TaylorSwiftNet: Taylor Driven Temporal Modeling for Swift Future Frame Prediction

Saber Pourheydari*[1]
m.saberpourheydari@gmail.com

Emad Bahrami*[1]
bahrami@iai.uni-bonn.de

Mohsen Fayyaz*[1, 2]
mohsenfayyaz@microsoft.com

Gianpiero Francesca[3]
gianpiero.francesca@toyota-europe.com

Mehdi Noroozi[4]
m.noroozi@samsung.com

Juergen Gall[1]
gall@iai.uni-bonn.de

[1] Computer Vision Group
   University of Bonn
   Bonn, Germany

[2] Microsoft
   Berlin, Germany

[3] Toyota Motor Europe
   Brussels, Belgium

[4] Samsung AI
   Cambridge, UK

 * indicates equal contribution

Abstract

While recurrent neural networks (RNNs) demonstrate outstanding capabilities for future video frame prediction, they model dynamics in a discrete time space, i.e., they predict the frames sequentially with a fixed temporal step. RNNs are therefore prone to accumulate the error as the number of future frames increases. In contrast, partial differential equations (PDEs) model physical phenomena like dynamics in a continuous time space. However, the estimated PDE for frame forecasting needs to be numerically solved, which is done by discretization of the PDE and diminishes most of the advantages compared to discrete models. In this work, we, therefore, propose to approximate the motion in a video by a continuous function using the Taylor series. To this end, we introduce TaylorSwiftNet, a novel convolutional neural network that learns to estimate the higher order terms of the Taylor series for a given input video. TaylorSwiftNet can swiftly predict future frames in parallel and it allows to change the temporal resolution of the forecast frames on-the-fly. The experimental results on various datasets demonstrate the superiority of our model.

1 Introduction

The ability to predict future frames of a video is essential for many applications such as weather forecasting [55], autonomous driving [25], robotics [14], or action recognition [29]. When only the raw video is given, the task is very challenging since it requires to learn the complex motion of the objects present in the video. To address this task, several approaches have been proposed over the last years. In particular, auto-regressive methods and recurrent neural networks (RNNs) have been popular [50, 51, 52, 55].
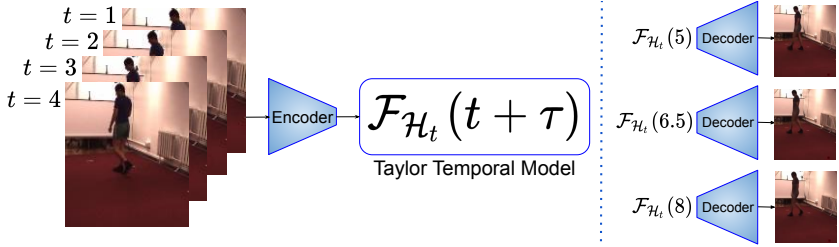
Figure 1: Given a sequence of observed frames until time t, the encoder maps them into a latent space. Our network infers from the observations in the latent space $\mathcal{H}_t$, a continuous function $\mathcal{F}_{\mathcal{H}_t}(t + \tau)$ in one forward pass. The inferred function can then be evaluated for any positive value τ in order to forecast future frames at $t + \tau$. This can be done in parallel for different values of τ and the decoder maps the forecast frames back to the image domain.

While these approaches learn the motion implicitly, recently a new line of work appeared that leverages partial differential equations (PDEs) and deep learning for forecasting video frames [18, 30, 51, 39, 41]. PDEs are very appealing for this task since they are an appropriate tool to model physical phenomena like dynamics. These methods model motion in the continuous time space in contrast to auto-regressive models or recurrent neural networks that model motion sequentially in a discrete time space. This has two major advantages. First, the motion model is independent of the sampling rate and it is, for instance, possible to forecast the motion at a higher sampling rate than the observed frames. Second, which is more important, a continuous PDE can be solved for any future point. It is therefore not required to sequentially go through all frames until the desired point in the future is reached, which increases the inference time for points that are more distant in the future and which is prone to error accumulation. The estimated PDEs, however, do not provide directly the prediction, but they need to be numerically solved, which diminishes most of the advantages compared to discrete models.

In this work, we therefore propose a novel approach that takes full advantage of a continuous representation of motion. Instead of learning a PDE that needs to be numerically solved for training and inference, we directly infer a continuous function that describes the future. In contrast to RNNs that forecast the future frame-by-frame or PDE-based approaches that discretize PDEs to solve them numerically, we infer a continuous function over time from the observations. This avoids discretization artifacts and provides an analytical function that can be swiftly evaluated for any future continuous point as illustrated in Fig. 1. This allows, for instance, to generate frames at different future points in parallel. We can also forecast future frames at a higher sampling rate than the observed frames. All these properties are very useful for practical applications and demonstrate the advantages of a continuous representation.

It is, however, very challenging to infer such a continuous function from a discrete set of high-dimensional observations as it is the case for video frame forecasting. We therefore propose to approximate the unknown function by the Taylor series around the last observation t up to a finite order. As illustrated in Fig. 1, we first map the observed frames into a learned embedding space $\mathcal{H}_t \in \mathcal{H}$ and then estimate each term of the Taylor expansion $\mathcal{F}_{\mathcal{H}_t}$. An important aspect of the proposed TaylorSwiftNet is that it learns to generate a full Taylor expansion from a discrete set of observations, i.e., after training, the network is capable of inferring a Taylor expansion of an unknown function only from a set of observations until

point t. We validate the capabilities of the network using simulated data of known functions where the ground-truth Taylor expansion is known as well for high-dimensional problems like video frame forecasting as shown in Fig. 1. Since the network infers a full function over time $\mathcal{F}_{\mathcal{H}_t} : \mathbb{R} \mapsto \mathcal{H}$, we can evaluate the function for any real value $t+\tau \in \mathbb{R}$. For instance, we can swiftly generate frames for $\tau = 1, 1.5, 4$ as in Fig. 1. Since the predictions are in the embedding space, the predictions are mapped back to the image domain by the decoder. The entire TaylorSwiftNet consisting of the encoder, the estimation of the Taylor expansion, and the decoder is trained end-to-end.

We compare the proposed approach with state-of-the-art methods on four datasets from different application domains ranging from video frame forecasting to forecasting sea surface temperature. On all datasets, TaylorSwiftNet outperforms RNN-based as well as PDE-based approaches. We furthermore demonstrate the capabilities of the approach, namely estimating Taylor expansions of analytical functions and the flexibility to forecast frames with a higher frame rate than the observed data.

2 Related Work

Several approaches have been proposed for video forecasting using unlabeled videos. In particular, deep neural networks have shown promising results for this task. [28, 34, 37] use optical flow to model changes of temporal dynamics and to better predict the future. Some works [14, 23, 56] focus on modeling the geometric transformations between frames to predict the future frames. [13] propose to solve the video prediction task by estimating and using the transformations of the signal in the frequency domain. [6, 48] focus on improving the sharpness of the predicted frames by using custom loss functions. To better handle the future uncertainty and generate sharp predictions, [10, 25, 55, 49] have used generative adversarial networks or variational autoencoders. Methods based on 2D or 3D CNNs [3, 16, 55, 49] have also been proposed. In particular recurrent neural networks (RNNs) have been popular [14, 32, 36, 45, 50, 51, 52, 53, 55] in recent years.

While these approaches learn the motion implicitly, recently a new line of work appeared that leverages partial differential equations (PDEs) and neural networks for forecasting video frames [1, 18, 30, 31, 39, 41]. Some recent PDE-based works demonstrate substantial improvements compared to recurrent neural networks for video frame forecasting. [1, 18, 40, 57] define the dynamics using learned ordinary differential equations following [8]. [15, 27, 43] employ differential equations for stochastic data. Some methods shape the prediction function or the cost function of their methods using prior physical knowledge [2, 7]. For instance, [9] uses general advection-diffusion principles as a guideline for designing a network. [2, 42, 44] discover the PDEs by sparse regression of potential differential terms. [5, 17, 46] introduce non-regression loss functions inspired by Hamiltonian mechanics [19]. [12, 58] have designed specific architectures for predicting and identifying dynamical systems inspired by numerical schemes for solving PDEs and residual neural networks [4, 26, 33, 59]. [11] proposes the separation of variables as a general paradigm based on a resolution method for partial differential equations for video prediction and disentanglement. PDE-Net [30, 31] discretizes a broad class of PDEs by approximating partial derivatives with convolutions.

Although partial differential equations model the motion in the continuous time space, the PDE-based approaches discretize the PDEs using, for instance, the forward Euler method. In this work, we propose a different continuous representation that does not need any dis-

cretization.

3 Forecasting Future Frames

The problem of forecasting future frames can be formulated as

$$p(x_{t+1}|\mathcal{X}_t) \quad \text{for} \quad \mathcal{X}_t = \{x_{t-k}, \dots, x_t\}, \tag{1}$$

where the probability of a future frame x_{t+1} is conditioned on the past k observed frames. For forecasting with a different temporal step, i.e., $\tau > 1$, $\tau \in \mathbb{N}$, this results in

$$p(x_{t+\tau}|\mathcal{X}_t) = \int_{x_{t+\tau-1}} \cdots \int_{x_{t+1}} p(x_{t+\tau}|x_{t+\tau-1}, \mathcal{X}_t) \dots$$
$$p(x_{t+2}|x_{t+1}, \mathcal{X}_t) p(x_{t+1}|\mathcal{X}_t) dx_{t+\tau-1} \dots dx_{t+1}.$$

When the motion is complex and $x_{t+\tau}$ is high-dimensional, as it is the case for video frame forecasting, the computation of the integrals is infeasible. Common auto-regressive approaches therefore approximate the solution frame-by-frame by taking the argmax of (1) and adding the new estimate x_{t+1} to the observations. This, however, has the disadvantage that one needs to iterate over all frames until $x_{t+\tau}$ is reached, the approximation error increases over time, and τ is constrained by the frame-rate of the training and observed data.

To overcome these issues, we propose to learn a mapping from the space of observations $\mathcal{X}$ to the space of infinitely differentiable functions $\mathcal{F} : \mathbb{R} \mapsto \mathcal{X}$. Note that we learn the mapping to a function space and not the Euclidean space $\mathcal{X}$. During inference, the learned mapping maps a given observation $\mathcal{X}_t$ to a continuous forecasting model $\mathcal{F}_{\mathcal{X}_t}$, which can then be evaluated for any $\tau \in \mathbb{R}_{>0}$:

$$x_{t+\tau} = \mathcal{F}_{\mathcal{X}_t}(t+\tau) \tag{2}$$

as it is illustrated in Fig. 1. This means that we can forecast a frame in a more distant future directly without forecasting all intermediate frames. It allows forecasting all future frames in parallel instead of forecasting them sequentially. Finally, we can even forecast at super temporal resolution, i.e., at a higher frame-rate than the observations, without the need to re-train the model.

4 Temporal Dynamics Modeling Through Taylor Series

As discussed, our novel approach for continuous forecasting can directly predict $\hat{x}_{t+\tau}$ for any value $\tau \in \mathbb{R}_{>0}$ from the observations $\mathcal{X}_t$. Since using the observed frames $\mathcal{X}_t$ directly is not practical, we embed the observed frames in an embedding space $\mathcal{H}_t$ and infer the continuous motion model (2) in the embedding space, i.e., $\mathcal{F}_{\mathcal{H}_t} : \mathbb{R} \mapsto \mathcal{H}$ where $\mathcal{F}_{\mathcal{H}_t}$ is infinitely differentiable and smooth. As illustrated in Fig. 2, the entire model thus consists of three parts that are learned end-to-end:

$$\hat{x}_{t+\tau} = \mathcal{D}(h_{t+\tau}) \, ; \, h_{t+\tau} = \mathcal{F}_{\mathcal{H}_t}(t+\tau) \, ; \, \mathcal{H}_t = \mathcal{E}(\mathcal{X}_t). \tag{3}$$

$\mathcal{E}(.)$ first maps the observed frames $\mathcal{X}_t$ into the learned embedding space. Given the embedding of video frames represented by $\mathcal{H}_t$, our network infers an observation specific function
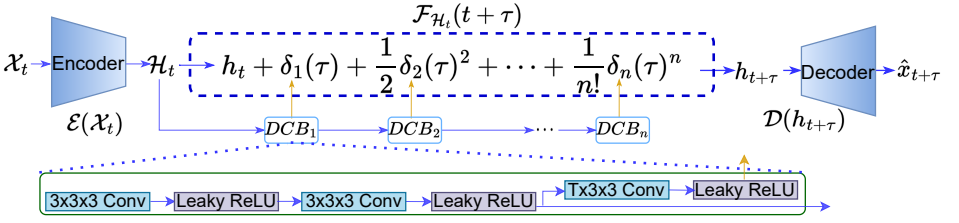
Figure 2: TaylorSwiftNet. The network gets a sequence of frames $\mathcal{X}_t \in \mathbb{R}^{C \times T \times H \times W}$ as input. The encoder maps all frames into a latent space. The frames in the latent space are denoted by $\mathcal{H}_t \in \mathbb{R}^{C' \times T \times H' \times W'}$ and $h_t \in \mathbb{R}^{C' \times H' \times W'}$ denotes the embedding for frame t. Using a Taylor series of order n, the function $\mathcal{F}_{\mathcal{H}_t}(t + \tau)$ is approximated to model the temporal dynamics. To estimate the higher order terms of the Taylor series, $\mathcal{H}_t$ is fed to the DC blocks. The DC blocks estimate sequentially the δ_is, which are the estimated derivatives of $\mathcal{F}_{\mathcal{H}_t}$ at the point t. Having the derivatives δ_i, the Taylor series approximates $\mathcal{F}_{\mathcal{H}_t}(t + \tau)$ at the future time step τ, which yields $h_{t+\tau}$. Finally, $h_{t+\tau}$ is fed to the decoder and the future frame $\hat{x}_{t+\tau}$ is predicted.

$\mathcal{F}_{\mathcal{H}_t}(.)$ that models the future dynamics in the embedding space. It is important to note that our approach infers a full function with respect to τ and not a single point estimate as it is done by standard networks. The function $\mathcal{F}_{\mathcal{H}_t}(.)$ can therefore be swiftly evaluated for any value. $\mathcal{D}(.)$ finally decodes the forecast embedding $h_{t+\tau}$ and predicts the future frame $\hat{x}_{t+\tau}$. While $\mathcal{E}(.)$ and $\mathcal{D}(.)$ will be discussed in Section 5, we first discuss $\mathcal{F}_{\mathcal{H}_t}$.

Our goal is to learn the unknown continuous function $\mathcal{F}_{\mathcal{H}_t}$ that represents the future dynamics in the embedding space. In case of $\tau = 0$, $\mathcal{F}_{\mathcal{H}_t}(t) = h_t$, which is equal to the last vector of $\mathcal{H}_t$. For $\tau > 0$, however, this function is very complex. We therefore propose to approximate the function using the Taylor series:

$$\mathcal{F}_{\mathcal{H}_t}(t + \tau) \simeq \sum_{n=0}^{\gamma} \frac{\mathcal{F}_{\mathcal{H}_t}^{(n)}(t)}{n!} \tau^n \,; \, \mathcal{F}_{\mathcal{H}_t}^{(n)} = \frac{\partial^n \mathcal{F}_{\mathcal{H}_t}}{\partial \tau^n}. \tag{4}$$

If $\mathcal{F}_{\mathcal{H}_t}(t + \tau)$ is an analytic function, the Taylor series with $\gamma = \infty$ is equal to $\mathcal{F}_{\mathcal{H}_t}(t + \tau)$. In practice, we approximate it by using only a finite number of terms.

Although the approximation (4) requires to compute $\mathcal{F}_{\mathcal{H}_t}^{(n)}$ for higher order terms at t in order to get a good approximation, this needs to be done only once and the function can be evaluated for any $\tau > 0$. Since computing the derivatives of an unknown function in a very high dimensional space is impractical, we propose to learn a network that infers them from $\mathcal{H}_t$:

$$\mathcal{F}_{\mathcal{H}_t}^{(n)}(t) \simeq f_n \left(\Delta_n \left(\mathcal{H}_t^{(n-1)} \right) \right) \,; \, \mathcal{H}_t^{(n-1)} = \Delta_{n-1} \left(\mathcal{H}_t^{(n-2)} \right) \tag{5}$$

where f_n and Δ_n are trainable blocks that will be described in Section 5 and $\mathcal{H}_t^{(0)} = \mathcal{H}_t$. Considering (4) and (5), we can reformulate (3) as:

$$\hat{x}_{t+\tau} = \mathcal{D} \left(h_t + \sum_{n=1}^{\gamma} \frac{f_n(\Delta_n(\Delta_{n-1}(\dots \Delta_1(\mathcal{E}(\mathcal{X}_t)))))}{n!} \tau^n \right). \tag{6}$$

Note that the γ terms of the Taylor series are computed recursively, but only once for a given observation. All future frames $t + \tau$ can be predicted in parallel as it is illustrated in Fig. 1.

Having the future frame $x_{t+\tau}$ as ground-truth during training, we can compute the loss between the predicted frame $\hat{x}_{t+\tau}$ and the ground-truth frame $x_{t+\tau}$ and update the parameters of $\mathcal{D}$, f_n, Δ_n, and $\mathcal{E}$.

5 Proposed Architecture

We now describe the network architecture that learns the model formulated in (6). As illustrated in Fig. 2, the network first encodes the input video frames, i.e., $\mathcal{H}_t = \mathcal{E}(\mathcal{X}_t)$. Then the temporal model approximates the function $\mathcal{F}_{\mathcal{H}_t}(t+\tau)$, and finally, we only need to apply the decoder $\mathcal{D}$ to the output of the temporal model $\mathcal{F}_{\mathcal{H}_t}(t+\tau)$ to generate images for all relevant positive values τ in parallel, as illustrated in Fig. 1.

Encoder. The encoder $\mathcal{E}(\mathcal{X}_t)$ maps the input video frames $\mathcal{X}_t \in \mathbb{R}^{C \times T \times H \times W}$ to $\mathcal{H}_t \in \mathbb{R}^{C' \times T \times H' \times W'}$. C, T, H, W are the video channels, number of frames, height, and width of the frames. C', H', W' are the channels of the feature maps and their height and width. We have used a modified version of 3DResNet [20, 21] for the encoder, which is described in the supplemental material, but any 3D convolutional neural network can be used in principle.

Temporal Model. The temporal model $\mathcal{F}_{\mathcal{H}_t}(t+\tau)$ models the future temporal dynamics and forecasts the frame at the future temporal step $t+\tau \in \mathbb{R}$ in the embedded space $h_{t+\tau} \in \mathbb{R}^{C' \times H' \times W'}$. As illustrated in Fig. 2, we estimate $\mathcal{F}_{\mathcal{H}_t}^{(n)}$ (5) recursively. We use for each $\Delta_n(.)$ a convolutional block called delta convolutional block (DCB). The first 2 convolutional layers of DCB use kernels with size $3 \times 3 \times 3$ and stride $1 \times 1 \times 1$. The input and output size remains the same.

The output feature map is then fed to: (a) the final convolutional layer f_n to output the estimated derivative $\delta_n \in \mathbb{R}^{C' \times H' \times W'}$; (b) the next DC block to estimate the next order derivative δ_{n+1}. f_n uses kernels with size $T \times 3 \times 3$. The estimated derivatives are then used in (5) to model the temporal dynamics and forecast the embedding $h_{t+\tau} \in \mathbb{R}^{C' \times H' \times W'}$. While in our experiments the different DC blocks do not share their weights, we also evaluate a recurrent version with shared weights in the supplemental material.

Decoder. The decoder is a convolutional neural network that consists of 6 convolutional layers with kernel size $1 \times 3 \times 3$. The decoder $\mathcal{D}(h_{t+\tau})$ decodes the embedding and predicts the future frame $\hat{x}_{t+\tau} \in \mathbb{R}^{C \times H \times W}$. For more details regarding the implementation details of the decoder, we refer to the supplemental material.

6 Experiments

6.1 Datasets and Evaluation Metrics

Following the state-of-the-art [18], we also evaluate our method on four datasets from very different domains, namely Moving MNIST [45], Human 3.6M [22], Traffic BJ [53], and Sea Surface Temperature [8].

Moving MNIST is a standard dataset for sequence prediction which consists of two random digits moving inside a 64×64 grid. Data for training were generated on the fly and a test set of 10,000 sequences was used for evaluation. We predict 10 unseen future frames given 10 seen input frames.

Human 3.6M contains human actions with their corresponding 3D poses for 17 action scenarios. Following the setting of [18, 53], we select subjects S1, S5, S6, S7, and S8 for

Method	Moving MNIST			Traffic BJ			Sea Surface Temperature			Human 3.6M		
	MSE ↓	MAE ↓	SSIM ↑	MSE ×100	MAE	SSIM	MSE ×10	MAE	SSIM	MSE /10	MAE /100	SSIM
Advection-diffusion [■]	-	-	-	-	-	-	34.1	54.1	0.966	-	-	-
DDPAE [■]	38.9	90.7	0.922	-	-	-	-	-	-	-	-	-
ConvLSTM [■]	103.3	182.9	0.707	48.5	17.7	0.978	45.6	63.1	0.949	50.4	18.9	0.776
PredRNN [■]	56.8	126.1	0.867	46.4	17.1	0.971	41.9	62.1	0.955	48.4	18.9	0.781
Causal LSTM [■]	46.5	106.8	0.898	44.8	16.9	0.977	39.1	62.3	0.929	45.8	17.2	0.851
MIM [■]	44.2	101.1	0.910	42.9	16.6	0.971	42.1	60.8	0.955	42.9	17.8	0.790
E3D-LSTM [■]	41.3	86.4	0.920	43.2	16.9	0.979	34.7	59.1	0.969	46.4	16.6	0.869
PhyDNet [■]	24.4	70.3	0.947	41.9	16.2	0.982	31.9	53.3	0.972	36.9	16.2	0.901
SimVP [■]	23.8	68.9	0.948	41.4	16.2	0.982	-	-	-	31.6	**15.1**	0.904
TaylorSwiftNet (ours)	**17.8**	**42.5**	**0.965**	**35.3**	**13.7**	**0.992**	**29.8**	**52.2**	**0.978**	**23.1**	15.8	**0.910**

Table 1: Comparison to the state-of-the-art on four datasets. The results are the mean over all predicted frames.

training and subjects S9 and S11 for testing using the walking action. Human 3.6M includes originally RGB images of size $1000 \times 1000 \times 3$ which we resize to $128 \times 128 \times 3$ for our experiments. We predict 4 unseen frames given 4 input seen frames.

Traffic BJ contains the hourly taxi flows of Beijing in a 32×32 grid. Each frame has two channels corresponding to the traffic flow entering and leaving a district. We use 4 input seen frames to predict 4 unseen frames.

Sea Surface Temperature consists of meteorological data of the Atlantic ocean generated by NEMO (Nucleus for European Modeling of the Ocean), which is a state-of-the-art simulation engine for modeling ocean dynamics. Following the protocol of [8], we use the Sea Surface Temperature (SST) data of 64×64 sized sub-regions extracted from the original 481×781 sized data. We predict 4 future frames given 4 unseen input frames.

Evaluation Metrics. Following the state-of-the-art methods [18, 30, 51], we use the following evaluation metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), and the Structural Similarity (SSIM) [54]. We average the metrics over all frames of the predicted output sequence. While lower MSE and MAE indicate better performance, a higher SSIM is better.

If not otherwise specified, we use a Taylor model of order 4 for the Moving MNIST dataset and of order 2 for the other datasets. We refer to the supplemental material for more implementation details.

6.2 Comparison to State-of-the-Art

We compare our TaylorSwiftNet with various state-of-the-art methods. As it can be seen in Table 1, TaylorSwiftNet significantly outperforms the state-of-the-art methods on all datasets and for all metrics. Only the very recent CNN-based approach SimVP [16] achieves a slightly lower MAE on Human3.6M. Our direct future temporal forecasting method significantly outperforms state-of-the-art architectures based on PDEs or recurrent neural networks such as PhyDNet [18], ConvLSTM [55], PredRNN [50], Causal LSTM [51], or Memory in Memory (MIM) [53].

For the Human 3.6M dataset, the approach [47] uses additional supervision like human poses. Even without this additional supervision, our approach achieves a Peak Signal over Noise Ratio (PSNR) of 25.75 while [47] reports in the appendix for time step 4 a PSNR below 21 and around 22 for the sequences with the least human motion. The results demonstrate that the proposed approach learns a very good temporal model from various videos of different application domains.

Fig. 3 shows some qualitative results for all datasets. The qualitative results demonstrate the high quality of the forecast results that are generated by the proposed TaylorSwiftNet.
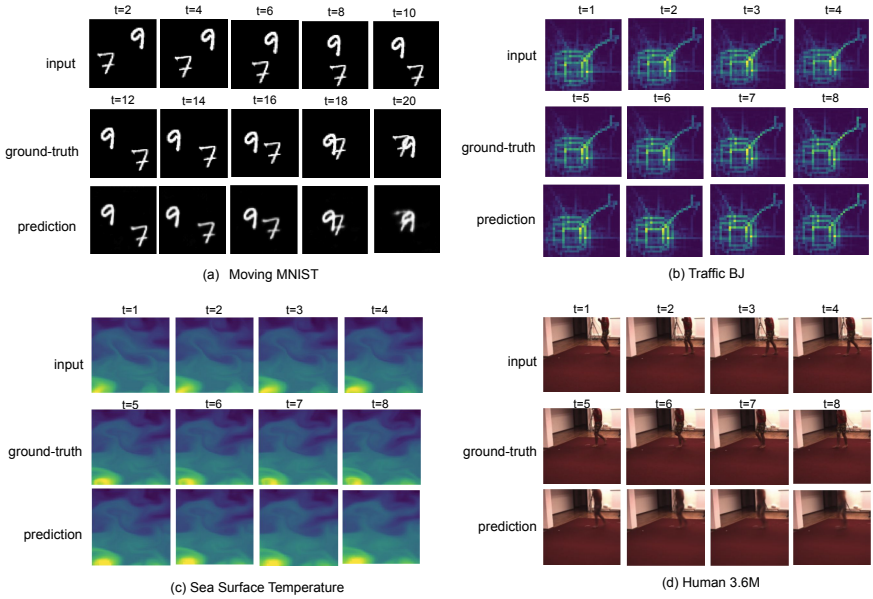
Figure 3: Qualitative results. For each dataset, the first row shows the input of the model, the second row shows the ground-truth, and the third row shows our prediction.

For Moving MNIST, TaylorSwiftNet predicts accurately the digits even when they overlap. For Traffic BJ, the hourly taxi flows are correctly predicted. The sea surface temperature, which depends on phenomena that can be described by PDEs, is accurately predicted as well. And finally, TaylorSwiftNet also precisely anticipates the future position and pose of the person in the video from the Human 3.6M dataset.

6.3   Ablation Experiments

For the ablation studies, we compare the approach to different baselines, analyze the impact of the order of the Taylor series (γ), the performance for long-term forecasting, and the ability to forecast at a higher frame-rate than the observation without re-training the model. Furthermore, we analyze the accuracy of the learned terms of the Taylor expansion in the supplemental material.

Comparison to Baselines.   To demonstrate that the accuracy is due to the proposed model that infers a continuous function over τ from the observations and not due to the encoder and decoder, we compare our approach with two variants that use the same encoder and decoder. However the baseline models use τ as a conditioning variable, i.e., they use τ as input and generate a point estimate for a single τ.

The *Flatten* approach flattens the given hidden embedding $\mathcal{H}_t$ using three convolutional layers with kernel size of $3 \times 3 \times 3$, concatenates it to v_τ and afterwards up-samples it to obtain $h_{t+\tau}$ by using three transposed convolutional layers with kernel size of $1 \times 3 \times 3$. The *Expand* approach expands the vector v_τ to the tensor $C' \times H' \times W'$ and concatenates it with the tensor from $\mathcal{H}_t$ which is temporally squeezed. By concatenating these two tensors channel-wise, we get a $2C' \times H' \times W'$ tensor. With one convolutional layer, we can down-sample the channels to estimate $h_{t+\tau}$. Both approaches are visualized in the supplemental

Method	MSE	MAE	SSIM
Point Estimate (Expand)	45.1	69.4	0.642
Point Estimate (Flatten)	43.6	63.6	0.887
Numerical Derivatives	23.1	46.3	0.956
TaylorSwiftNet	**17.8**	**42.5**	**0.965**

Table 2: Comparison of the proposed temporal model to three variants on Moving MNIST.

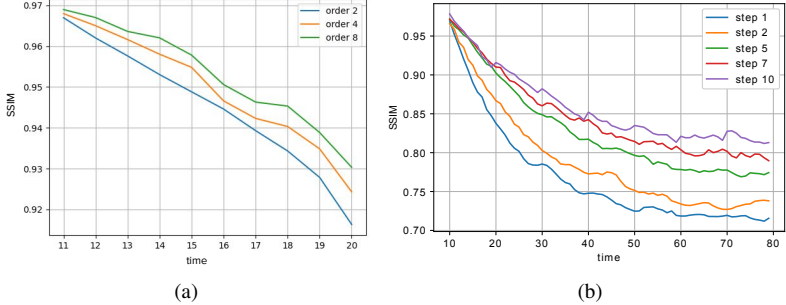(a)   (b)

Figure 4: (a) Comparing different orders of our temporal model using 10 frames as observation to predict the next 10 frames. (b) Comparison of future predictions for time horizons that are larger than the 10 frames used for training. The step size indicates after how many frames the Taylor approximation is performed. Step size x means that our model forecasts x frames, adds the forecast frames to the observations, and continues to predict the next x frames. For both plots, we reduced the number of training epochs compared to the other experiments.

material. In addition, we evaluate TaylorSwiftNet using numerical derivatives instead of DC blocks. We numerically calculate the derivatives for the Taylor terms over the embeddings, i.e., $\delta_1 = h_t - h_{t-1}$ and $\delta_2 = \delta_1 - (h_{t-1} - h_{t-2})$ where δ_1 and δ_2 are the first and second order derivatives. Even in this setup, we train the model end-to-end. The results in Table 2 show that learning a function for all values of τ performs better than adding τ as input to the network and that the DC blocks perform better than the numerical derivatives.

Impact of γ. We approximate $\mathcal{F}_{\mathcal{H}_t}(t + \tau)$ (4) by γ terms where γ defines the order of the Taylor series. We therefore evaluate the effect of using different orders of the Taylor series on Moving MNIST. As it can be seen in Fig. 4a, all three models have approximately the same prediction performance for τ=1 (time step 11). However, as we increase τ the difference between the prediction performance of the 3 models increases. This is expected since having higher orders of the series will result in a better approximation of $\mathcal{F}_{\mathcal{H}_t}(t + \tau)$.

Forecasting at Different Temporal Resolutions. To evaluate the capability of our model for forecasting at a higher temporal resolution than the observations, we train our model on Moving MNIST but we sample only every second frame, i.e., we use 5 out of 10 frames as observation and 5 out of 10 frames for prediction during training. For inference, we also sample 5 out of 10 frames as observation, but we aim to forecast all 10 frames. With our model, we can do this directly by generating the frames for $\tau = 0.5, 1, 1.5, \ldots, 5$ and we compare the forecast frames to the 10 ground-truth frames. Note that the model is trained on the reduced temporal resolution (0.5x). Therefore, $\tau = 5$ corresponds to the future frame 10 of the original sequence. The results are reported in Table 3. For comparison, we generate the frames only for $\tau = 1, 2, \ldots, 5$ as during training and linearly interpolate between the frames to get 10 frames. We also compare our approach to the state-of-the-art approach

Temporal Resolution	TaylorSwiftNet				SimVP [16]	
	Observed 0.5x	Future 1x	Observed 0.5x	Future 1x (Interp.)	Observed 0.5x	Future 1x (Interp.)
SSIM ↑	**0.819**		0.761		0.779	

Table 3: Results when the model is trained on lower temporal resolution (0.5x). While the observed frames are also sub-sampled, the future frames are predicted at full temporal resolution (1x). In case of 'Interp.', the frames are predicted at lower temporal resolution (0.5x), but upsampled by interpolation. Note that the sampling rate of the state-of-the-art method [16] cannot be changed and a higher frame-rate can only be achieved by interpolation.

SimVP [16]. Since the sampling rate of SimVP cannot be changed after training, we use interpolation as well to get 10 frames. Our model achieves a much higher SSIM compared to approaches that require interpolation, which shows the benefit of a continuous representation that allows forecasting frames at a different temporal resolution than the observations.

Long-term Forecasting. In this experiment, we explore the long-term future forecasting capability of our TaylorSwiftNet. We use the same setup as in the previous experiments for the Moving MNIST dataset, but instead of predicting for the future temporal horizon of 10 frames we predict 70 frames, i.e., evaluating far beyond the prediction range seen during training. We therefore evaluate our model in a partially auto-regressive mode where we directly forecast the first 10 frames and feed them back to predict the next 10 frames. In other words, we do a Taylor approximation every 10 frames. For comparison, we also perform the Taylor approximation every 7, 5, 2, and each frame. The latter is a standard auto-regressive setting.

As it can be seen in Fig. 4b, performing the Taylor approximation every 10 frames performs best for forecasting longer sequences of future frames. In contrast, the accuracy of the standard auto-regressive setting where the frames are predicted frame-by-frame performs worst. The reason for such a fast drop in accuracy is due to the error propagation through the recursive steps. The models with the longer step size need fewer auto-regressive steps while the models with shorter step sizes need more auto-regressive steps. This shows that approaches that forecast frames frame-by-frame suffer from error propagation and are not suitable to forecast longer sequences.

7 Conclusion

In this work, we presented an approach that forecasts future frames by modeling the dynamics in the continuous time space without requiring any discretization. Since the motion can be very complex in a video, we use the Taylor series as the approximation method and train a network to infer the higher order terms of the Taylor series from the observed frames. We evaluated our approach on four datasets from different domains like forecasting human motion, hourly taxi flows, or sea surface temperature. For all datasets, our approach achieves state-of-the-art results. We also demonstrated that our approach is capable of forecasting frames at a higher temporal resolution than the observations. The approach, however, has some limitations. The resolution of the images is low and the predicted images become blurry for long-term predictions. The latter can be alleviated by using an additional adversarial loss to ensure that the images remain sharp.

References

[1] Ibrahim Ayed, Emmanuel de Bézenac, Arthur Pajot, and Patrick Gallinari. Learning the spatio-temporal dynamics of physical processes from partial observations. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3232–3236. IEEE, 2020.

[2] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. In *Proceedings of the National Academy of Sciences (PNAS)*, 2016.

[3] Wonmin Byeon, Qin Wang, Rupesh Kumar Srivastava, and Petros Koumoutsakos. Contextvp: Fully context-aware video prediction. In *European Conference on Computer Vision (ECCV)*, 2018.

[4] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[5] Zhengdao Chen, Jianyu Zhang, Martin Arjovsky, and Léon Bottou. Symplectic recurrent neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[6] Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *International Conference on Machine Learning (ICML)*, 2017.

[7] Filipe de Avila Belbute-Peres, Kevin Smith, Kelsey Allen, Josh Tenenbaum, and J Zico Kolter. End-to-end differentiable physics for learning and control. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7178–7189, 2018.

[8] Emmanuel de Bezenac, Arthur Pajot, and Patrick Gallinari. Deep learning for physical processes: Incorporating prior scientific knowledge. In *International Conference on Learning Representations (ICLR)*, 2018.

[9] Emmanuel de Bezenac, Arthur Pajot, and Patrick Gallinari. Deep learning for physical processes: Incorporating prior scientific knowledge. *Journal of Statistical Mechanics: Theory and Experiment (JSTAT)*, 2019(12):124009, 2019.

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

[11] Jérémie Donà, Jean-Yves Franceschi, Sylvain Lamprier, and Patrick Gallinari. Pde-driven spatiotemporal disentanglement. In *International Conference on Learning Representations (ICLR)*, 2021.

[12] Ronan Fablet, Said Ouala, and Cédric Herzet. Bilinear residual neural network for the identification and forecasting of geophysical dynamics. In *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018.

[13] Hafez Farazi and Sven Behnke. Frequency domain transformer networks for video prediction. In *European Symposium on Artificial Neural Networks (ESANN), Computational Intelligence and Machine Learning*, 2019.

[14] Chelsea Finn, Ian J Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[15] Jean-Yves Franceschi, Edouard Delasalles, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. Stochastic latent residual video prediction. *International Conference on Machine Learning (ICML)*, 2020.

[16] 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 (CVPR)*, pages 3170–3180, June 2022.

[17] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 15379–15389, 2019.

[18] Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[19] William Rowan Hamilton. Vii. second essay on a general method in dynamics. *Philosophical Transactions of the Royal Society of London*, (125):95–144, 1835.

[20] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, 2017.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

[22] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2014.

[23] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[24] Hsieh Jun-Ting, Liu Bingbin, Huang De-An, Fei-Fei Li, and Carlos Niebles Juan. Learning to decompose and disentangle representations for video prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[25] Yong-Hoon Kwon and Min-Gyu Park. Predicting future frames using retrospective cycle gan. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[26] Qianxiao Li, Ting Lin, and Zuowei Shen. Deep learning via dynamical systems: An approximation perspective. *Journal of the European Mathematical Society (JEMS)*, 2019.

[27] Xuechen Li, Ting-Kam Leonard Wong, Ricky TQ Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

[28] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Flow-grounded spatial-temporal video prediction from still images. In *European Conference on Computer Vision (ECCV)*, 2018.

[29] 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 (ICCV)*, pages 1744–1752, 2017.

[30] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. PDE-Net: Learning PDEs from Data. In *International Conference on Machine Learning (ICML)*, 2018.

[31] Zichao Long, Yiping Lu, and Bin Dong. PDE-Net 2.0: Learning PDEs from Data with A Numeric-Symbolic Hybrid Deep Network. In *Journal of Computational Physics (JCP)*, 2019.

[32] Chaochao Lu, Michael Hirsch, and Bernhard Scholkopf. Flexible spatio-temporal networks for video prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[33] Yiping Lu, Aoxiao Zhong, Quanzheng Li, and Bin Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *International Conference on Machine Learning (ICML)*, 2018.

[34] Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[35] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *International Conference on Learning Representations (ICLR)*, 2016.

[36] Marc Oliu, Javier Selva, and Sergio Escalera. Folded recurrent neural networks for future video prediction. In *European Conference on Computer Vision (ECCV)*, 2018.

[37] Viorica Patraucean, Ankur Handa, and Roberto Cipolla. Spatio-temporal video autoencoder with differentiable memory. In *International Conference on Learning Representations (ICLR) Workshop Track*, 2016.

[38] Tong Qin, Kailiang Wu, and Dongbin Xiu. Data driven governing equations approximation using deep neural networks. In *Journal of Computational Physics (JCP)*, 2019.

[39] Maziar Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. In *The Journal of Machine Learning Research (JMLR)*, 2018.

[40] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5320–5330, 2019.

[41] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. In *Science Advances*, 2017.

[42] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. In *Science Advances*, 2017.

[43] Tom Ryder, Andrew Golightly, A Stephen McGough, and Dennis Prangle. Black-box variational inference for stochastic differential equations. In *International Conference on Machine Learning (ICML)*, pages 4423–4432, 2018.

[44] Hayden Schaeffer. Learning partial differential equations via data discovery and sparse optimization. In *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2017.

[45] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International Conference on Machine Learning (ICML)*, 2015.

[46] Peter Toth, Danilo J Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[47] 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 (ICML)*, 2017.

[48] LE Vincent and Nicolas Thome. Shape and time distortion loss for training deep time series forecasting models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[49] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[50] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and S Yu Philip. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[51] 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 (ICML)*, 2018.

[52] 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 (ICLR)*, 2018.

[53] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[54] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004.

[55] 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 (NeurIPS)*, 2015.

[56] Tianfan Xue, Jiajun Wu, Katherine Bouman, and Bill Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[57] Cagatay Yildiz, Markus Heinonen, and Harri Lahdesmaki. Ode2vae: Deep generative second order odes with bayesian neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

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

[59] Mai Zhu, Bo Chang, and Chong Fu. Convolutional neural networks combined with runge-kutta methods. *Neural Computing and Applications*, 2018.