# STLight: a Fully Convolutional Approach for Efficient Predictive Learning by Spatio-Temporal joint Processing

Andrea Alfarano[1,2*] Alberto Alfarano[3*] Linda Friso[4] Andrea Bacciu[5†]
Irene Amerini[5] Fabrizio Silvestri[5]

[1]DVS, University of Zurich [2]Max Planck Society [3]Meta [4]Google [5]Sapienza, University of Rome

andrea.alfarano@uzh.ch, albealfa@meta.com, lfriso@google.com,
bacciu@diag.uniroma1.it, amerini@diag.uniroma1.it, fsilvestri@diag.uniroma1.it

*Equal contribution †Work done prior to joining Amazon

## Abstract

*Spatio-Temporal predictive Learning is a self-supervised learning paradigm that enables models to identify spatial and temporal patterns by predicting future frames based on past frames. Traditional methods, which use recurrent neural networks to capture temporal patterns, have proven their effectiveness but come with high system complexity and computational demand. Convolutions could offer a more efficient alternative but are limited by their characteristic of treating all previous frames equally, resulting in poor temporal characterization, and by their local receptive field, limiting the capacity to capture distant correlations among frames. In this paper, we propose STLight, a novel method for spatio-temporal learning that relies solely on channel-wise and depth-wise convolutions as learnable layers. STLight overcomes the limitations of traditional convolutional approaches by rearranging spatial and temporal dimensions together, using a single convolution to mix both types of features into a comprehensive spatio-temporal patch representation. This representation is then processed in a purely convolutional framework, capable of focusing simultaneously on the interaction among near and distant patches, and subsequently allowing for efficient reconstruction of the predicted frames. Our architecture achieves state-of-the-art performance on STL benchmarks across different datasets and settings, while significantly improving computational efficiency in terms of parameters and computational FLOPs. The code is publicly available* [1].

## 1. Introduction

Spatio-Temporal predictive Learning (STL) aims to extract hidden spatial and temporal patterns by predicting future frames based on previous ones. Utilizing self-supervision, STL models decode complex correlations in sequences of raw data, removing the necessity for labor-intensive manual annotation and facilitating precise forecasts. STL finds large applications in several resource-constrained domains, such as autonomous driving to predict pedestrian and vehicle movements and prevent accidents [1, 6, 14, 21]; and in human-robot interactions to anticipate movements and enhance safety [2, 3, 16, 53]. An effective STL model should capture, process, and reconstruct spatial and temporal information from the input frames while balancing efficiency to ensure its applicability in real-world settings.

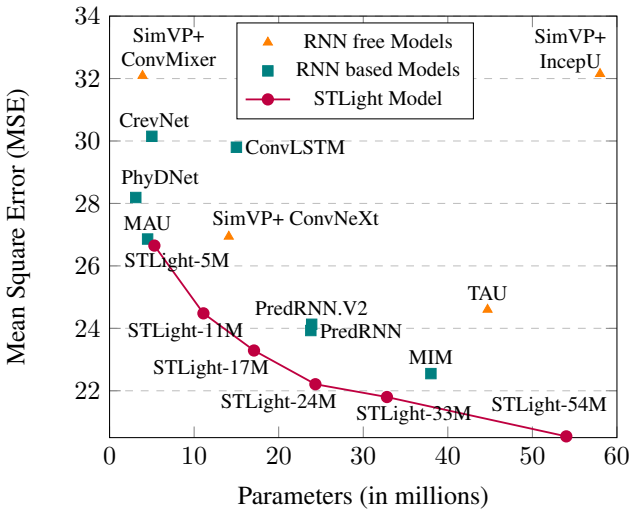


Figure 1. MSE vs Number of parameters for existing STL models and STLight on Moving MNIST dataset trained and evaluated under the same settings.

Relevant methods follow the *Spatial-Temporal-Spatial*

[1]https://github.com/AlfaranoAndrea/STLight/

framework [11, 31, 32], employing a deep Convolutional Neural Network (CNN) to individually encode the spatial details within each frame, a Recurrent Neural Network (RNN) to find and analyze temporal dynamics, and deconvolutional layers to reconstruct frames. However, the sequential nature of RNNs leads to high training and inference costs and limits STL efficiency [11, 48].

Fully convolutional architectures, which directly predict all the sequences, could be a more efficient alternative to recurrent-based models [25, 31, 48]. However, CNNs are not as effective as RNNs in capturing temporal relationships, as they tend to: (1) treat all frames uniformly, being unable to model the temporal dynamics existing in a continuous Markov process [48], and (2) focus on local relationships, thereby losing the global context and necessitating multiple layers to potentially perceive the correlations between distant sequence's portions [7, 45].

Attempts to enhance CNNs to surpass their inherent limitations have not succeeded in matching the accuracy of RNNs [32] and may limit model understanding [11]. Investigated enhancements include the introduction of dynamic attention layers [31], leveraging optical flow information [15, 19], incorporating physical knowledge [47], proposing tailored losses [15, 19, 32], and mimicking RNNs' sequential processing [15, 25]. In our study, we introduce a novel STL architecture that, solely relying on convolutional layers as learnable components, with the focus on minimizing the Mean Square Error (MSE) as the training objective, not only matches or surpasses RNNs in accuracy but also enhances the efficiency of previous CNN models.

We addressed the lack of temporal characterization in CNNs by providing a unified representation of spatial and temporal dimensions and rearranging both dimensions into a comprehensive spatio-temporal patch representation.

Our model further enhances the joint processing of temporal and spatial dynamics, employing a mixer paradigm [5, 33, 34] which repeatedly integrates inter-patches temporal information and intra-patches spatial information. To overcome CNNs' local receptive field limitations, we explicitly process near and distant intra-patches relationships, inspired by [17, 31].

Our contributions can be summarized as follows:

- For the first time in the STL context, we jointly process temporal and spatial dynamics, as opposed to the *Spatial-Temporal-Spatial* approach of current methods. In particular, as shown in Fig. 2, we employ (1) a Patch Embedding Encoder for joint spatio-temporal representation with a single convolution layer, (2) a STLMixer with large receptive field as convolutional-only backbone, and (3) an efficient Decoder relying on parameter-free pixel shuffle and a single convolution layer.
- We extensively assess our proposed model using challenging STL datasets. We focus on the possibility of scaling efficiently from low-resource to high-accuracy scenarios. Additionally, we rigorously evaluate each component of our model in various settings, establishing a solid foundation for enhanced performance.
- We achieve state-of-the-art results with a full-CNN architecture on major STL datasets, outperforming or matching previous methods in terms of accuracy, parameters, and FLOPs and improved convergence speed during training.

## 2. Background and Related work

### 2.1. Problem definition

In STL, we aim to forecast future video frames based on past observations. Given the past $T$ frames up to current time $t_0$, denoted as

$$\mathcal{X} = \{x_i\}_{t=t_0-T+1}^{t=t_0} \in \mathbb{R}^{T\times C\times H\times W} \tag{1}$$

our objective is to predict the next $T'$ frames, denoted as

$$\mathcal{Y} = \{x_i\}_{t=t_0+1}^{t=t_0+T'} \in \mathbb{R}^{T'\times C\times H\times W} \tag{2}$$

where each frame $x_i \in \mathbb{R}^{C\times H\times W}$ is usually a $H\times W$ image with $C$ channels.

### 2.2. Spatio-Temporal predictive Learning

Most STL approaches utilize a general *Spatial-Temporal-Spatial* framework [11, 31, 32]. In this framework, the encoder stage processes the input video frames, focusing on capturing spatial correlations only. The temporal module then leverages the spatial correlations within the frame representation to translate it into a corresponding representation of a future time point. Finally, the decoder stage reconstructs the spatial dimensions of the time-shifted representation, resulting in the output video frames. Based on the type of temporal block used, we can categorize STL models into recurrent-based and recurrent-free models.

**Recurrent-based models** Recurrent-based models leverage past predictions for individually forecasting each frame, explicitly modeling Markovian temporal evolution but limiting efficiency due to the necessity of making multiple single predictions. In this framework, ConvLSTM [27] enriches traditional LSTMs [29] by integrating convolutional layers, providing both spatial and temporal insights. PredRNN [41] and its advanced version, PredRNN++ [39], focus on the dual extraction of space and time representations, with the latter improving gradient propagation. MIM [43] investigates the video nuances with its self-renewed memory module, capturing the dynamic and stationary aspects of videos. MAU [4] utilizes a specialized unit to detect and encapsulate motion patterns within sequential data. IAM4VP [25]

and DMVFN [15] attempt to surpass RNNs limitations by using convolutions, while still retaining recurrent processing.

**Recurrent-free models** Recurrent-free approaches improve efficiency by directly forecasting all frames in a sequence, but face challenges with the coherence of predictions, and increased memory consumption as sequence length grows. To improve coherence, DVF [19] leverages optical flow information, while Pastnet [47] incorporates physical knowledge. SimVP [11] introduces a seminal framework, which employs stacked convolution layers for both encoding and decoding stages. A UNet architecture [22] based on inception blocks [30] is placed in the middle as a translator. Several works have built upon the SimVP framework, focusing particularly on proposing a more competitive translator. TAU [31] incorporates static attention mechanisms to analyze relations between frames and dynamic attention to monitor changes over time. Tan et al. [32] further expand the versatility of the SimVP framework by replacing the temporal translator with successful vision architectures inspired by the transformer model [36] such as ViT [9], MLP-Mixer [33], ConvMixer [35], and ConvNeXt [18], leveraging optical flow information [15, 19], incorporating physical knowledge [47], and proposing tailored losses [15, 19, 32].

# 3. Methods

In this section, we present our architecture, depicted in Figure 2. The sequence of input frames is first encoded as spatio-temporal patches (Section 3.1), processed by STLMixer blocks (Section 3.2), and decoded through patch shuffle and reassemble (Section 3.3).

## 3.1. Spatio-Temporal Patches

Given an input batch composed by the sequences of observed frames $\mathcal{B}_T \in \mathbb{R}^{B\times T\times C\times H\times W}$, previous methods [11, 31, 32] encode each frame individually, rearranging them into the batch dimension and describing the spatial features within each frame of the batch of sequences. Unfortunately, this process treats each frame equally, without explicitly exploiting temporal relationships. Given the limitations of a fully convolutional approach in capturing both spatial and temporal information, we fuse those dimensions by dividing each frame into patches, with each patch including both spatial and temporal details.

This is achieved by interleaving frames in the channel dimensions $Z_T \in \mathbb{R}^{B\times(T\cdot C)\times H\times W}$, as Voleti et al. [38], and dividing $H \times W$ into patches by using a stride of $p$ and determining the kernel size and padding based on the value of $O$. Specifically, if $O \geq 2$, the stride is set to $p$, the kernel size is $p \cdot O$, and the padding is $\left\lfloor \frac{(O-1)\cdot p}{2} \right\rfloor$. Otherwise, the stride is $p$, the kernel size is $p$, and no padding is applied. To ensure continuity in the representation and maximize spatial information, we adopt extensive patch overlapping. To further enhance efficiency, the encoding is performed with a single convolution. where $p$ is the patch size and $O$ is the desired overlap. Following the convolution, we obtain a tensor $Z'_T \in \mathbb{R}^{B\times d\times H/p\times W/p}$, representing the embedded frames sequence in the hidden dimension $d$.

Specifically, our embedding strategy aims to preserve the spatial resolution of the embedded tensor as an integer divisor of the initial frames' spatial resolution, allowing the patch shuffle method employed in the decoding phase, as described in Section 3.3, and facilitating an efficient fine-tuning strategy, as discussed in Section 4.5. To assist the efficient frame representation, we recommend a significant overlap, small patches, and large hidden temporal dimension, as we will demonstrate in the experimental section. This approach minimizes the information compressed per patch while still enabling a $p^2$ resolution reduction.

## 3.2. STLMixer

After encoding, the input frames' spatio-temporal relations are represented via patches and their connections. To capture the complex dynamics inherent spatio-temporal learning we necessitate to analyze both proximal and distal patch relationships. This is particularly important with small patches, as proposed in Section 3.1, which increases their number and, consequently, their distances.

To facilitate this processing, we introduce a Mixer architecture [5, 33, 34], which cyclically intermixes information among and within patches, respectively in their spatial and temporal dimension. To effectively grasp local and global relationships, we propose the STLMixer, an advanced ConvMixer [35] that includes a two-stage intra-patches mixer based on [17, 32]. In the STLMixer block, a compact kernel $k_{T_1}$ captures local fine-grained details. Subsequently, a dilated convolution layer broadens the receptive field to incorporate global information, utilizing a larger kernel size $k_{T_2}$. This dual convolution approach merges the intricate details of the representation with an overarching contextual understanding. This STLMixer block is repeated $\texttt{de}$ times in our architecture, with a skip connection between the block $\texttt{de}/3$ and $2 \cdot \texttt{de}/3$, providing into later stage processing an earlier frames representation to guide output frames reconstruction.

To facilitate better frame representation by encoding each patch with a larger hidden dimension $d$, as proposed in Section 3.1, we avoid any intra-patches attention mechanism, due to the quadratic cost associated in $d$ in a $d \times d$ attention mechanism. Our approach, which retains a similar depth-wise and point-wise scheme while avoiding any specialized attention mechanisms [31], contrasts with the assumptions of TAU [31] regarding the efficacy of convolution in processing spatio-temporal dynamics. We investi-

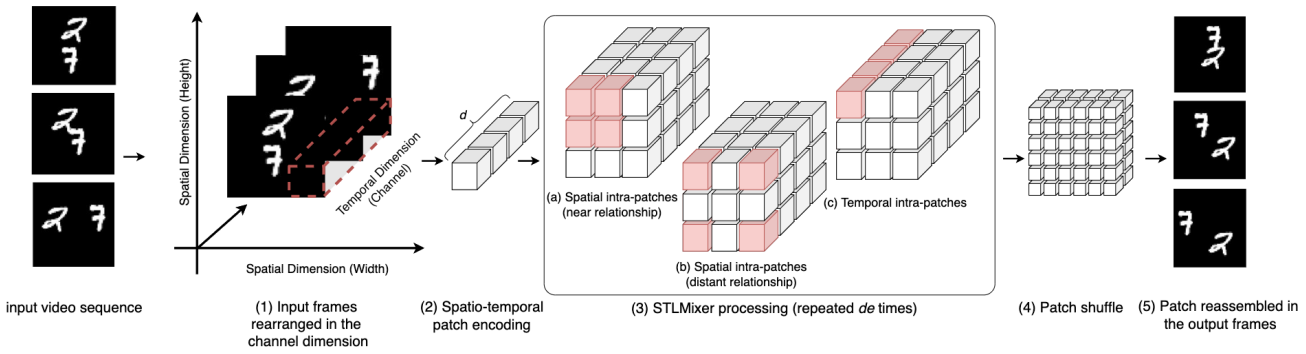


Figure 2. STLight model workflow. We rearrange the input sequence of frames along the channel dimension (1), and through a single convolutional layer, we encode the sequence into patches of size $p \times p$ with hidden temporal dimension $d$, containing both spatial and temporal information (2). The patches are processed through a custom STLMixer block repeated $\texttt{de}$ times (3). Each block processes the relationships between near (a) and distant (b) intra patches along the spatial dimension, as well as the intra-patch relationships on the temporal dimension (c). We decode the output sequence, restoring the initial spatial resolution through a patch shuffle (4) and the temporal resolution by reassembling the patches into the final output sequence (5).

gate the major effectiveness of STLMixer compared to ConvMixer and TAU through an ablation study in Section 4.6.

### 3.3. Patch Shuffle and Reassemble

While many methods employ sequences of transposed convolutions to recover lost resolution [11, 31, 32, 49], our approach utilizes a single convolution coupled with a learnable, parameter-free shuffle technique.

Following our proposed encoder stage (3.1), the spatial resolution is diminished by a factor of $p^2$. To efficiently restore the original resolution, we apply the PixelShuffle operator [26], which reorganizes elements from a tensor of shape $(B \times d \times H/p \times W/p)$ into a tensor of shape $(B \times d/p^2 \times H \times W)$ achieving the desired output resolution rearranging the patches without learnable layers.

Next, we reassemble the feature space from $(B \times d/p^2 \times H \times W)$ to the targeted $(B \times (T' \cdot C) \times H \times W)$ output sequence. This step is executed through an efficient $1 \times 1$ convolutional layer, reassembling $d/p^2$ input channels to $T' \cdot C$ output channels. The process can be represented as follows:

$$Z'''_T = \text{Conv}_{\text{kernel\_size}=1}\left(\text{PixelShuffle}(Z''_{t,T})\right) \quad (3)$$

With $Z''_T$ corresponding to the spatio-temporal patches after the STLMixer processing. The final target shape $(B \times T' \times C \times H \times W)$ is restored by reshaping.

## 4. Experiments

In this section, we present experiments to validate the effectiveness of our method. Additional Experiments can be found in the Appendix.

- **Standard Spatio-Temporal Predictive Learning:** (Section 4.2) Predicting a constant number of output frames is a standard problem in spatio-temporal predictive learning [31]. We compare STLight with established methods on Moving MNIST [28] and TaxiBJ benchmark datasets [52].

- **Long Sequence Frame Prediction:** (Section 4.3) Predicting longer frame sequences is a central task in STL because it involves a deep understanding of the evolution of the scene. We evaluate our model on the KTH dataset [24] for the task of predicting the next 20 or 40 frames given 10 past observations, also considering the computational demands of processing longer sequences.

- **Self-Supervised Learning Capabilities:** (Section 4.4) Central to self-supervised learning is the acquisition of robust, domain-independent knowledge by learning more from each data sample. We investigate the domain generalization effectiveness by training STLight models with different parameter ranges (from 0.1 to 15 million) on the KITTI dataset and testing them on the Caltech Pedestrian dataset, assessing efficiency by comparing the training speed of our method with other unsupervised approaches.

- **Hyperparameter Tuning for Accuracy and Efficiency:** (Section 4.5) The impact of STLight's hyperparameter configuration on model performance is explored, offering strategies for fine-tuning these parameters within a specified computational budget to achieve optimal outcomes.

- **Ablation Study:** (Section 4.6) Through a comprehensive ablation study on various parameter settings, we compare our solutions with conventional practices,

providing insights into the efficacy of our methods compared to established approaches.

### 4.1. Experimental Setups

**Datasets** In line with common choices by relevant prior works [39–41], we quantify the performance of our model on the following synthetic and real-world datasets:

- **Moving MNIST** [28] (MMNIST) is the fundamental benchmark in STL, consisting of video sequences that depict two independently moving digits. These digits move at different speeds, frequently intersect, and bounce off the edges.
- **TaxiBJ** [52] includes taxi GPS trajectory inflow and outflow data, collected from taxicabs in Beijing. Consistent with prior research [31, 43], we normalize the data to the range $[0, 1]$.
- **KTH** [24] is a human motion dataset that features six types of movements performed by 25 individuals across four different scenarios. In alignment with previous studies [37, 40], we use individuals 1-16 for training and 17-25 for validation.
- **KITTI** [12] and **Caltech** [8] are urban datasets featuring videos from vehicles navigating urban environments. Following the protocols established by previous studies [20, 50], we train STLight on the KITTI dataset and evaluate its performance on the Caltech Pedestrian dataset.

Table 1. Datasets composition. The training and testing set have $N_{\text{train}}$ and $N_{\text{test}}$ samples, respectively. We predict the future $T'$ frames from the past $T$.

| | $N_{\text{train}}$ | $N_{\text{test}}$ | $(C, H, W)$ | $T$ | $T'$ |
|---|---|---|---|---|---|
| MMNIST | Generated | 10000 | (1, 64, 64) | 10 | 10 |
| TaxiBJ | 20461 | 500 | (2, 32, 32) | 4 | 4 |
| KTH | 4940 | 3030 | (1, 128, 128) | 10 | 20 |
| Caltech | 2042 | 1983 | (3, 128, 160) | 10 | 1 |

**Measurement** In this work, we consider both accuracy and computational resources utilization to provide a comprehensive evaluation. Accuracy is measured through Mean Squared Error (MSE), Mean Absolute Error (MAE), Structure Similarity Index Measure (SSIM), and Peak Signal-to-Noise Ratio (PSNR). MSE and MAE estimate the absolute pixel-wise errors, SSIM measures the similarity of structural information within the spatial neighborhoods, and PSNR measures the quality difference between an original and a reconstructed image, quantifying the level of distortion or noise. Computational resources are assessed by the parameters count and FLOPs, measured through the fvcore library [10].

**Train-eval settings** We implement our work using OpenSTL [32], a well-established open-source framework. Our model is trained using the Mean Squared Error (MSE) as the training objective, and the best hyper-parameters are identified through a grid-search approach. To guarantee replicability and fair evaluation, we compare our results against OpenSTL's public results, ensuring the same consistent training setting for all comparisons. To robustly evaluate the performance and scalability of STLight, we train multiple instances of the model on each dataset, varying only the number of parameters. The training parameters used for each dataset are reported in Table 2, and we will release OpenSTL configuration files to facilitate the reproduction of our results. Experiments are conducted on a single NVIDIA RTX A6000 GPU with 48GB of VRAM.

Table 2. Hyperparameters settings for each dataset.

| | MMNIST | TaxiBJ | KTH | Caltech |
|---|---|---|---|---|
| Learning Rate (lr) | 0.003 | 0.003 | 0.0005 | 0.01 |
| Final div factor | 10000 | 10000 | 10000 | 3000 |
| Batch Size | 16 | 16 | 12 | 8 |
| LR Scheduler | OneCycle | Cosine | OneCycle | OneCycle |
| Optimizer | Adam | Adam | Adam | Adam |
| Epochs | 200 | 50 | 100 | 100 |

### 4.2. Standard STL benchmarks

#### 4.2.1 Moving MNIST

This dataset is a standard benchmark in STL. We evaluate four variants of STLight, which differ in the number of parameters used, against existing STL methods. The results are reported in Table 3.

STLight, with its convolutional design, outperforms both recurrent-based and recurrent-free models across all metrics, achieving more accuracy while requiring significantly less computational resources (FLOPs). Remarkably, under standardized settings, STLight-XS surpasses the current state-of-the-art recurrent-free architecture by using only 25% of its parameters, while STLight-S surpasses the best recurrent architecture with only 14% of its FLOPs.

In Figure 1, we demonstrate that STLight can efficiently scale across all ranges and surpass previous methods in both accuracy and efficiency. Qualitative visualizations of the predicted results are shown in Figure 3. Even when the input frames vary significantly from future frames, our model effectively generates dependable results, capturing movement properties such as direction, speed, and changes in direction at the edges. This underscores the model's capability to capture the underlying dynamics and generate accurate future sequences.

Table 3. Quantitative results demonstrating our model's performance in accuracy and computational efficiency compared to OpenSTL's published benchmark baselines under equivalent training and evaluation conditions on MMNIST, TaxiBJ, and KTH datasets.

| Method | MMNIST | | | | | TaxiBJ | | | | | KTH | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSE ↓ | MAE ↓ | SSIM ↑ | Params | FLOPs | MSE × 100 ↓ | MAE ↓ | SSIM ↑ | Params | FLOPs | MAE ↓ | PSNR ↑ | SSIM ↑ | Params | FLOPs |
| **Recurrent Methods** | | | | | | | | | | | | | | | |
| ConvLSTM [27] | 29.80 | 90.64 | 0.9288 | 15.0M | 56.8G | 33.58 | 15.32 | 0.9836 | 14.98M | 20.74G | 445.5 | 26.99 | 0.8977 | 14.9M | 1368G |
| E3D-LSTM [40] | 35.97 | 78.28 | 0.9320 | 51.0M | 298.9G | 34.27 | 14.98 | 0.9842 | 50.99M | 98.19G | 136.40 | 892.7 | 0.8153 | 53.5M | 217G |
| PhyDNet [13] | 28.19 | 78.64 | 0.9374 | 3.1M | 15.3G | 36.22 | 15.53 | 0.9828 | 3.09M | 5.60G | 765.6 | - | 0.8322 | 3.1M | 93.6G |
| MAU [4] | 26.86 | 78.22 | 0.9398 | 4.5M | 17.8G | 32.68 | 15.26 | 0.9834 | 4.41M | 6.02G | 471.2 | 26.73 | 0.8945 | 20.1M | 399G |
| MIM [43] | 22.55 | 69.97 | 0.9498 | 38.0M | 179.2G | 31.1 | 14.96 | 0.9847 | 37.86M | 64.10G | 380.8 | 27.78 | 0.9025 | 39.8M | 1099G |
| PredRNN [41] | 23.97 | 72.82 | 0.9462 | 23.8M | 116.0G | 31.94 | 15.31 | 0.9838 | 23.66M | 42.40G | 380.6 | 27.81 | 0.9097 | 23.6M | 2800G |
| PredRNN++ [39] | 22.06 | 69.58 | 0.9509 | 38.6M | 171.7G | 33.48 | 15.37 | 0.9834 | 38.40M | 62.95G | 370.4 | **28.13** | **0.9124** | 38.3M | 4162G |
| PredRNNv2 [42] | 24.13 | 73.73 | 0.9453 | 23.9M | 116.6G | 38.34 | 15.55 | 0.9826 | 23.67M | 42.63G | 368.8 | 28.01 | 0.9099 | 23.6M | 2815G |
| DMVFN [15] | 123.67 | 179.96 | 0.8140 | 3.5M | 0.2G | 339.5 | 45.526 | 0.8321 | 3.54M | 0.057G | 413.2 | 26.65 | 0.8976 | 3.5M | 0.88G |
| **Recurrent-free Methods** | | | | | | | | | | | | | | | |
| SimVP+ConvMixer [32] | 32.09 | 88.93 | 0.9259 | 3.9M | 5.5G | 36.34 | 15.63 | 0.9831 | 0.84M | 0.23G | 446.1 | 26.66 | 0.8993 | 1.5M | 18.3G |
| SimVP+ViT [32] | 35.15 | 95.87 | 0.9139 | 46.1M | 16.9G | - | - | - | - | - | - | - | - | - | - |
| SimVP+InceptU [11] | 32.15 | 89.05 | 0.9402 | 58.0M | 19.4G | 32.82 | 15.45 | 0.9835 | 13.79M | 3.61G | 397.1 | 27.46 | 0.9065 | 12.2M | 62.8G |
| TAU [31] | 24.60 | 71.93 | 0.9454 | 44.7M | 16.0G | 31.08 | **14.93** | 0.9848 | 9.55M | 2.49G | 421.7 | 27.10 | 0.9086 | 15.0M | 73.8G |
| *STLight-XS (Ours)* | 24.48 | 71.21 | 0.9444 | 11.1M | 10.6G | - | - | - | - | - | 376.1 | 27.50 | 0.9052 | 1.4M | 5.4G |
| *STLight-S (Ours)* | 23.29 | 68.33 | 0.9454 | 17.1M | 16.5G | 34.79 | 15.58 | 0.9825 | 0.99M | 0.82G | 377.8 | 27.52 | 0.9078 | 5.4M | 20.9G |
| *STLight-M (Ours)* | 22.21 | 68.33 | 0.9496 | 24.3M | 23.7G | 32.54 | 15.25 | 0.9839 | 1.65M | 1.37G | 367.9 | 27.54 | 0.9102 | 9.5M | 37.1G |
| *STLight-L (Ours)* | **21.80** | **66.92** | **0.9515** | 32.9M | 32.3G | **30.87** | 15.00 | **0.9853** | 2.96M | 2.71G | **363.8** | 27.57 | 0.9113 | 14.6M | 57.8G |

#### 4.2.2 TaxiBJ

In this section, we evaluate STLight on the TaxiBJ dataset, a standard benchmark in real-world traffic prediction. This dataset poses the challenge of discerning how external factors, from weather shifts to unexpected events, significantly alter traffic behaviors. Given the low-resolution nature of this dataset, which is only $32 \times 32$ pixels, we utilize a patch size of $p = 1$. As presented in Table 3, STLight outperforms other notable methods. For example STLight-L achieves the most optimal MSE $\times$ 100 and SSIM, despite having 69% fewer parameters than the current state-of-the-art model, TAU. Qualitative results are shown in Figure 3.

### 4.3. Long sequence frames prediction

Recurrent methods are capable of predicting long coherent sequences of frames from a limited number of past observations, by feeding predicted frames back into the network and recursively outputting predictions. However, recurrent-free methods [11, 31] can emulate such processing, but at the cost of efficiency due to the necessity for multiple computations.

In contrast, we evaluate STLight's ability to directly predict the complete target long sequence, leveraging only 10 given input frames to generate the next 20 frames. This approach facilitates efficient parallelization and demonstrates effective long-range video prediction capabilities.

In Table 3 STLight-L and STLight-M match or surpass current state-of-the-art methods. Notably, our STLight-L matches PredRNNv2 while using only 61% of its parameters and 2% of its FLOPs. Other STLight variants are even more cost-efficient, with only a slight loss in accuracy. These results highlight STLight's ability to predict extended sequences with high accuracy while requiring significantly fewer computational resources, aligning with findings in Sections 4.2.1 and 4.2.2.

### 4.4. Unsupervised Learning effectiveness

#### 4.4.1 Domain generalization

Domain adaptation, or out-of-distribution generalization, is a common evaluation scenario for STL [11, 20, 51]. This evaluation focuses on the challenging task of training a model on one domain, with the goal of successfully generalizing to a new domain. This presents a unique challenge due to differing data distributions across domains, requiring models to extract transferable knowledge and adapt to new settings.

We assess STLight's capability to generalize by using the KITTI and Caltech pedestrian datasets, as described in Section 4.1. Additionally, we assess STLight's scaling capabilities by varying the number of its parameters, ranging from 0.1M to 15M, and comparing it with OpenSTL baselines of comparable sizes, not exceeding 25M parameters. Figure 4 presents our findings. STLight consistently demonstrates strong domain generalization across all parameter sets, outperforming notable methods while using significantly fewer parameters. Notably, STLight surpasses competitive models such as MAU, PredRNN.V2, SimVP, and PredRNN, using respectively only 1.5%, 20%, 60%, and 65% of their parameters.

#### 4.4.2 Sample Efficiency in Training

Sample efficiency measures a model's ability to achieve a predefined performance level with the minimal amount of training samples; in other words, it reflects the learning speed from training data [44, 46].

In this section, we train the existing models and STLight-32M on the Moving MNIST dataset under standardized


Figure 3. Qualitative results on Moving MNIST and TaxiBJ datasets.

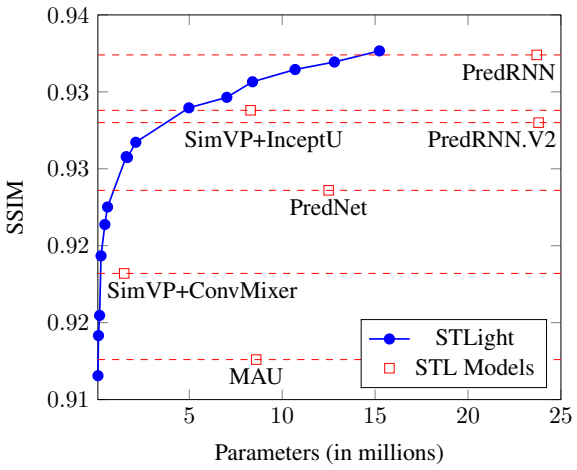


Figure 4. STLight models trained on KITTI (0.1M-15M parameters) outperform baselines on Caltech, demonstrating strong cross-dataset generalization with more efficient resource utilization.

conditions.

As depicted in Figure 5, STLight consistently outperforms other methods in terms of sample efficiency. Beyond the initial 21 epochs, STLight achieves a lower MSE compared to established methods across all subsequent training durations. This superior performance is exemplified by STLight achieving an MSE of 37.38 at epoch 50, whereas other models require at least an additional 19 epochs to reach comparable accuracy. This demonstrates STLight's remarkable ability to learn effectively with fewer training samples, highlighting its potential for efficient training and superior data utilization.

### 4.5. Tuning for accuracy and efficiency

An effective learning model must balance accuracy with computational efficiency. In this section, we investigate the role of STLMixer's hyperparameters in achieving this balance. We begin by examining their relationship under preliminary settings, then explore strategies for scaling the STLMixer in low-resource environments, and finally, we adjust them for an increase in accuracy, covering a wide spectrum of applications.

**Preliminaries** As shown in Appendix A, the parameter complexity of STLight scales with $O(\texttt{de} \cdot d^2)$: the balance between the hidden dimension $d$ of spatio-temporal patches and the number of repeated STLMixer blocks $\texttt{de}$ is crucial for STLight's optimal efficiency. We focus on the validation loss curve $L(\texttt{de}, d)$ (Figure 6). In particular, for a fixed $d$, we are interested in the "elbow" point [23] of $L$, which is the point that marks the transition from significant to negligible MSE improvements, thereby providing valuable information regarding the optimal configuration of the model parameters. Empirical data indicates that this elbow occurs when $\texttt{de} \approx 16$. Additionally, the parameters $k_{T_1}$ and $k_{T_2}$ significantly affect STLight's ability to exploit local and global features. As discussed in Appendix B, we find that $k_{T_1} = 3$ and $k_{T_2} = 7$ are optimal for general applications.

**Low parameters tuning** To effectively reduce the STLight parameter count and limit the loss in accuracy, we maintain fixed to their optimal values ($k_{T_1} = 3$, $k_{T_2} = 7$ and $\texttt{de} = 16$), and we tune the parameter $d$, scaling the parameters count by a $d^2$ factor, considering that STLMixer blocks has complexity $O(\texttt{de} \cdot d^2)$, as described in Appendix A. Reducing $d$ limits the expressive power of each patch. Hence we suggest to represent the input sequences by more patches, reducing their size dimension $p$. Moreover, with a reduced $d$, the increase in FLOPs caused by more patches is mitigated, achieving an advantageous balance.

**High accuracy tuning** To increase STLight accuracy, efficiently utilizing more computational budget, we propose to maintain fixed to their optimal values $k_{T_1}$, $k_{T_2}$ and $\texttt{de}$, and we suggest to improve patch representation power, increasing $d$ and using large patch overlapping. In Figure 7, we

Table 4. Ablation study of our proposed method

| Our modules with ... | Architecture composition | | | | MSE for different # Parameters | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Encoder | Decoder | Translator | Inter-block skip connection | 11M | 17M | 24M | 32M | 42M | 79M |
| $(B \cdot T) \times C$ Encoder and Decoder | TAU | TAU | Ours | at 1/3 and 2/3 | 27.89 | 26.56 | 25.69 | 25.26 | 24.47 | - |
| TAU Encoder and Decoder | TAU | TAU | Ours | at 1/3 and 2/3 | 27.89 | 26.56 | 25.69 | 25.26 | 24.47 | - |
| Pyramidal convolution Encoder | Pyramid | Ours | Ours | at 1/3 and 2/3 | 26.81 | 25.27 | 23.98 | 23.33 | 22.94 | - |
| Pyramidal deconvolution Decoder | Ours | Pyramid | Ours | at 1/3 and 2/3 | 27.89 | 27.36 | 26.44 | 25.59 | 24.75 | - |
| Standard ConvMixer Translator | Ours | Ours | ConvMixer | NA | 28.64 | 27.81 | 26.53 | 25.74 | 24.82 | - |
| Standard TAU Translator | Ours | Ours | TAU | NA | - | - | - | - | - | 45.12 |
| No inter-block skip connection | Ours | Ours | Ours | no skip | 24.86 | 23.66 | 22.59 | 22.28 | 21.97 | - |
| Skip connection at 1/5 and 4/5 | Ours | Ours | Ours | at 1/5 and 4/5 | 24.94 | 23.68 | 22.62 | 22.13 | 21.92 | - |
| STLight (Ours) | Ours | Ours | Ours | at 1/3 and 2/3 | 24.48 | 23.29 | 22.21 | 21.80 | 21.55 | - |

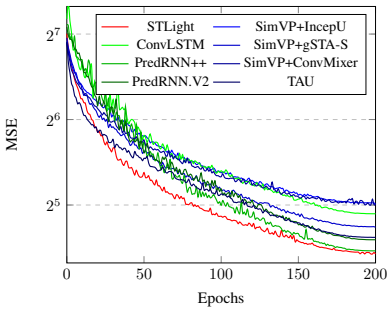


Figure 5. Learning curve comparison between state-of-the-art methods and ours.

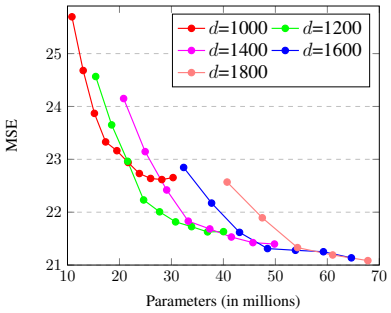


Figure 6. Learning curve comparison for different values of $d$ and `de`.

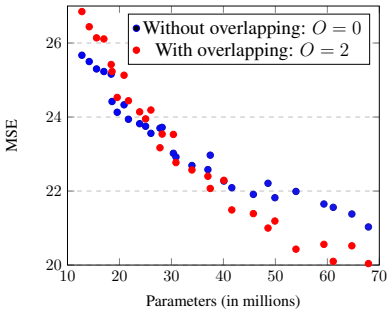


Figure 7. Learning curves with different patch overlapping, keeping $p = 2$.

explore the impact of the overlapping parameter $O$ on the validation loss across different model sizes. Our empirical analysis demonstrates that larger models benefit from large overlap ($O = p = 2$), while smaller models achieve optimal performance with no overlap ($O = 0$).

### 4.6. Ablation study

We assess our methods against well-established practices in STL through an extensive ablation study. Our evaluations, summarized in Table 4, involve replacing each of our components with a comparable architecture, while maintaining a parameter range between 11M and 42M and ensuring a uniform depth of 16 layers.

We validate the integration of spatial and temporal data into spatio-temporal patches from two perspectives: by replacing the encoder and decoder within the SimVP framework [11, 31], and by comparing our approach of using a single convolutional layer against multiple convolutions in encoders or decoders. This replacement led to decreased performance across configurations due to the increased computational demands, despite maintaining the translator block's budget. Additionally, we assessed our mixer backbone by incorporating the STLMixer with a spatio-temporal layer from TAU [31]. We observed that adding extra attention layers significantly increased the parameter count leading to inefficiency. These findings suggest the potential for exploring attention layers that are better suited for spatio-temporal patches. Substituting the translator with ConvMixer [35] underscored the importance of a wide receptive field in STL.

## 5. Conclusions

We introduce STLight, an innovative STL architecture that either matches or surpasses the accuracy of the state-of-the-art recurrent methods, while also offering greater efficiency than recurrent-free methods. This work challenges the prevailing perspectives in STL, showing that convolutions alone can effectively joint capture spatial and temporal dependencies, eliminating the need for complex additional modules and strategies. We achieve these results by introducing spatio-temporal patches, an enhanced representation of a sequence of frames, that joint integrates both temporal and spatial information, and by surpassing traditional *Spatial-Temporal-Spatial* paradigm towards a more comprehensive framework where both spatial and temporal information are jointly integrated. Furthermore, the reduced computational demand facilitates scaling to a wider range of scenarios. We believe this contribution could serve as a robust baseline and inspire future research in the field.

**Acknowledgement** This study has been partially supported by SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU

## References


[1] Shivam Akhauri, Laura Zheng, Tom Goldstein, and Ming Lin. Improving generalization of transfer learning across domains using spatio-temporal features in autonomous driving. *arXiv preprint arXiv:2103.08116*, 2021. 1

[2] Alessandro Antonucci, Gastone Pietro Rosati Papini, Paolo Bevilacqua, Luigi Palopoli, and Daniele Fontanelli. Efficient prediction of human motion for real-time robotics applications with physics-inspired neural networks. *IEEE Access*, 10:144–157, 2021. 1

[3] Judith Bütepage, Hedvig Kjellström, and Danica Kragic. Anticipating many futures: Online human motion prediction and generation for human-robot interaction. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4563–4570. IEEE, 2018. 1

[4] Zheng Chang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, Yan Ye, Xiang Xinguang, and Wen Gao. Mau: A motion-aware unit for video prediction and beyond. *Advances in Neural Information Processing Systems*, 34:26950–26962, 2021. 2, 6

[5] Shoufa Chen, Enze Xie, Chongjian Ge, Runjian Chen, Ding Liang, and Ping Luo. Cyclemlp: A mlp-like architecture for dense prediction. *arXiv preprint arXiv:2107.10224*, 2021. 2, 3

[6] Shuo Cheng, Bo Yang, Zheng Wang, and Kimihiko Nakano. Spatio-temporal image representation and deep-learning-based decision framework for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):24866–24875, 2022. 1

[7] Nadav Cohen and Amnon Shashua. Inductive bias of deep convolutional networks through pooling geometry. *arXiv preprint arXiv:1605.06743*, 2016. 2

[8] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *2009 IEEE conference on computer vision and pattern recognition*, pages 304–311. IEEE, 2009. 5

[9] 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. 3

[10] facebookresearch. fvcore. https://github.com/facebookresearch/fvcore, 2019. 5

[11] Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z Li. Simvp: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3170–3180, 2022. 2, 3, 4, 6, 8

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 5

[13] 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. 6

[14] Pratik Gujjar and Richard Vaughan. Classifying pedestrian actions in advance using predicted video of urban driving scenes. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2097–2103. IEEE, 2019. 1

[15] Xiaotao Hu, Zhewei Huang, Ailin Huang, Jun Xu, and Shuchang Zhou. A dynamic multi-scale voxel flow network for video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6121–6131, 2023. 2, 3

[16] Jangwon Lee and Michael S Ryoo. Learning robot activities from first-person human videos using convolutional future regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–2, 2017. 1

[17] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Mykola Pechenizkiy, Decebal Mocanu, and Zhangyang Wang. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. *arXiv preprint arXiv:2207.03620*, 2022. 2, 3

[18] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 3

[19] 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

[20] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016. 5, 6

[21] Khushdeep S Mann, Abhishek Tomy, Anshul Paigwar, Alessandro Renzaglia, and Christian Laugier. Predicting future occupancy grids in dynamic environment with spatio-temporal learning. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 1121–1126. IEEE, 2022. 1

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 3

[23] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. 7

[24] 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. 4, 5

[25] Minseok Seo, Hakjin Lee, Doyi Kim, and Junghoon Seo. Implicit stacked autoregressive model for video prediction. *arXiv preprint arXiv:2303.07849*, 2023. 2

[26] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In

*Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 4

[27] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015. 2, 6

[28] 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. 4, 5

[29] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014. 2

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 3

[31] Cheng Tan, Zhangyang Gao, Lirong Wu, Yongjie Xu, Jun Xia, Siyuan Li, and Stan Z Li. Temporal attention unit: Towards efficient spatiotemporal predictive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18770–18782, 2023. 2, 3, 4, 5, 6, 8

[32] Cheng Tan, Siyuan Li, Zhangyang Gao, Wenfei Guan, Zedong Wang, Zicheng Liu, Lirong Wu, and Stan Z Li. Openstl: A comprehensive benchmark of spatio-temporal predictive learning. *arXiv preprint arXiv:2306.11249*, 2023. 2, 3, 4, 5, 6

[33] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021. 2, 3

[34] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):5314–5321, 2022. 2, 3

[35] Asher Trockman and J Zico Kolter. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022. 3, 8

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[37] 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. 5

[38] Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in Neural Information Processing Systems*, 35:23371–23385, 2022. 3

[39] 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. 2, 5, 6

[40] 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. 5, 6

[41] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Advances in neural information processing systems*, 30, 2017. 2, 5, 6

[42] Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, S Yu Philip, and Mingsheng Long. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2208–2225, 2022. 6

[43] 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 nonstationarity from spatiotemporal dynamics. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9154–9162, 2019. 2, 5, 6

[44] Zhihai Wang, Jie Wang, Qi Zhou, Bin Li, and Houqiang Li. Sample-efficient reinforcement learning via conservative model-based actor-critic. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8612–8620, 2022. 6

[45] Zihao Wang and Lei Wu. Theoretical analysis of the inductive biases in deep convolutional networks. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[46] Gellért Weisz, Paweł Budzianowski, Pei-Hao Su, and Milica Gašić. Sample efficient deep reinforcement learning for dialogue systems with large action spaces. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(11):2083–2097, 2018. 6

[47] Hao Wu, Wei Xion, Fan Xu, Xiao Luo, Chong Chen, Xian-Sheng Hua, and Haixin Wang. Pastnet: Introducing physical inductive biases for spatio-temporal video prediction. *arXiv preprint arXiv:2305.11421*, 2023. 2, 3

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

[49] Xi Ye and Guillaume-Alexandre Bilodeau. Vptr: Efficient transformers for video prediction. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 3492–3499. IEEE, 2022. 4

[50] Wei Yu, Yichao Lu, Steve Easterbrook, and Sanja Fidler. Crevnet: Conditionally reversible video prediction, 2019. 5

[51] Wei Yu, Yichao Lu, Steve Easterbrook, and Sanja Fidler. Efficient and information-preserving future frame prediction and beyond. 2020. 6

[52] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, Xiuwen Yi, and Tianrui Li. Predicting citywide crowd flows using deep spatio-temporal residual networks. *Artificial Intelligence*, 259:147–166, 2018. 4, 5

[53] Yang Zhao and Yong Dou. Pose-forecasting aided human video prediction with graph convolutional networks. *IEEE Access*, 8:147256–147264, 2020. 1