimedia*, 2023.

[6] Y. Yu, X. Zhao, R. Ni, S. Yang, Y. Zhao, and A. C. Kot, "Augmented multi-scale spatiotemporal inconsistency magnifier for generalized deepfake detection," *IEEE Transactions on Multimedia*, vol. 25, pp. 8487–8498, 2023.

[7] J. Liu, Z. Fan, Z. Yang, Y. Su, and X. Yang, "Multi-stage spatio-temporal fusion network for fast and accurate video bit-depth enhancement," *IEEE Transactions on Multimedia*, 2023.

[8] P. Li, C. Zhang, and X. Xu, "Fast fourier inception networks for occluded video prediction," *IEEE Transactions on Multimedia*, 2023.

[9] W. Wen, W. Ren, Y. Shi, Y. Nie, J. Zhang, and X. Cao, "Video super-resolution via a spatio-temporal alignment network," *IEEE Transactions on Image Processing*, vol. 31, pp. 1761–1773, 2022.

[10] L. Yu, Y. Cheng, K. Sohn, J. Lezama, H. Zhang, H. Chang, A. G. Hauptmann, M.-H. Yang, Y. Hao, I. Essa *et al.*, "Magvit: Masked generative video transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10 459–10 469.

[11] J. Wang, Y. Jiang, Z. Yuan, B. Peng, Z. Wu, and Y.-G. Jiang, "Omnitokenizer: A joint image-video tokenizer for visual generation," *arXiv preprint arXiv:2406.09399*, 2024.

[12] A. Gupta, L. Yu, K. Sohn, X. Gu, M. Hahn, L. Fei-Fei, I. Essa, L. Jiang, and J. Lezama, "Photorealistic video generation with diffusion models," *arXiv preprint arXiv:2312.06662*, 2023.

[13] 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.

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

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

[16] 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.

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

[18] 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.

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

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

[21] 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.

[22] 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.

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

[24] 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.

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

[26] 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.

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

[28] 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.

[29] 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.

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

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

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

[39] 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.

[40] 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.

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

[42] 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.

[43] 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.

[44] 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.

[45] S. Tang, C. Li, P. Zhang, and R. Tang, "Swinlstm: Improving spatiotemporal prediction accuracy using swin transformer and lstm," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 13 470–13 479.

[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] Z. Chang, X. Zhang, S. Wang, S. Ma, and W. Gao, "Stam: A spatiotemporal attention based memory for video prediction," *IEEE Transactions on Multimedia*, vol. 25, pp. 2354–2367, 2022.

[50] ——, "Strpm: A spatiotemporal residual predictive model for high-resolution video prediction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 13 946–13 955.

[51] Y. Tang, P. Dong, Z. Tang, X. Chu, and J. Liang, "Vmrnn: Integrating vision mamba and lstm for efficient and accurate spatiotemporal forecasting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5663–5673.

[52] O. Shouno, "Photo-realistic video prediction on natural videos of largely changing frames," *arXiv preprint arXiv:2003.08635*, 2020.

[53] 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.

[54] X. Hu, Z. Huang, A. Huang, J. Xu, and S. Zhou, "A dynamic multi-scale voxel flow network for video prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6121–6131.

[55] Y. Zhong, L. Liang, I. Zharkov, and U. Neumann, "Mmvp: Motion-matrix-based video prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4273–4283.

[56] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, "Video transformer network," *arXiv preprint arXiv:2102.00719*, 2021.

[64] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" *arXiv preprint arXiv:2102.05095*, 2021.

[65] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," *arXiv preprint arXiv:2103.15691*, 2021.

[66] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, "Multiscale vision transformers," *arXiv preprint arXiv:2104.11227*, 2021.

[67] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," *arXiv preprint arXiv:2106.13230*, 2021.

[68] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," in *Forty-first International Conference on Machine Learning*.

[69] 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.

[70] 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.

[71] 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.

[72] 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.

[73] 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.

[74] 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.

[75] 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.

[76] 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.

[77] 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.

[78] 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.

[79] 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.

[80] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn, "Robonet: Large-scale multi-robot learning," *arXiv preprint arXiv:1910.11215*, 2019.

[81] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du *et al.*, "Bridgedata v2: A dataset for robot learning at scale," in *Conference on Robot Learning*. PMLR, 2023, pp. 1723–1736.

[82] Z. Wang, Z. Lu, D. Huang, T. He, X. Liu, W. Ouyang, and L. Bai, "Predbench: Benchmarking spatio-temporal prediction across diverse disciplines," *arXiv preprint arXiv:2407.08418*, 2024.

[83] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[84] 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.

[85] 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.

[86] F. Research, "fvcore," https://github.com/facebookresearch/fvcore, 2021.

[87] 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.

[88] 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.

[89] 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.

[90] 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.

[91] 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.

[92] 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.

[93] 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.

[94] W. Lotter, G. Kreiman, and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," in *International Conference on Learning Representations*, 2016.

[95] 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.

[96] 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.

[97] 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.

[98] 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.

[99] 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.

[100] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic adversarial video prediction," *arXiv preprint arXiv:1804.01523*, 2018.

[101] 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.

[102] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," *Advances in neural information processing systems*, vol. 29, 2016.

[103] 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.

[104] J. Lee, J. Lee, S. Lee, and S. Yoon, "Mutual suppression network for video prediction using disentangled features," *arXiv preprint arXiv:1804.04810*, 2018.

[105] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, and S. Yan, "Metaformer is actually what you need for vision," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 819–10 829.
[106] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Advances in neural information processing systems*, vol. 34, pp. 24 261–24 272, 2021.
[107] A. Trockman and J. Z. Kolter, "Patches are all you need?" *arXiv preprint arXiv:2201.09792*, 2022.