SimVP: Simpler yet Better Video Prediction

Zhangyang Gao[1,2,*], Cheng Tan[1,2,*], Lirong Wu[1,2], Stan Z. Li[1,2]

[1] AI Lab, School of Engineering, Westlake University

[2] Institute of Advanced Technology, Westlake Institute for Advanced Study

[*]{gaozhangyang,tancheng,wulirong,stan.zq.li}@westlake.edu.cn

Abstract

From CNN, RNN, to ViT, we have witnessed remarkable advancements in video prediction, incorporating auxiliary inputs, elaborate neural architectures, and sophisticated training strategies. We admire these progresses but are confused about the necessity: is there a simple method that can perform comparably well? This paper proposes SimVP, a simple video prediction model that is completely built upon CNN and trained by MSE loss in an end-to-end fashion. Without introducing any additional tricks and complicated strategies, we can achieve state-of-the-art performance on five benchmark datasets. Through extended experiments, we demonstrate that SimVP has strong generalization and extensibility on real-world datasets. The significant reduction of training cost makes it easier to scale to complex scenarios. We believe SimVP can serve as a solid baseline to stimulate the further development of video prediction.

1. Introduction

A wise person can foresee the future, and so should an intelligent vision model do. Due to spatio-temporal information implying the inner laws of the chaotic world, video prediction has recently attracted lots of attention in climate change [71], human motion forecasting [3], traffic flow prediction [69] and representation learning [55]. Struggling with the inherent complexity and randomness of video, lots of interesting works have appeared in the past years. These methods achieve impressive performance gain by introducing novel neural operators like various RNNs [65–69,71] or transformers [48,70], delicate architectures like autoregressive [21,28,49,55,71] or normalizing flow [73], and applying distinct training strategies such as adversarial training [1,8,39,40,51,52,58,64]. However, there is relatively little understanding of their necessity for good performance since many methods use different metrics and datasets. Moreover, the increasing model complexity further aggravates this dilemma. A question arises: can we develop a simpler

model to provide better understanding and performance?

Deep video prediction has made incredible progress in the last few years. We divide primary methods into four categories in Figure. 1, i.e., (1) RNN-RNN-RNN (2) CNN-RNN-CNN (3) CNN-ViT-CNN, and (4) CNN-CNN-CNN. Some representative works are collected in Table. 1, from which we observe that RNN models have been favored since 2014.
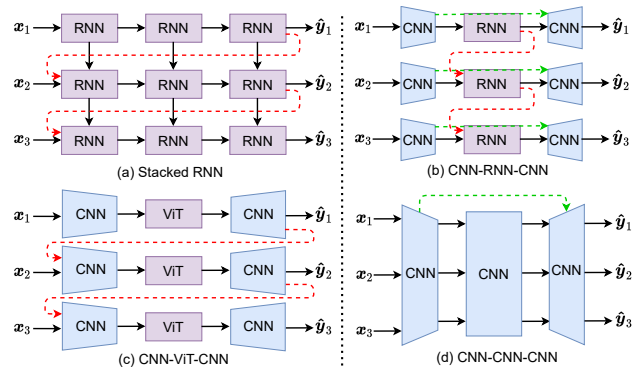

Figure 1. Different architectures for video prediction. Red and blue lines help to learn the temporal evolution and spatial dependency. SimVP belongs to the framework of CNN-CNN-CNN, which can outperform other state-of-the-art methods.

In this context, lots of novel RNNs are proposed. ConvLSTM [71] extends fully connected LSTMs to have convolutional structures for capturing spatio-temporal correlations. PredRNN [67] suggests simultaneously extracting and memorizing spatial and temporal representations. MIM-LSTM [69] applies a self-renewed memory module to model both non-stationary and stationary properties. E3D-LSTM [66] integrates 3D convolutions into RNNs. PhyCell [18] learns the partial differential equations dynamics in the latent space.

Recently, vision transformers (ViT) have gained tremendous popularity. AViT [70] merges ViT into the autoregressive framework, where the overall video is divided into volumes, and self-attention is performed within each block independently. Latent AViT [48] uses VQ-VAE [44] to com-

---

[*]Equal contribution

	RNN-RNN-RNN	CNN-RNN-CNN	CNN-ViT-CNN	CNN-CNN-CNN
2014-2015	[41, 55, 71]	[36, 46]	-	[40]
2016-2017	[38, 47, 63, 67]	[3, 9, 15, 32, 62]	-	[20, 35, 59]
2018-2019	[22, 43, 56, 65, 69, 74]	[6, 10, 61, 66, 73]	[70]	[16, 29, 72]
2020-2021	[68]	[18, 23]	[48]	[7, 54]

Table 1. Some representative video prediction works since 2014.

press the input images and apply AViT in the latent space to predict future frames.

In contrast, purely CNN-based models are not as favored as the approaches mentioned above, and fancy techniques are usually required to improve the novelty and performance, e.g., adversarial training [29], teacher-student distilling [7], and optical flow [16]. We admire their significant advancements but expect to exploit how far a simple model can go. In other words, we have made much progress against the baseline results, but have the baseline results been underestimated?

We aim to provide a simpler yet better video prediction model, namely SimVP. This model is fully based on CNN and trained by the MSE loss end-to-end. Without introducing any additional tricks and complex strategies, SimVP can achieve state-of-the-art performance on five benchmark datasets. The simplicity makes it easy to understand and use as a common baseline. The better performance provides a solid foundation for further improvements. We hope this study will shed light on future research.

2. Background

Problem statement Video prediction aims to infer the future frames using the previous ones. Given a video sequence $\boldsymbol{X}_{t,T} = \{\boldsymbol{x}_i\}_{t-T+1}^{t}$ at time t with the past T frames, our goal is to predict the future sequence $\boldsymbol{Y}_{t,T'} = \{\boldsymbol{x}_i\}_{t}^{t+T'}$ at time t that contains the next T' frames, where $\boldsymbol{x}_i \in \mathbb{R}^{C,H,W}$ is an image with channels C, height H, and width W. Formally, the predicting model is a mapping $\mathcal{F}_\Theta : \boldsymbol{X}_{t,T} \mapsto \boldsymbol{Y}_{t,T'}$ with learnable parameters $\boldsymbol{\Theta}$, optimized by:

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

where $\mathcal{L}$ can be various loss functions, and we simply employ MSE loss in our setting.

RNN-RNN-RNN As shown in Figure. 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 [41] stacks multiple gated autoencoders in a recurrent pyramid structure. ConvLSTM [71] extends fully connected LSTMs to have convolutional computing structures to capture spatio-temporal

correlations. PredRNN [67] suggests simultaneously extracting and memorizing spatial and temporal representations. PredRNN++ [65] proposes gradient highway unit to alleviate the gradient propagation difficulties for capturing long-term dependency. MIM-LSTM [69] uses a self-renewed memory module to model both the non-stationary and stationary properties of the video. dGRU [43] 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 video prediction.

CNN-RNN-CNN This framework projects video frames to the latent space and employs RNN to predict the future latent states, seeing Figure. 1 (b). In general, they focus on modifying the LSTM and encoding-decoding modules. Spatio-Temporal video autoencoder [46] incorporates ConvLSTM and an optical flow predictor to capture changes over time. Conditional VRNN [6] combines CNN encoder and RNN decoder in a variational generating framework. E3D-LSTM [66] applies 3D convolution for encoding and decoding and integrates it in latent RNNs for obtaining motion-aware and short-term features. CrevNet [73] proposes using CNN-based normalizing flow modules to encode and decode inputs for information-preserving feature transformations. PhyDNet [18] 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.

CNN-ViT-CNN This framework introduces Vision Transformer (ViT) to model latent video dynamics. By extending language transformer [60] to ViT [12], a wave of research has been sparked recently. As to image transformer, DeiT [57] and Swin Transformer [33] have achieved state-of-the-art performance on various vision tasks. The great success of image transformer has inspired the investigation of video transformer. VTN [42] applies sliding window attention on temporal dimension following a 2D spatial feature extractor. TimeSformer and ViViT [2, 4] study different space-time attention strategies and suggest that separately applying temporal and spatial attention can achieve superb performance. MViT [14] extracts multiscale
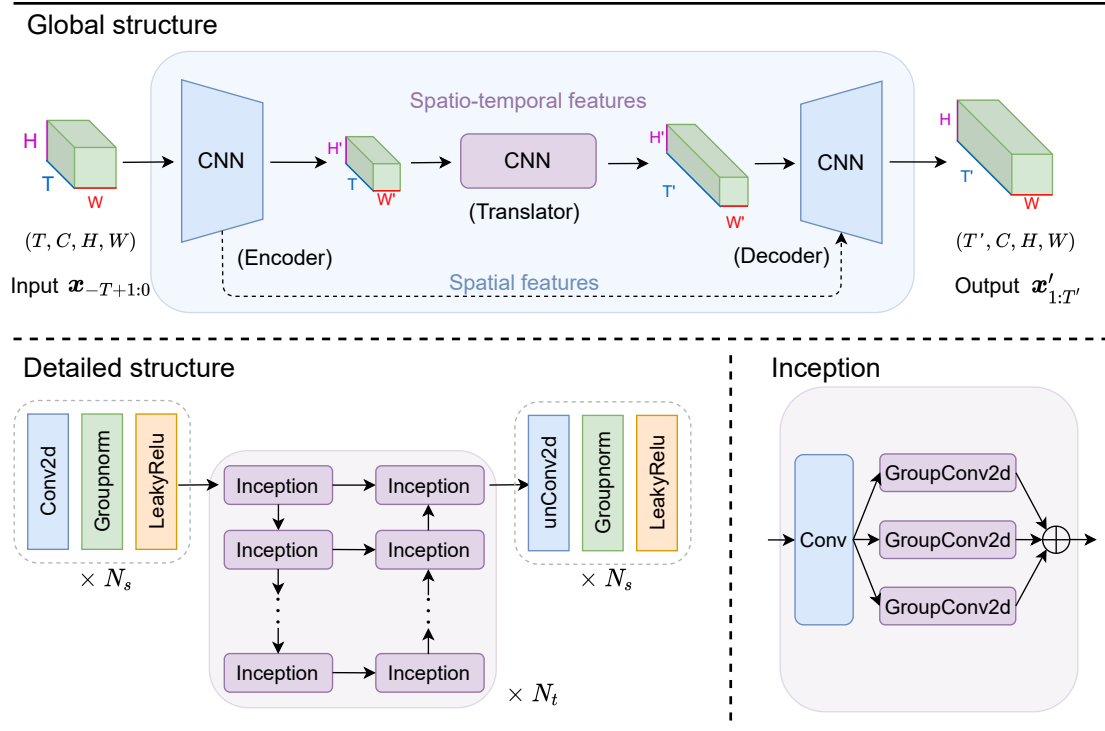
Global structure

Spatio-temporal features

CNN — (Translator)

(Encoder) Spatial features (Decoder)

(T, C, H, W) (T', C, H, W)

Input $\boldsymbol{x}_{-T+1:0}$ Output $\boldsymbol{x}'_{1:T'}$

Detailed structure **Inception**

Figure 2. The overall framework of SimVP. Both the Encoder, Translator, and Decoder are built upon CNN. The encoder stacks N_s ConvNormReLU block to extract spatial features, i.e., convoluting C channels on (H, W). The translator employs N_t Inception modules to learn temporal evolution, i.e., convoluting $T \times C$ channels on (H, W). The decoder utilizes N_s unConvNormReLU blocks to reconstruct the ground truth frames, which convolutes C channels on (H, W).

pyramid features to provide state-of-the-art results on SSv2. Video Swin Transformer [34] expands Swin Transformer from 2D to 3D, where the shiftable local attention schema leads to a better speed-accuracy trade-off. Most of the models above are designed for video classification; works about video prediction [48, 70] using ViT are still limited. More related works may emerge in the future.

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. DVF [35] suggests learning the voxel flow by CNN autoencoder to reconstruct a frame by borrowing voxels from nearby frames. PredCNN [72] combines cascade multiplicative units (CMU) with CNN to capture inter-frame dependencies. DPG [16] disentangles motion and background via a flow predictor and a context generator. [7] encodes RGB frames from the past and decodes the future semantic segmentation by using CNN and teacher-student distilling. [54] 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 made progress, we are curious about what happens if the complexity is reduced. Is there a solution that is are much simpler but can exceed or match the perfor-

mance of state-of-the-art methods?

Motivation We have witnessed many terrific methods that have achieved 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. This work does not propose new modules. Instead, we aim to build a simple network based on existing CNNs and see how far the simple model can go in video prediction.

3. SimVP

Our model, dubbed *SimVP*, consists of an encoder, a translator and a decoder built on CNN, seeing Figure. 2. The encoder is used to extract spatial features, the translator learns temporal evolution, and the decoder integrates spatio-temporal information to predict future frames.

Encoder The encoder stacks N_s ConvNormReLU blocks (Conv2d+LayerNorm+LeakyReLU) to extract spatial features, i.e., convoluting C channels on (H, W). The hidden feature is:

$$z_i = \sigma(\text{LayerNorm}(\text{Conv2d}(z_{i-1}))), 1 \le i \le N_s \quad (2)$$

where the input z_{i-1} and output z_i shapes are (T, C, H, W) and $(T, \hat{C}, \hat{H}, \hat{W})$, respectively.

Translator The translator employs N_t Inception modules to learn temporal evolution, i.e., convoluting $T \times C$ channels on (H, W). The Inception module consists of a bottleneck Conv2d with 1×1 kernel followed by parallel GroupConv2d operators. The hidden feature is:

$$z_j = \text{Inception}(z_{j-1}), N_s < j \leq N_s + N_t \qquad (3)$$

where the inputs z_{j-1} and output z_j shapes are $(T \times C, H, W)$ and $(\hat{T} \times \hat{C}, H, W)$.

Decoder The decoder utilizes N_s unConvNormReLU blocks (ConvTranspose2d+GroupNorm+LeakyReLU) to reconstruct the ground truth frames, which convolutes C channels on (H, W). The hidden feature is:

$$z_k = \sigma(\text{GroupNorm}(\text{unConv2d}(z_{k-1}))), \\ N_s + N_t < k \leq 2N_s + N_t \qquad (4)$$

where the shapes of input z_{k-1} and output z_k are $(T, \hat{C}, \hat{H}, \hat{W})$ and (T, C, H, W), respectively. We use ConvTranspose2d [13] to serve as the unConv2d operator.

Summary SimVP does not use advanced modules such as RNN, LSTM and Transformer, nor introduce complex training strategies such as adversarial training and curriculumn learning. All the things we need are CNN, shortcuts and vanilla MSE loss.

4. Experiments

Metrics We employ MSE, MAE, Structural Similarity Index Measure (SSIM), and Peak Signal to Noise Ratio (PSNR) to evaluate the quality of predictions, following [18, 45, 73]. We also report the running time per epoch and the memory footprint per sample on a single NVIDIA-V100 to provide a comprehensive view for future research. [1]

Datasets We conduct experiments on five datasets for evaluation. The statistics are summarized in Table. 2, including the number of training samples n_{train}, number of testing samples n_{test}, image resolution (C, H, W), input sequence length T and forecasting sequence length T'. The detailed dataset description can be found in the appendix.

4.1. How far can SimVP go?

We pursue the simple but effective model. The simplicity has been described in Section. 3. The effectiveness will be verified through exploratory experiments in this section.

---

[1] The realistic running time is more reliable to FLOPs or MACs of the model, e.g., when RNN and CNN have the same FLOPs, RNN takes much more time due to its recurrent computation.

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

	N_{train}	N_{test}	(C, H, W)	T	T'
MMNIST	10000	10000	(1, 64, 64)	10	10
TrafficBJ	19627	1334	(2, 32, 32)	4	4
Human3.6	2624	1135	(3, 128, 128)	4	4
Caltech Pedestrian	2042	1983	(3, 128, 160)	10	1
KTH	5200	3167	(1, 128, 128)	10	20 or 40

Challenge We aim to provide a comprehensive and rigorous view of video prediction methods. However, three challenges stand in our way:

- Various methods may adopt disparate metrics.
- These methods apply experiments on different datasets with distinct protocols.
- They use different code frameworks and unique tricks, making it difficult to compare fairly.

Solution To overcome aforemesioned challenges, we choose the common used dataset (Moving MNIST) and metrics (MSE and SSIM) to evaluate recent important researches. We directly report the best metrics according to the original papers, avoiding the risk of performance degradation caused by our reproduction. Any method that does not use the same dataset, metrics, or protocol will be neglected here. For convenience, we provide the publication status and Github links of these method in Table. 3.

Results and Discovery As shown in Table. 3, SimVP achieves state-of-the-art MSE and SSIM on Moving MNIST. We observe that SimVP, PhyDNet, and CrevNet significantly outperform previous methods, with MSE reduction up to 42%. However, SimVP is much simpler than PhyDNet and CrevNet, without using RNN, LSTM, or complicated modules, which are considered as the important reason for performance improvement. Through these explorations, we are excited to find that **it is promising to achieve better performance with a extremely simple model.** Perhaps previous works pay too much attention to the model complexity and novelty, and it's time to go back to basics because a simpler model makes things clearer.

Simplicity leads to efficiency. Another benefit that comes from simplicity is good computational efficiency. In Table. 4, we compare GPU memory (per sample), FLOPs (per image) and training time of SOTA methods on Moving MNIST. As CNN has good computational optimization and avoids iterative calculation, the training process of SimVP is much faster than others, which means that SimVP can be used and extended more easily.

Table 3. Performance comparison of various methods on Moving MNIST. The source code of SimVP will be released soon.

Method	Conference	MSE	SSIM	Github
ConvLSTM [71]	(NIPS 2015)	103.3	0.707	PyTorch
PredRNN [67]	(NIPS 2017)	56.8	0.867	PyTorch
PredRNN-V2 [68]	(Arxiv 2021)	48.4	0.891	PyTorch
CausalLSTM [65]	(ICML 2018)	46.5	0.898	Tensorflow
MIM [69]	(CVPR 2019)	44.2	0.910	Tensorflow
E3D-LSTM [66]	(ICLR 2018)	41.3	0.920	Tensorflow
PhyDNet [18]	(CVPR 2020)	24.4	0.947	PyTorch
CrevNet [73]	(ICLR 2020)	22.3	0.949	PyTorch
SimVP	–	23.8	0.948	PyTorch

Table 4. Computation comparision on Moving MNIST. We report the per sample memory overhead, per frame FLOPs, and total training time. For methods marked with ∗, we report the results reproduced by their official codes. Other results refer to [73].

Method	Memory	FLOPs	Training time
ConvLSTM	1043MB	107.4G	–
PredRNN	1666 MB	192.9 G	–
CausalLSTM	2017 MB	106.8 G	–
E3D-LSTM	2695 MB	381.3 G	–
CrevNet ∗	224 MB	1.652 G	$\approx 10d$ (300k iters)
PhyDNet ∗	200 MB	1.633 G	$\approx 10d$ (2k epochs)
SimVP ∗	412 MB	1.676 G	$\approx 2d$ (2k epochs)

4.2. Translator: should we use RNN, Transformer or CNN?

With the rapid development of novel temporal modules based on RNN, Transformer, and CNN, researchers may be dazzled and confused about which one to choose for video modeling. In SimVP, the Translator module is responsible for learning temporal evolution. We replace the CNN-based Translator with the most representative RNNs and Transformers to reveal that which time module is more suitable for video modeling under the SimVP framework.

Translator selection The CNN-based Translator serves as the baseline, since it is quite simple yet effective. We hope to find new modules that is promising to outperform the Translator to inspire subsequent researches. For RNN, we choose currently state-of-the-art **PhyDNet** [18] and **CrevNet** [73]. As suggested in PhyDNet, we use the PhyCell, a novel time module considering physical dynamics, for temporal modeling. As to CrevNet, we use normalizing flow autoencoder + ST-LSTM [67] as the translator. For Transformers, we chose recent influential work such as Video Swin Transformer [34] and Latent Video Transformer [48, 70]. To make these modules workable under the SimVP framework, we may modify a few implementation details without changing the core algorithm, such

as removing the autoregressive generation scheme and re-implementing position encoding.

Setting We compare five translators on Moving MNIST and Human, keeping the encoder and decoder the same. We adjust the hyperparameters of these translators to make them work with similar GPU memory footprints. By default, we use batch size 16, epoch 100, and Adam optimizer. The number of encoder and decoder layers is 4. We choose the largest learning rate from $\{1e^{-2}, 1e^{-3}, 1e^{-4}\}$ under the premise of stable training.
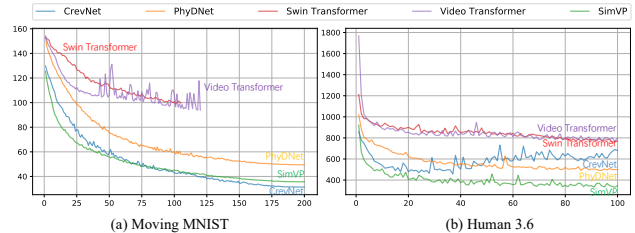

(a) Moving MNIST   (b) Human 3.6

Figure 3. Training dynamics of translators on Moving MNIST and Human3.6. The x-axis is the training epochs and the y-axis is the evaluated MSE. SimVP converges faster than other methods in the early training phase, but CrevNet performs better in the long run.

Results and Discoveries In Figure. 3, we show the training dynamics using various translators on Moving MNIST and Human3.6, respectively. CrevNet looks like a good solution because the inherent simplicity of Moving MNIST, while our SimVP is suboptimal. However, this does not always hold true when it comes to real-world datasets. For example, on Human dataset (Figure. 3(b)), SimVP significantly outperforms other methods. As to the training stability, CNN performs better than RNN (Figure. 7, appendix), since it will not fluctuate violently under large learning rate. In short, our discoveries are:

1. CNN and RNN achieves state-of-the-art performance under limited computation costs.
2. RNN converges faster than others in the long run (Figure. 3) if the model capacity is sufficient.
3. CNN training is more robust and does not fluctuate dramatically at large learning rates (Figure. 7).
4. Transformer has no advantage in our SimVP framework under the similar resource consumption.

4.3. Does SimVP achieve SOTA on general cases?

From previous analysis, we believe that SimVP has potential to outperform recent state-of-the-art methods. In this section, we provide more experimental evidence to confirm this claim. Specifically, we aim to answer three questions:

- **Q1:** Can SimVP achieve SOTA results on other common benchmarks?

Table 5. SimVP vs SOTA. The optimal(or suboptimal) results are marked by **bold**(or <u>underlined</u>).

	Moving MNIST			TrafficBJ			Human3.6		
	MSE↓	MAE↓	SSIM↑	MSE × 100 ↓	MAE↓	SSIM↑	MSE / 10 ↓	MAE / 100 ↓	SSIM↑
ConvLSTM	103.3	182.9	0.707	48.5	17.7	0.978	50.4	18.9	0.776
PredRNN	56.8	126.1	0.867	46.4	17.1	0.971	48.4	18.9	0.781
CausalLSTM	46.5	106.8	0.898	44.8	16.9	0.977	45.8	17.2	0.851
MIM	44.2	101.1	0.910	42.9	16.6	0.971	42.9	17.8	0.790
E3D-LSTM	41.3	86.4	0.910	43.2	16.9	0.979	46.4	16.6	0.869
PhyDNet	<u>24.4</u>	<u>70.3</u>	<u>0.947</u>	<u>41.9</u>	**16.2**	**0.982**	<u>36.9</u>	<u>16.2</u>	<u>0.901</u>
SimVP	**23.8**	**68.9**	**0.948**	**41.4**	**16.2**	**0.982**	**31.6**	**15.1**	**0.904**

- **Q2:** Does SimVP generalize well across different datasets?
- **Q3:** Does SimVP extend well to the case of flexible predictive length?

Setting of Q1 Reporting the widely used metrics on standard benchmark datasets is the key to advancing the research progress. We evaluate SimVP on three common used benchmarks, i.e., Moving MNIST, TrafficBJ and Human3.6. Moving MNIST [55] consists of two digits independently moving within the 64×64 grid and bounced off the boundary. By assigning different initial locations and velocities to each digit, we can get an infinite number of sequences, and predict the future 10 frames from previous 10 frames. TrafficBJ contains the trajectory data in Beijing collected from taxicab GPS with two channels, i.e. inflow or outflow defined in [75]. Following [69], 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 [18,69]. Models are trained to predict 4 next frames by observing prioring 4 frames. Human3.6 [24] is a complex human pose dataset with 3.6 million samples, recording different activities. Similar to [18, 55, 69], only videos with "walking" scenario are used, and 4 future frames are generated by feeding previous 4 RGB frames. Following [18], we report the MSE, MAE and SSIM of SimVP in Table. 5. Six state-of-the-art RNN baselines are chosen for comparison, including ConvLSTM [71], PredRNN [67], CausalLSTM [65], MIM [69], E3D-LSTM [66] and PhyDNet [18]. We train SimVP on Moving MNIST, Human3.6, and TrafficBJ for 2k, 100 and 80 epochs, respectively. We use Adam optimizer and the learning rate is 0.01.

Answer of Q1 SimVP can achieve SOTA results on lightweight benchmarks. From Table. 5, we observe that SimVP outperform all RNN baselines in all settings. The improvements on Moving MNIST and TrafficBJ are modest; On Human3.6, the relative improvement is significant. In addition, SimVP takes training time compared to previous PhyDNet. Both performance and computing advantages support our answer.
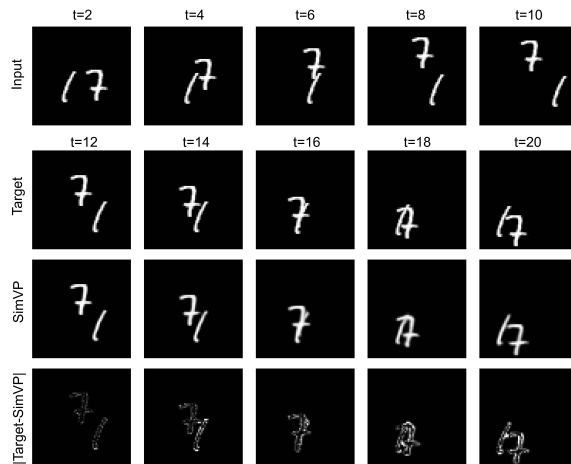

Figure 4. Visualization of Moving MNIST.

Setting of Q2 Generalizing the knowledge across different datasets, especially under the unsupervised setting, is the core research point of machine learning. To investigate the generalization ability of SimVP, we train the model for 50 epochs on KITTI and evaluate it on Caltech Pedestrian. KITTI [17] 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 [11] 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 KITTI dataset to predict the next frame after 10-frame warm-up and are evaluated on Caltech Pedestrian. Compared with the previous experiments, carmounted camera videos dataset and the distinct training-evaluating data present another level of difficulty for video prediction as it describes various nonlinear three-dimensional dynamics of multiple moving objects including backgrounds. Following [37, 45, 73], we center-crop Sixteen baselines are selected for comparision, including BeyondMSE [40], MCnet [62], DVF [35], Dual-GAN [32],

CtrlGen [19],PredNet [37], ContextVP [5], GAN-VGG, G-VGG, G-MAE, GAN-MAE [54], SDC-Net [50], rCycle-Gan [29], DPG [16], CrevNet [73] and STMFANet [26].

Table 6. Results on Caltech Pedestrian dataset.

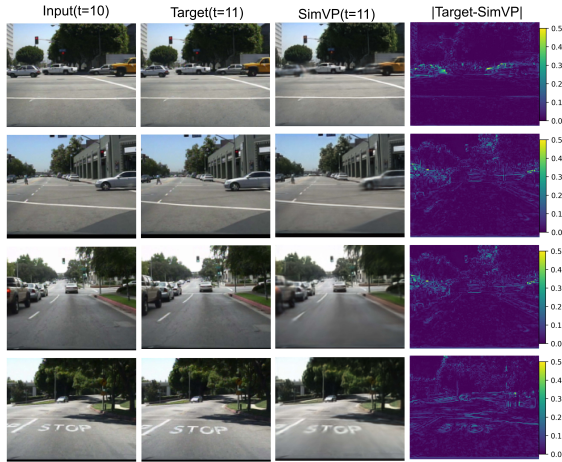| Method	Caltech Pedestrian (10 → 1)		
	MSE↓	SSIM↑	PSNR↑
BeyondMSE [40]	3.42	0.847	-
MCnet [62]	2.50	0.879	-
DVF [35]	-	0.897	26.2
Dual-GAN [32]	2.41	0.899	-
CtrlGen [19]	-	0.900	26.5
PredNet [37]	2.42	0.905	27.6
ContextVP [5]	1.94	0.921	28.7
GAN-VGG [54]	-	0.916	-
G-VGG [54]	-	0.917	-
SDC-Net [50]	1.62	0.918	-
rCycleGan [29]	_1.61_	0.919	29.2
DPG [16]	-	0.923	28.2
G-MAE [54]	-	0.923	-
GAN-MAE [54]	-	0.923	-
CrevNet [73]	-	0.925	_29.3_
STMFANet [26]	-	_0.927_	29.1
SimVP (ours)	**1.56**	**0.940**	**33.1**

Figure 5. Visualization of Caltech (10 → 1).

Answer of Q2 SimVP generalize well cross different datasets, seeing Figure. 5. Although there is room to improve the clarity of the generated objects, the evaluated metrics have exceeded previous methods. As shown in Table. 6, the MSE is reduced by 3.1%, the SSIM is improved by 1.4%, and the PSNR is improved by 13.0%. The training process can be finished within 4h.

Setting of Q3 A possible limitation of CNN-based methods is that it may be diffifcult to scale to prediction with flexible length. We handle this problem by imitating RNN,

that is taking previous predictions as recent inputs to recursively produce long-term predictions. Following [45, 66], we compare the PSNR and SSIM of SimVP with other baselines on KTH, seeing Table.8, where we train SimVP for 100 epochs. The KTH dataset [53] contains 25 individuals performing 6 types of actions, i.e., walking, jogging, running, boxing, hand waving and hand clapping. Following [62, 66], we use person 1-16 for training and 17-25 for testing. Models are trained to predict next 20 or 40 frames from the previous 10 observations. Sixteen baselines are included, such as MCnet [62], ConvLSTM [71], SAVP, SAVP-VAE [30], VPN [28], DFN [25], fRNN [43], Znet [74], SV2P [3], PredRNN [67], VarNet [27], PredRNN++ [65], MSNET [31], E3d-LSTM [66] and STMFANet [26].

Table 8. Results on KTH dataset.

Method	KTH (10 → 20)		KTH (10 → 40)	
	SSIM↑	PSNR↑	SSIM↑	PSNR↑
MCnet [62]	0.804	25.95	0.73	23.89
ConvLSTM [71]	0.712	23.58	0.639	22.85
SAVP [30]	0.746	25.38	0.701	23.97
VPN [28]	0.746	23.76	–	–
DFN [25]	0.794	27.26	0.652	23.01
fRNN [43]	0.771	26.12	0.678	23.77
Znet [74]	0.817	27.58	–	–
SV2Pi [3]	0.826	27.56	0.778	25.92
SV2Pv [3]	0.838	27.79	0.789	26.12
PredRNN [67]	0.839	27.55	0.703	24.16
VarNet [27]	0.843	28.48	0.739	25.37
SVAP-VAE [30]	0.852	27.77	0.811	26.18
PredRNN++ [65]	0.865	28.47	0.741	25.21
MSNET [31]	0.876	27.08	–	–
E3d-LSTM [66]	0.879	29.31	0.810	27.24
STMFANet [26]	_0.893_	_29.85_	_0.851_	_27.56_
SimVP (ours)	**0.905**	**33.72**	**0.886**	**32.93**

Answer of Q3 SimVP extend well to the case of flexible predictive length. From Table. 8, we know that SimVP achieve state-of-the-art performance. Notably, the PSNR improves by 11.8% and 19.5% in both (10 → 20) and (10 → 40) settings. This phenomenon further indicates that the performance degradation of SimVP is less than others on long-term prediction tasks.

5. Ablation study

While SimVP is quite simple, we believe there still exists unapprehended parts. We are eager to know:

- **Q1:** Which architectural design plays the key role in improving performance?
- **Q2:** How the Conv kernel influence the performance?
- **Q3:** What roles do the Enc, Translator, and Dec play?

		model 1	model 2	model 3	model 4	model 5	model 6	model 7	model 8	model 9	SimVP
Archi	S-UNet	–	✓	✓	✓	✓	✓	✓	✓	✓	✓
	T-UNet	✓	–	✓	✓	✓	✓	✓	✓	✓	✓
	#Groups	4	4	1	4	4	4	4	4	4	4
	Norm	G	G	G	B	G	G	G	G	G	G
Kernel	$(3)+c_t$	–	–	–	–	✓	–	–	–	–	–
	$(5)+c_t$	–	–	–	–	–	✓	–	–	–	–
	$(7)+c_t$	–	–	–	–	–	–	✓	–	–	–
	$(11)+c_t$	–	–	–	–	–	–	–	✓	–	–
	$(11)+2c_t$	–	–	–	–	–	–	–	–	✓	–
	$(3,5,7,11)+c_t$	✓	✓	✓	✓	–	–	–	–	–	✓
MSE	Moving MNIST	41.7	41.5	44.8	41.0	58.9	51.1	49.1	46.3	44.8	41.7
	TrafficBJ ($\times100$)	43.5	41.5	44.5	44.3	43.5	44.0	43.3	42.3	42.3	42.0
	Human3.6 ($/10$)	32.4	33.4	33.0	34.8	37.3	34.0	32.9	33.4	32.1	32.0
	Summary	↓	↓	↓↓↓	↓	↓↓↓	↓↓↓	↓↓↓	↓↓	↓	–

Table 7. Ablation study. S-UNet or T-UNet denotes the shortcut connection in the spatial or temporal encoder-decoder. #Groups is the number of convolutional groups. G and B indicate group normalization and batch normalization. $(3,5,7,11)+c_t$ means the Conv kernels of the Inception module plus the translator's hidden dimension. Note that the spatial Enc and Dec are fixed on each dataset, seeing Table. 10 (appendix) for detailed settings. Finally, we compare models with SimVP, and results with a gap of less than 0.5 are regarded as the same.

Setting of Q1 and Q2 We perform the ablation study on Moving MNIST, TrafficBJ, and Human3.6. All models are trained up to 100 epochs, different from previous settings. For neural architecture design, we study whether using spatial UNet shortcut, temporal UNet shortcut, group convolution, and group normalization can bring performance gain. As to the convolutional kernel, we study how the kernel size and hidden dimension affect the model performance. We report the MSE metric for all datasets.

Answer of Q1 All of S-UNet, T-UNet, group convolution and group normalization can bring performance gain, and the order of significance is: group convolution $>$ group normalization $\approx$ S-UNet $\approx$ T-UNet. Please see Table. 7 (from model 1 to model 4) for experimental evidence.

Answer of Q2 Larger kernel size and more model parameters lead to better performance. In Table. 7, from model 5 to model 8, with the increasing kernel size, we can see the significant performance gain. This improvement can be further enhanced by doubling the hidden dimension of model 8 to construct model 9. SimVP chooses multi-scale kernels, and the parameters of the Translator are 84% of model 9.

Setting of Q3 We represent submodules trained with n epochs as $\text{Enc}_n, \text{Translator}_n, \text{Dec}_n$, mix them and evaluate the results at $t = 20$ (the last prediction) to reveal the role of each well-tuned module.

Answer of Q3 As shown in Figure. 6, we conduct that the translator mainly focus on predict the position and content of the objects. The decoder is responsible for optimizing the shape of the foreground objects. The encoder can erase the background error by means of spatial UNet connection.
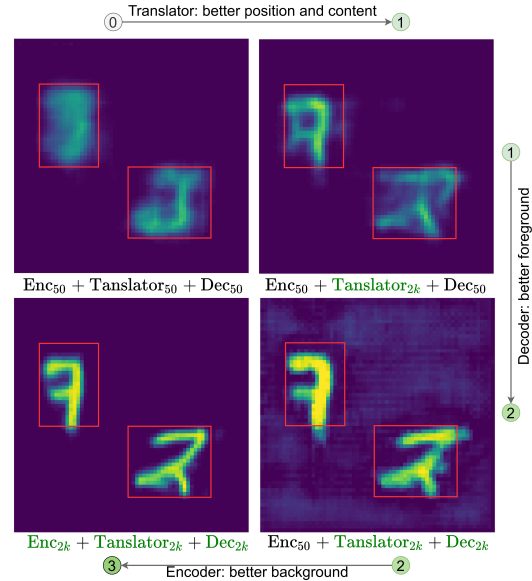

Figure 6. The role of the Translator, Encoder and Decoder.

6. Conclusion

We propose SimVP, a simpler yet effective CNN model for video prediction. We show that SimVP can achieve state-of-the-art results without introducing any complex modules, strategies and tricks. Meanwhile, the reduced computing cost makes it easy to scale up to more scenarios. We believe simpler is better, and SimVP may serve as a strong baseline and provide inspiration for future research.

References

[1] Dinesh Acharya, Zhiwu Huang, Danda Pani Paudel, and Luc Van Gool. Towards high resolution video generation with progressive growing of sliced wasserstein gans. *arXiv preprint arXiv:1810.02419*, 2018. 1

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021. 2

[3] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017. 1, 2, 7

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021. 2

[5] Wonmin Byeon, Qin Wang, Rupesh Kumar Srivastava, and Petros Koumoutsakos. Contextvp: Fully context-aware video prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 753–769, 2018. 7

[6] Lluis Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional vrnns for video prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7608–7617, 2019. 2

[7] Hsu-kuang Chiu, Ehsan Adeli, and Juan Carlos Niebles. Segmenting the future. *IEEE Robotics and Automation Letters*, 5(3):4202–4209, 2020. 2, 3

[8] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019. 1

[9] Emily Denton and Vighnesh Birodkar. Unsupervised learning of disentangled representations from video. *arXiv preprint arXiv:1705.10915*, 2017. 2

[10] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *International Conference on Machine Learning*, pages 1174–1183. PMLR, 2018. 2

[11] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, June 2009. 6, 12

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[13] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *ArXiv e-prints*, mar 2016. 4

[14] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *arXiv preprint arXiv:2104.11227*, 2021. 2

[15] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *Advances in neural information processing systems*, 29:64–72, 2016. 2

[16] Hang Gao, Huazhe Xu, Qi-Zhi Cai, Ruth Wang, Fisher Yu, and Trevor Darrell. Disentangling propagation and generation for video prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9006–9015, 2019. 2, 3, 7

[17] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 6, 12

[18] Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11474–11484, 2020. 1, 2, 4, 5, 6, 12

[19] Zekun Hao, Xun Huang, and Serge Belongie. Controllable video generation with sparse trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7854–7863, 2018. 7

[20] Mikael Henaff, Junbo Zhao, and Yann LeCun. Prediction under uncertainty with error-encoding networks. *arXiv preprint arXiv:1711.04994*, 2017. 2

[21] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019. 1

[22] Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. *arXiv preprint arXiv:1806.04166*, 2018. 2

[23] Anthony Hu, Fergal Cotter, Nikhil Mohan, Corina Gurau, and Alex Kendall. Probabilistic future prediction for video scene understanding. In *European Conference on Computer Vision*, pages 767–785. Springer, 2020. 2

[24] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013. 6, 12

[25] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *NIPS*, 2016. 7

[26] Beibei Jin, Yu Hu, Qiankun Tang, Jingyu Niu, Zhiping Shi, Yinhe Han, and Xiaowei 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*, pages 4554–4563, 2020. 7

[27] Beibei Jin, Yu Hu, Yiming Zeng, Qiankun Tang, Shice Liu, and Jing Ye. Varnet: Exploring variations for unsupervised video prediction. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5801–5806. IEEE, 2018. 7

[28] Nal Kalchbrenner, Aäron Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. In *International Conference on Machine Learning*, pages 1771–1779. PMLR, 2017. 1, 7

[29] Yong-Hoon Kwon and Min-Gyu Park. Predicting future frames using retrospective cycle gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1811–1820, 2019. 2, 7

[30] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial

video prediction. *arXiv preprint arXiv:1804.01523*, 2018. 7

[31] Jungbeom Lee, Jangho Lee, Sungmin Lee, and Sungroh Yoon. Mutual suppression network for video prediction using disentangled features. *arXiv preprint arXiv:1804.04810*, 2018. 7

[32] Xiaodan Liang, Lisa Lee, Wei Dai, and Eric P Xing. Dual motion gan for future-flow embedded video prediction. In *proceedings of the IEEE international conference on computer vision*, pages 1744–1752, 2017. 2, 6, 7

[33] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 2

[34] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021. 3, 5

[35] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4463–4471, 2017. 2, 3, 6, 7

[36] William Lotter, Gabriel Kreiman, and David Cox. Unsupervised learning of visual structure using predictive generative networks. *arXiv preprint arXiv:1511.06380*, 2015. 2

[37] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016. 6, 7, 12

[38] Chaochao Lu, Michael Hirsch, and Bernhard Scholkopf. Flexible spatio-temporal networks for video prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6523–6531, 2017. 2

[39] Pauline Luc, Aidan Clark, Sander Dieleman, Diego de Las Casas, Yotam Doron, Albin Cassirer, and Karen Simonyan. Transformation-based adversarial video prediction on large-scale data. *arXiv preprint arXiv:2003.04035*, 2020. 1

[40] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015. 1, 2, 6, 7

[41] Vincent Michalski, Roland Memisevic, and Kishore Konda. Modeling deep temporal dependencies with recurrent grammar cells"". *Advances in neural information processing systems*, 27:1925–1933, 2014. 2

[42] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021. 2

[43] Marc Oliu, Javier Selva, and Sergio Escalera. Folded recurrent neural networks for future video prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 716–731, 2018. 2, 7

[44] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*, 2017. 1

[45] Sergiu Oprea, Pablo Martinez-Gonzalez, Alberto Garcia-Garcia, John Alejandro Castro-Vargas, Sergio Orts-Escolano, Jose Garcia-Rodriguez, and Antonis Argyros. A review on deep learning techniques for video prediction.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020. 4, 6, 7

[46] Viorica Patraucean, Ankur Handa, and Roberto Cipolla. Spatio-temporal video autoencoder with differentiable memory. *arXiv preprint arXiv:1511.06309*, 2015. 2

[47] Isabeau Prémont-Schwarz, Alexander Ilin, Tele Hotloo Hao, Antti Rasmus, Rinu Boney, and Harri Valpola. Recurrent ladder networks. *arXiv preprint arXiv:1707.09219*, 2017. 2

[48] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent video transformer. *arXiv preprint arXiv:2006.10704*, 2020. 1, 2, 3, 5

[49] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014. 1

[50] Fitsum A Reda, Guilin Liu, Kevin J Shih, Robert Kirby, Jon Barker, David Tarjan, Andrew Tao, and Bryan Catanzaro. Sdc-net: Video prediction using spatially-displaced convolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 718–733, 2018. 7

[51] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE international conference on computer vision*, pages 2830–2839, 2017. 1

[52] Masaki Saito and Shunta Saito. Tganv2: Efficient training of large models for video generation with multiple subsampling layers. 2018. 1

[53] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36. IEEE, 2004. 7, 12

[54] Osamu Shouno. Photo-realistic video prediction on natural videos of largely changing frames. *arXiv preprint arXiv:2003.08635*, 2020. 2, 3, 7

[55] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852. PMLR, 2015. 1, 2, 6, 12

[56] Jiangxin Sun, Jiafeng Xie, Jian-Fang Hu, Zihang Lin, Jianhuang Lai, Wenjun Zeng, and Wei-Shi Zheng. Predicting future instance segmentation with contextual pyramid convlstms. In *Proceedings of the 27th acm international conference on multimedia*, pages 2043–2051, 2019. 2

[57] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 2

[58] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018. 1

[59] Joost Van Amersfoort, Anitha Kannan, Marc'Aurelio Ranzato, Arthur Szlam, Du Tran, and Soumith Chintala. Transformation-based models of video sequences. *arXiv preprint arXiv:1701.08435*, 2017. 2

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2

[61] Ruben Villegas, Dumitru Erhan, Honglak Lee, et al. Hierarchical long-term video prediction without supervision. In *International Conference on Machine Learning*, pages 6038–6046. PMLR, 2018. 2

[62] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017. 2, 6, 7, 12

[63] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to generate long-term future via hierarchical prediction. In *international conference on machine learning*, pages 3560–3569. PMLR, 2017. 2

[64] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. *Advances in neural information processing systems*, 29:613–621, 2016. 1

[65] Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and S Yu Philip. Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *International Conference on Machine Learning*, pages 5123–5132. PMLR, 2018. 1, 2, 5, 6, 7

[66] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *International conference on learning representations*, 2018. 1, 2, 5, 6, 7, 12

[67] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 879–888, 2017. 1, 2, 5, 6, 7

[68] Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, Philip S Yu, and Mingsheng Long. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *arXiv preprint arXiv:2103.09504*, 2021. 1, 2, 5

[69] Yunbo Wang, Jianjin Zhang, Hongyu Zhu, Mingsheng Long, Jianmin Wang, and Philip 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*, pages 9154–9162, 2019. 1, 2, 5, 6, 12

[70] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. *arXiv preprint arXiv:1906.02634*, 2019. 1, 2, 3, 5

[71] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015. 1, 2, 5, 6, 7

[72] Ziru Xu, Yunbo Wang, Mingsheng Long, Jianmin Wang, and MOE KLiss. Predcnn: Predictive learning with cascade convolutions. In *IJCAI*, pages 2940–2947, 2018. 2, 3

[73] Wei Yu, Yichao Lu, Steve Easterbrook, and Sanja Fidler. Efficient and information-preserving future frame prediction and beyond. In *International Conference on Learning Representations*, 2019. 1, 2, 4, 5, 6, 7, 12

[74] Jianjin Zhang, Yunbo Wang, Mingsheng Long, Wang Jianmin, and S Yu Philip. Z-order recurrent neural networks for video prediction. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 230–235. IEEE, 2019. 2, 7

[75] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017. 6, 12