DFDNet: Disentangling and Filtering Dynamics for Enhanced Video Prediction

Lianqiang Gan [1], Junyu Lai [1, 2*], Jingze Ju [1], Lianli Gao [3], Yi Bin [4]

[1] School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, China
[2] Aircraft Swarm Intelligent Sensing and Cooperative Control Key Laboratory of Sichuan Province, China
[3] School of Computer Science and Engineering, University of Electronic Science and Technology of China, China
[4] School of Computer Science and Technology, Tongji University, China
ganlq@std.uestc.edu.cn, laijy@uestc.edu.cn, jujz@std.uestc.edu.cn, lianli.gao@uestc.edu.cn, yi.bin@hotmail.com

Abstract

Videos inherently contain complex temporal dynamics across various spatial directions, often entangled in ways that obscure effective dynamic extraction. Previous studies typically process video spatiotemporal features without disentangling, which hampers their ability to extract dynamic information. Additionally, the extraction of dynamics is disrupted by transient high-dynamic information in video sequences, e.g., noise or flicker, which has received limited attention in the literature. To tackle those problems, this paper proposes the Disentangling and Filtering Dynamics Network (DFDNet). Firstly, to disentangle the interwoven dynamics, DFDNet decomposes the spatially encoded video sequences into lower dimensional sequences. Secondly, a learnable threshold filter is proposed to eliminate the transient high-dynamic information. Thirdly, the model incorporates an MLP to extract the temporal dependencies from the disentangled and filtered sequences. DFDNet demonstrates competitive performance across four chosen datasets, including both low and high-resolution videos. Specifically, on the low-resolution Moving MNIST dataset, DFDNet achieves a **19%** improvement on MSE over the previous state-of-the-art model. On the high-resolution SJTU4K dataset, it outperforms the previous state-of-the-art model by **10%** on the LPIPS metric under similar inference time.

Code — https://github.com/lintureforsub/DFDNet

Intorduction

Video predictive learning is geared towards accurately and efficiently inferring future frames based on the observation of previous frames. This looking-ahead ability for the chaotic real world has attracted significant research interest in weather nowcasting (Shi et al. 2015; Reichstein et al. 2019), human action forecasting (Li, Zhou, and Liu 2019), and autonomous vehicles (Hu, Zhan, and Tomizuka 2018). However, it is a great challenge to build a precise prediction model because it needs to process multi-dimensional data including both spatial and temporal information.

The early prevailing approaches for video predictive learning primarily depend on Recurrent Neural Networks (RNNs), with considerable progress made by redesigning recurrent units (Wang et al. 2017), as well as exploring the attention mechanism (Lee et al. 2021; Chen and Wang 2019). While these methods have demonstrated encouraging results, their efficiencies are constrained by the intrinsic properties of recurrent architecture. Consequently, current mainstream prediction methods discard the recurrent architecture to improve prediction efficiency (Gao et al. 2022; Tan et al. 2023). The spatial encoder of those non-recurrent models processes whole video sequences at once, then extracts temporal-dependent features from them, and finally generates all predicted frames in a one-shot manner. However, the video sequences encompass intertwined temporal dynamics across different directions in space. Without disentangling

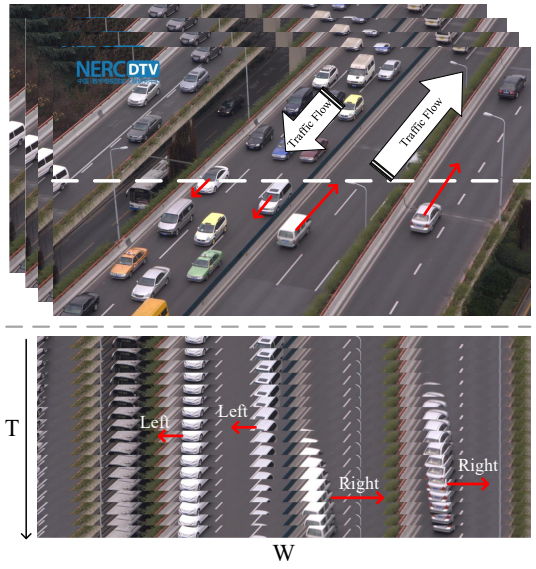

Figure 1: Schematic of temporal dynamic disentanglement. The video depicts a large number of vehicles traveling with different speeds along various directions. The red arrows indicate the direction of the motion and the magnitude of the speed. After disentangling along the white dotted line in the figure, those complex motions is simplified into motion to the left and to the right. This facilitates the model to focus on extracting motions along a certain direction without being disturbed by other directions.

*Corresponding author
Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

the spatially encoded video sequences, the model struggles to extract these intertwined dynamics effectively. In this context, a key problem is to design video prediction methods able to disentangle and extract the intertwined dynamics.

The extraction of dynamic information is recognized as an essential role in video predictive learning. Various methods have been proposed to enhance temporal dynamics. For example, Optical flow (Li et al. 2020) and inter-frame differential (Lin et al. 2020) represent two fundamental approaches to highlight dynamic information in video frames. Some studies utilize Attention Mechanisms to better capture temporal evolution (Tan et al. 2023; Tang et al. 2023; Ning et al. 2023). However, few studies have focused on transient high-dynamic information in video sequences. This type of information manifests as noise, flicker, and other disturbances in videos, characterized by rapid changes and short durations. It is not only challenging to predict but also interferes with the extraction of other dynamic information.

To address the aforementioned issues, we introduce DFD-Net featuring a multilayer architecture and a novel temporal module named DFDBlock. DFDBlock contains SpatioTemporal Aggregation Unit (STAU) and Disentangling and Filtering Dynamics Unit (DFDU). STAU enhances spatiotemporal dependencies for mitigating correlation losses in the process of disentangling interwoven dynamics. DFDU consists of three parts: feature decomposition, learnable threshold filter, and an MLP prediction layer. Firstly, to disentangle the interwoven dynamics, feature decomposition flatten the spatially encoded video sequences into low-dimensional sequences. Specifically, we concatenate the width and height dimensions of the images with the time dimension to project the chaotic motions along the horizontal and vertical directions, as shown in Figure 1. This yields two sets of sequences containing motion information along the horizontal and vertical directions, respectively. Secondly, recognizing the challenges posed by transient high-dynamic information on prediction, we employ fast Fourier transform (FFT) to analyze its frequency spectrum. Transient high-dynamic information corresponds to the high-frequency components with low amplitude in the spectrum. Therefore, a learnable threshold filter is proposed to adaptively eliminate these components. Finally, we use an MLP to extract temporal patterns from the disentangled and filtered sequence. Our contributions can be summarized as follows:

- For dynamic information entangled in various spatial directions,we propose to disentangle these dynamics horizontally and vertically in space. This approach can effectively enhance the model's ability to extract temporal features.
- A spatio-temporal aggregation unit is proposed to reduce information loss caused by disentangling dynamics.
- To address the interference caused by transient high-dynamic information on predictions, we introduce a learnable threshold filter to eliminate it.

Related Work

Deep learning-based video prediction can be categorized into predictive learning and generative learning, each with distinct goals and technical approaches.

Video generative learning models the pixel distributions of video data, subsequently generating more diverse and stochastic prediction frames through conditional distributions. Consequently, powerful generative techniques such as Variational Autoencoders (VAEs) (Lee et al. 2018) and Generative Adversarial Networks (GANs) (Kwon and Park 2019; Liang et al. 2017) have been integrated into video prediction. With the success of Diffusion models in generative learning, numerous studies incorporate them into the realm of video prediction (Ye and Bilodeau 2024). However, the reverse denoising process of diffusion models significantly impedes the generation efficiency. We will use the most advanced video generative learning method based on Diffusion models for comparison with video predictive learning.

Video predictive learning models aim to efficiently and accurately infer the optimal predicted frames based on observations of previous frames. These models are categorized into two types based on architecture: recurrent and non-recurrent model. Recurrent models, as pioneers in video predictive learning, have significantly advanced the field. For example, ConvLSTM (Shi et al. 2015) represents an advancement over traditional LSTM by replacing the fully connected layer with a convolutional layer for the integration of spatial information, which serves as a prominent example for subsequent research endeavors in spatiotemporal prediction. PredRNN (Wang et al. 2017) introduces the ST-LSTM unit by incorporating spatiotemporal memory in ConvLSTM, which facilitates simultaneous memorization of spatial and temporal states. LMC-memory (Lee et al. 2021) proposes a long-term motion context memory that combines the attention mechanism with ConvLSTMs. Swin-LSTM (Tang et al. 2023) proposes a new recurrent cell SwinLSTM by integrating Swin Transformer blocks and the simplified LSTM to increase the efficiency in capturing spatiotemporal dependencies. However, their efficiency has been constrained by the inherent nature of recurrent structures. Hence, non-recurrent models have emerged. For instance, SimVP (Gao et al. 2022) has made a profound impact by handling and predicting videos in a one-shot manner. TAU (Tan et al. 2023) introduces the temporal attention unit to enhance the capability of temporal features extraction. Noteably, these methods directly capture temporal evolution from the spatially encoded video sequences, overlooking the intertwined dynamic information and transient high-dynamic information.

Problem Statement

Video predictive learning by a DNN model can be formulated as follows. Given an input video sequence $X = \{x_i\}, (1 \leq i \leq T)$ of length T frames, we aim to predict the next T' ground truth (GT) frames $Y = \{y_i\}, \ (1 \leq i \leq T')$, where $x_i, y_i \in \mathbb{R}^{C \times H \times W}$ denotes the ith frame. C, H, and W represent as a 3D tensor with dimensions of channel, height, and width, respectively.

Inputting the sequence X into the mapping function Ψ_θ with learnable parameters θ, the DNN model generates the predicted future frames $\hat{Y} = \{\hat{y}_i\}, (1 \leq i \leq T'), \ \hat{y}_i \in$

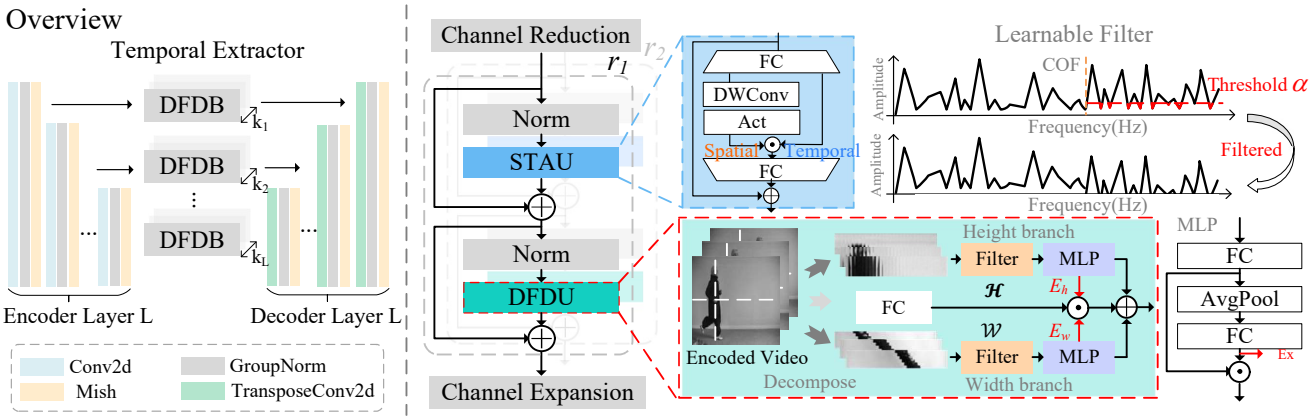

Figure 2: **Left:** The overall structure of DFDNet. It contains three parts: spatial encoder, temporal extractor, and spatial decoder. Each layer of the temporal extractor consists of k_i DFDBs in series; **Right:** The structure of DFDB. DFDB internally contains LayerNorm, STAU (blue dashed box), LayerNorm, and the DFDU (red dashed box). DFDU consists of three pivotal modules: Feature Decomposition, Learnable Threshold Filter, and MLP prediction layer.

$\mathbb{R}^{C\times H\times W}$. The goal is to minimize the error between the predicted frames and the GT by learning the optimal parameters θ^*, which can be obtained by minimizing the loss function $\mathcal{L}$ in the following:

$$\theta^* = \arg\min \mathcal{L}\left\{\Psi_\theta\left(X\right), Y\right\} \tag{1}$$

Methodology

As previously mentioned, our work aims to bridge the gaps in existing video predictive learning models through the introduction of DFDNet. The overall model architecture is illustrated in Figure 2 left. DFDNet based on the architecture proposed by (Lai et al. 2024), introduces a novel DFDBlock, which consists of four steps: 1) aggregating spatiotemporal information; 2) disentangling dynamics; 3) filtering transient high dynamic information; and 4) extracting temporal dependencies. Step 1) is incorporated within STAU, while steps 2), 3), and 4) are integrated within DFDU. They are connected in a residual manner, as shown in Figure 2 right.

Spatial Encoder

The spatial encoder is responsible for encoding video frame sequences into the latent space. The spatial encoder consists of stacked L layers of CNNs. Each CNN layer comprises 3×3 convolution operator, LayerNorm, and *Mish* activation function. The mapping function of encoder is defined as:

$$\mathcal{E}_i\left(\cdot\right) = \sigma\left(GroupNorm\left(Conv\left(\cdot\right)\right)\right), i = 1, 2, \ldots, L. \tag{2}$$

The hidden feature sequence generated by the encoder is:

$$\left\{f_i\right\} = \left\{\mathcal{E}_i\left(f_{i-1}\right)\right\}, i = 1, 2, \ldots, L, \quad f_0 = X, \tag{3}$$

where $X \in \mathbb{R}^{B\times T\times C\times H\times W}$, T and C represent the number of frames and the number of channels of the input video sequence, respectively; $f_i \in \mathbb{R}^{B\times C_{hid}\times H_i\times W_i}$, $C_{hid} = T\times C'$, where C' is the hidden channel, H_i and W_i denote the high and width of the output tensor in ith layer. Then, each output f_i of the CNN layers in the encoder is not only passed vertically to deeper encoder layers for feature extraction but also horizontally input into DFDB for temporal extraction.

SpatioTemporal Aggregation Unit

Prior to inputting into STAU, channel reduction is essential for reducing the computational and memory costs. Assuming $z_i^j \in \mathbb{R}^{B\times C_{hid}\times H_i\times W_i}$ denotes the input tensor of jth DFDBlock in ith layer, we have:

$$z_i^j = ReLU(BatchNorm(Conv_{1\times1}(z_i^j))), z_i^0 = f_i. \tag{4}$$

Subsequently, rearrange the dimensions of z_i^j into $B\times H_i\times W_i\times C_r$, where $C_r = C_{hid}/r$ is the number of reduced channels, and r represents the reduction ratio. In DFDBlock, r can be multiple. Channel reduced variants with different r are processed parallelly and then concatenate together to perform channel expansion.

After spatial encoding, the dynamic information in various spatial directions is correlated. The dynamic disentanglement in step 2) inevitably leads to the loss of some correlation. Therefore, the purpose of STAU is to interact with dynamics in different directions to minimize correlation loss.

Firstly, STAU employs a linear layer with W_{st} to extract dynamic features along the channel dimension. Secondly, spatial features are extracted using depth-wise convolution and multiplied by the channel dynamics. The final linear layer then use weight matrix W_o to apply linear mapping back for the feature channels. The STAU is formalized as:

$$\begin{aligned} s, t &= split(W_{st} z_i^j + b_{st}), \\ z_i^j &= W_o(\sigma(DWConv_{3\times3}(s)) \odot t) + b_o + z_i^j, \end{aligned} \tag{5}$$

where s, t are the spatial variation and the temporal variation, and $\odot$ denotes element-wise multiplication.

Feature Decomposition

Feature Decomposition concatenates the width and height dimensions of the images with the time dimension to project

the chaotic motions along the horizontal and vertical directions. Given the input z_i^j from previous step, we first split it into G groups along the channel dimension, satisfying $C_r = N * G$. We then perform permutation operatios to facilitate the concatenation of the height and width dimensions with the channel dimension. This yields two sets of sequences: $z_i^j \rightarrow \mathcal{H}_i^j \in \mathbb{R}^{B\times W_i G\times(H_i\times C_r/G)}$ and $z_i^j \rightarrow \mathcal{W}_i^j \in \mathbb{R}^{B\times H_i G\times(W_i\times C_r/G)}$. $\mathcal{H}_i^j$ and $\mathcal{W}_i^j$ contain dynamic information along the horizontal and vertical directions, respectively. In the following steps, $\mathcal{H}_i^j$, $\mathcal{W}_i^j$, and z_i^j are independently processed across three distinct pathways, i.e., height branch, width branch, and channel branch. Considering the height branch, the MLP's input dimensions are $H_i \times (C_r/G)$, indicating that it solely applies linear transformations to the last dimension of $\mathcal{H}_i^j$. Consequently, it exclusively captures the dynamics along the height dimension, free from the influence of the dynamics in width dimension.

Learnable Threshold Filter

The transient high-dynamic information in video sequences manifests as as noise, flicker, and other disturbances, characterized by rapid changes and short durations. The Fourier transform reveals that rapidly varying information is represented by high-frequency component in the spectrum. Therefore, it can be deduced that the transient high-dynamic information corresponds to the high-frequency components with low amplitude. Those components generally offers minimal contribution to the precision of predictions and it is challenging to forecast, which may predispose models to overfitting. Hence, a learnable threshold filter has been introduced to selectively eliminate high-frequency components that are below a learnable threshold, effectively filtering the transient high-dynamic information in $\mathcal{H}_i^j$ and $\mathcal{W}_i^j$:

$$\begin{aligned} F = FFT(\mathcal{H}_i^j), F = \{f_1, f_2, \ldots, f_{\frac{L_h}{2}}\}, \\ mask = \underset{f\in\{f_C,\ldots,f_{\frac{L_h}{2}}\}}{torch.ge}(Amp(F), \alpha_h), \\ \mathcal{H}_i^j = IFFT(F * mask). \end{aligned} \tag{6}$$

Here, $FFT(\cdot)$ and $Amp(\cdot)$ denote the FFT and the calculation of amplitude values, tensor $F \in \mathbb{R}^{B\times W_i G\times(L_h/2)}$ is the spectrum, comprising $L_h/2$ frequency points, where $L_h = (H_i \times C_r/G)$. The $torch.ge(\cdot)$ function compares the spectral amplitude with a learnable threshold α_h, to generate a frequency mask. This mask is then used for filtering, followed by an inverse transformation $IFFT(\cdot)$ to obtain the filtered $\mathcal{H}_i^j$; the same principle applies to $\mathcal{W}_i^j$.

MLP Prediction Layer

MLP captures temporal dependency from disentangled and filtered $\mathcal{H}_i^j$ and $\mathcal{W}_i^j$ in previous two steps. Below, the working process of the MLP is illustrated, taking the height branch as an example. The extraction of temporal dependency is simple as we only need a fully-connected layer with $W_1 \in \mathbb{R}^{L_h\times L_h}$ to perform a linear projection with respect to the input $\mathcal{H}_i^j$:

$$\mathcal{H}_i^j = W_1\mathcal{H}_i^j + b_1 + \mathcal{H}_i^j. \tag{7}$$

Subsequently, we reshape $\mathcal{H}_i^j$ into tensor of shape $B \times H_i \times W_i \times C_r$, denoted as h_i^j. After performing average pooling on h_i^j, the second fully connected layer with $W_2 \in \mathbb{R}^{C_r\times C_r}$ carries out a linear transformation on h to obtain the excitation E_h:

$$E_h = W_2 AvgPool(h_i^j, 1) + b_2, \tag{8}$$

where $E_h \in \mathbb{R}^{B\times 1\times W_i\times C_r}$. The same operation is performed on $\mathcal{W}_i^j$ to obtain w_i^j and $E_w \in \mathbb{R}^{B\times H_i\times 1\times C_r}$. Finally, excitation operations are carried out:

$$\begin{aligned} c_i^j &= W_3 z_i^j + b_3, \\ z_i^j &= h_i^j \odot E_h + w_i^j \odot E_w + c_i^j(E_h + E_w). \end{aligned} \tag{9}$$

Here, c_i^j represents the channel dynamics, which receives the sum of the excitations from both the width and height branches. $W_3 \in \mathbb{R}^{C_r\times C_r}$ is the weight matrix of the fully connected layer in Channel branch.

The resultant tensor z_i^j necessitates a channel expansion to augment the channel count from C_r back to C_{hid}, thereby aligning with the input size for consistency:

$$z_i^{j+1} = ReLU(BatchNorm(Conv_{1\times1}(z_i^j))), z_i^k = f_i^{'}, \tag{10}$$

where, z_i^{j+1} denotes the input tensor to the subsequent $(j+1)$th DFDBlock in ith layer, k_i signifies the number of DFDBlock stacks at the ith layer, and $f_i^{'}$ represents the output of the ith layer in the temporal extractor.

Spatial Decoder

The decoder is responsible for mapping the processed spatiotemporal features back to predicted video sequences. The decoder requires L layers of CNNs to receive and decode L different depth features, enabling multi-scale feature fusion. Similar to the encoder, each layer of the decoder consists of 3×3 transpose convolution, LayerNorm, and *Mish* activation function. The mapping function of the decoder is:

$$\mathcal{D}_i(\cdot) = \sigma(GroupNorm(unConv(\cdot))), i = 1, 2, \ldots, L. \tag{11}$$

The decoder fuses these processed L layer features $f_i^{'}$ to generate predicted video frame sequences:

$$\begin{aligned} {f'}_{i-1} &= \mathcal{D}_i({f'}_i) + {f'}_{i-1}, i = 2, \ldots, L, \\ \hat{Y} &= \mathcal{D}_1({f'}_1), \qquad i = 1, \end{aligned} \tag{12}$$

where $\hat{Y}$ is the predicted frame sequence.

Experiments

Datasets

In this section, we present experiments to demonstrate the effectiveness of our proposed DFDNet on various datasets with different resolution including **Moving MNIST** (resolution 64×64) (Srivastava, Mansimov, and Salakhudinov 2015), **KTH** (resolution 128×128) (Schuldt, Laptev, and Caputo 2004), **Human3.6M** (resolution 1024×1024) (Ionescu et al. 2013), and **SJTU4K** (resolution 2160×3840) (Chang et al. 2022). All these datasets are commonly used in existing literature. More details about those datasets and experiment configurations please refer to the Appendix.

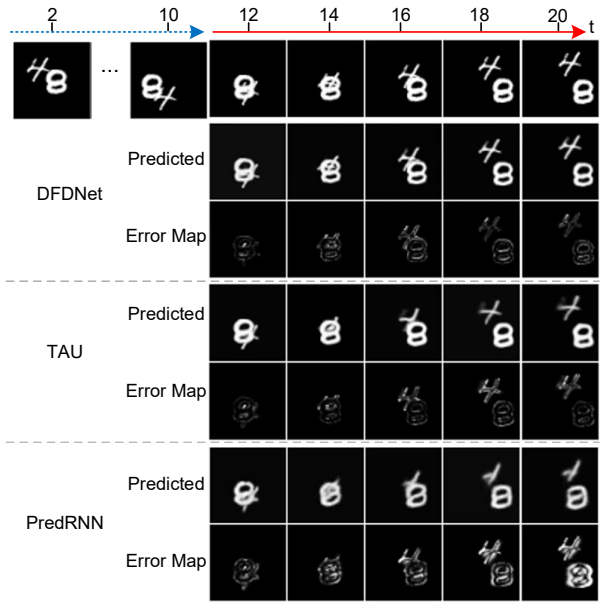

Figure 3: Qualitative visualization of predicted results on Moving MNIST (10→10 frames).

Quantitative and Qualitative Comparison

Results on Moving MNIST. Following previous works (Gao et al. 2022; Tan et al. 2023), DFDNet is trained to predict the future 10 frames based on observing the previous 10 frames. We evaluate DFDNet on the Moving MNIST dataset, and compare it with twelve baselines, including ConvLSTM (Shi et al. 2015), PredRNN (Wang et al. 2017), PredRNNv2 (Wang et al. 2022), E3D-LSTM (Wang et al. 2018b), PhyDnet (Guen and Thome 2020), CrevNet (Yu et al. 2019), SimVP (Gao et al. 2022), TAU (Tan et al. 2023), FFINet (Li, Zhang, and Xu 2024), SwinLSTM (Tang et al. 2023), MIMO-VP (Ning et al. 2023) and SIAM (Zheng et al. 2024). The quantitative comparisons are reported in Table 1, and the qualitative visualizations of predictive results are shown in Figure 3. DFDNet performs the best on MSE, MAE, and SSIM. Particular, DFDNet outperforms the SOTA model by 19.65% on the MSE metric.

Results on KTH. Following (Gao et al. 2022), DFDNet is trained to generate the subsequent 20 or 40 frames based on the previous 10 observations. We choose 11 representative baseline models for comparison, including ConvLSTM (Shi et al. 2015), DFN (Jia et al. 2016), MMVP (Zhong et al. 2023), ExtDM (Zhang et al. 2024), PredRNN (Wang et al. 2017), PredRNNv2 (Wang et al. 2022), E3d-LSTM (Wang et al. 2018b), STMFANet (Jin et al. 2020), SimVP (Gao et al. 2022), and TAU (Tan et al. 2023). Table 2 report the results on SSIM, PSNR, and LPIPS with output length of 20 and 40 for comparison to the mentioned models. It can be observed that DFDNet surpasses all the compared baseline models in SSIM, PSNR, and LPIPS metrics, demonstrating its capability for accurate long-term and variable-length predictions.

Additionally, we further discussed DFDNet and the currently best diffusion-based video prediction model STDiff (Ye and Bilodeau 2024). A comparison of the performance and inference time between STDiff and DFDNet is presented in the Table 3. Note that the generative model aims to produce more diverse and stochastic future frames, implying that the model needs to undergo multiple sampling iterations for generation. Although it has a higher generation quality, STDiff is constrained by the reverse denoising and sampling processes, resulting in longer inference times than DFDNet.

Method	MSE↓	MAE↓	SSIM↑
ConvLSTM	103.3	182.9	0.707
PredRNN	56.8	126.1	0.867
PredRNNv2	46.5	106.8	0.898
E3D-LSTM	41.3	86.4	0.910
PhyDNet	24.4	70.3	0.947
SimVP	23.8	68.9	0.948
CrevNet	22.3	-	0.949
TAU	19.8	60.3	0.957
FFINet	19.2	60.4	0.958
SwinLSTM	17.7	-	0.962
MIMO-VP	17.7	51.6	0.964
SIAM	17.3	55.4	0.962
DFDNet	**13.9**	**48.9**	**0.969**

Table 1: Quantitative results of different methods on Moving MNIST dataset (10→10 frames).

	KTH(10→20)			KTH(10→40)		
	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑
ConvLSTM	25.2	24.31	0.712	32.9	22.93	0.639
DFN	-	27.26	0.794	-	23.01	0.652
ExtDM	-	28.53	0.838	-	27.91	0.799
PredRNN	18.4	27.64	0.839	22.4	24.16	0.703
SVAP-VAE	-	27.77	0.852		26.18	0.811
PredRNNv2	16.4	28.48	0.865	21.3	25.21	0.741
E3d-LSTM	-	29.31	0.879	-	27.24	0.810
STMFANet	-	29.85	0.893	-	27.56	0.851
MMVP	-	27.54	0.906	-	26.35	0.888
SimVP	23.4	33.72	0.905	30.0	32.93	0.886
TAU	21.2	34.13	0.911	28.7	33.01	0.892
DFDNet	**12.5**	**35.11**	**0.916**	**18.2**	**33.68**	**0.896**

Table 2: Quantitative results of different methods on the KTH dataset (10→20 frames & 10→40 frames). Lower LPIPS (10^{-2}) and higher PSNR (dB) scores indicate better results.

	FVD↓	Inference time
STDiff	**89.67**	6.56s
DFDNet	94.82	**0.44s**

Table 3: Quantitative results of DFDNet and STDiff on KTH dataset (10→20 frames).

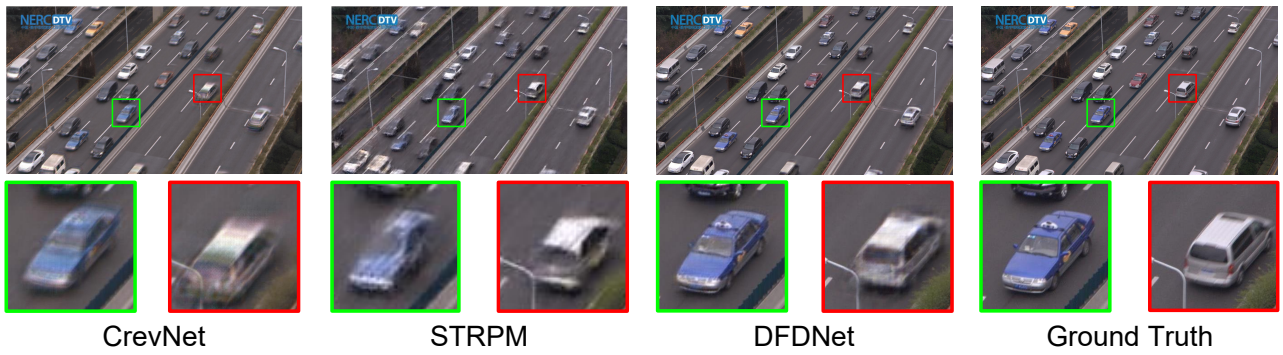

Figure 4: Qualitative visualization of predicted results on SJTU4K (4→1 frames).

	$t=5$	$t=8$
	PSNR↑/LPIPS↓	PSNR↑/LPIPS↓
PredRNN	31.91 / 12.62	25.65 / 14.01
PredRNN++	32.05 / 13.85	27.51 / 14.94
SV2P	31.93 / 13.91	27.33 / 15.02
HFVP	32.11 / 13.41	27.31 / 14.55
E3D-LSTM	32.35 / 13.12	27.66 / 13.95
CycleGAN	32.83 / 10.18	28.26 / 11.03
CrevNet	33.18 / 11.54	28.31 / 12.37
MotionRNN	32.20 / 12.11	28.03 / 13.29
STRPM	33.32 / 9.74	29.01 / 10.44
DFDNet	**35.21 / 7.32**	**29.59 / 9.36**

Table 4: Quantitative results of different methods on Human3.6M dataset (4→4 RGB frames).

	$t=5$	$t=8$	Inference
	PSNR↑/LPIPS↓	PSNR↑/LPIPS↓	Time
ConvLSTM	22.74 / 67.81	17.91 / 86.84	27.67s
PredRNN	23.25 / 66.60	18.20 / 87.04	28.21s
PredRNN++	23.43 / 64.07	18.55 / 86.34	37.14s
SAVP	23.41 / 61.44	18.63 / 80.45	70.39s
CrevNet	24.35 / 62.31	19.61 / 80.91	37.26s
MotionRNN	23.47 / 65.21	19.72 / 81.39	44.19s
STRPM	24.37 / 57.12	19.77 / 66.68	28.06s
DFDNet	**25.44 / 51.41**	**20.31 / 61.93**	**21.12s**

Table 5: Quantitative results of different methods on SJTU4K dataset (4→4 RGB frames).

Therefore, video prediction learning and video generation learning each have their own applicable scenarios.

Results on Human3.6M. Human 3.6M is another human pose dataset similar to KTH but with more complicated backgrounds and higher resolution, which posing difficulties for video prediction. Following (Chang et al. 2022), we resize the resolution to 1024×1024 the train model to predict the following 4 frames given the previous 4 RGB frames. Nine baseline models including PredRNN (Wang et al. 2017), PredRNN++ (Wang et al. 2018a), SV2P (Babaeizadeh et al. 2018), HFVP (Villegas et al. 2019), E3D-LSTM (Wang et al. 2018b), CycleGAN (Kwon and Park 2019), CrevNet (Yu et al. 2019), MotionRNN (Wu et al. 2021), and STRPM (Chang et al. 2022) are selected for comparison. For the quantitative evaluation, we utilize PSNR and LPIPS. Table 4 reports the quantitative performance comparison of DFDNet and the baselines on Human3.6M dataset. DFDNet achieves the state-of-the-art performance under both PSNR and LPIPS metrics.

Results on SJTU4K. SJTU4K is a 4K dataset with ultra-high resolution of 2160×3840, posing challenges for both model training and inference. Following (Chang et al. 2022), the inputs and outputs are all 4K videos without being down-sampled. Seven baseline models including ConvLSTM (Shi et al. 2015), PredRNN (Wang et al. 2017), PredRNN++ (Wang et al. 2018a), SAVP (Lee et al. 2018), CrevNet (Yu et al. 2019), MotionRNN (Wu et al. 2021), and STRPM (Chang et al. 2022) are selected for comparison. All models are trained with 4 previous frames to predict the following 4 frames. The quantitative results are summarized in Table 5. Figure 4 shows the predicted 4K video frames from different methods. DFDNet achieves the best qualitative and quantitative results on 4K videos with a satisfactory inference speed.

Ablation Study

We compare the performance and inference time of our model with state-of-the-art methods in the first several rows in Table 6. The inference time of our proposed model is 43.06ms, which is higher than SimVP but significantly lower than other models. The performance improvement of DFDNet is effective relative to the increased inference time:

To assess the impact of STAU, Feature Decomposition and Learnable Threshold Filter of DFDU, we considered 3 ablation methods and evaluated them on Moving MNIST. The following explains the variants of its implementation:

1. **w/o STAU:** we removed the Spatiotemporal Aggregation Unit from the model.
2. **w/o D:** we removed the Feature Decomposition on the H and W branches, retaining only the C branch.

	MSE↓	SSIM	inference time (ms)
PredRNN	56.8	0.867	120.96
PredRNNv2	46.5	0.898	173.34
PhyDNet	24.4	0.947	60.91
SimVP	23.8	0.948	9.89
CrevNet	22.3	0.949	469.44
Ours w/o STAU	15.3	0.965	34.74
Ours w/o D	41.6	0.896	18.38
Ours w/o F	14.2	0.967	34.02
Ours	**13.9**	**0.969**	43.06

Table 6: An ablation study on Moving MNIST (10→10 frames) shows the influence of the STAU and DFDU.

3. **w/o F:** we removed Learnable Threshold Filters and directly extract temporal dependencies from disentangled features without filtering.

Table 6 the results of the ablation study. Specifically,we summarized the following three improvements:

1. **Improvement of STAU:** After removing the SpatioTemporal Aggregation Unit, the performance of the model showd a significant decrease.
2. **Improvement of Feature Decomposition:** Based on the results of the variant w/o D, it can be infered that Feature Decomposition have significant contribution to prediction performance. This finding suggests that Feature Decomposition disentangled the interwoven dynamics.
3. **Improvement of Learnable Threshold Filter:** The result of variant w/o F indicates that Learnable Threshold Filter can eliminate the transient high-dynamic information as w/o F improving the performance of DFDNet.

Visualization of Feature Decomposition

To validate the effectiveness of Feature Decomposition in DFDU, we randomly selected a sample from the Moving MNIST test set and visualized the feature maps of the height and width branches. Figure 6 presents the results, where blue and red arrows indicate the magnitude of horizontal dynamics and vertical dynamics, respectively. The feature maps of the width branch describe only horizontal dynamics, while the feature maps of the height branch capture only vertical dynamics. The results demonstrate that they have learned the dynamics in the corresponding direction, confirming the validity of dynamic disentanglement.

Visualization of Learnable Threshold Filter

To verify the effectiveness of the Learnable Threshold Filter in DFDU, we visualize the input and output signals of the filter. As shown in the Figure 6, the filter eliminates the high-frequency signal with low amplitude. These signals are irregular and close to noise, posing a significant challenge to prediction. Our proposed filter removes these signals, which is beneficial for enhancing the prediction accuracy.

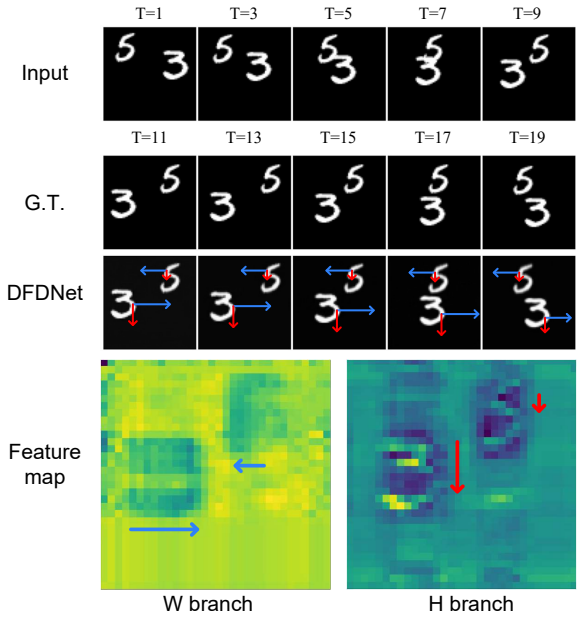

Figure 5: Visualization of Feature Decomposition on Moving MNIST dataset.

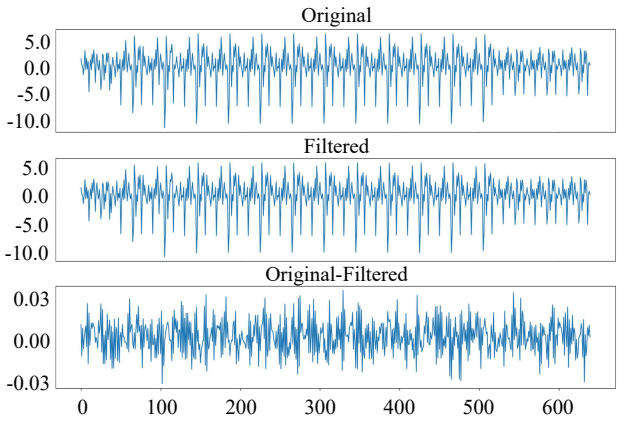

Figure 6: Visualization of Learnable Threshold Filter.

Conclusion

In this paper, we introduce DFDNet to address two critical challenges: the entanglement of dynamics across spatial directions and the interference of transient high-dynamic information for enhanced video prediction. Our model leverages Feature Decomposition and the Learnable Threshold Filter to disentangle interwoven dynamics and eliminate transient high-dynamic information, synergistically improving the model's ability to extract temporal features. Through extensive experiments on various low and high-resolution datasets, we demonstrate that DFDNet outperforms existing models in prediction accuracy and dynamic extraction. Our findings underscore the importance of disentangling and filtering dynamic information in video prediction.

References

Babaeizadeh, M.; Finn, C.; Erhan, D.; Campbell, R.; and Levine, S. 2018. Stochastic variational video prediction. In *6th International Conference on Learning Representations, ICLR 2018*.

Chang, Z.; Zhang, X.; Wang, S.; Ma, S.; and Gao, W. 2022. Strpm: A spatiotemporal residual predictive model for high-resolution video prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13946–13955.

Chen, X.; and Wang, W. 2019. Uni-and-bi-directional video prediction via learning object-centric transformation. *IEEE Trans. Multimedia*, 22(6): 1591–1604.

Gao, Z.; Tan, C.; Wu, L.; and Li, S. Z. 2022. Simvp: Simpler yet better video prediction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 3170–3180.

Guen, V. L.; and Thome, N. 2020. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 11474–11484.

Hu, Y.; Zhan, W.; and Tomizuka, M. 2018. Probabilistic prediction of vehicle semantic intention and motion. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, 307–313. IEEE.

Ionescu, C.; Papava, D.; Olaru, V.; and Sminchisescu, C. 2013. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(7): 1325–1339.

Jia, X.; De Brabandere, B.; Tuytelaars, T.; and Gool, L. V. 2016. Dynamic filter networks. *Adv. Neural Inform. Process. Syst.*, 29.

Jin, B.; Hu, Y.; Tang, Q.; Niu, J.; Shi, Z.; Han, Y.; and Li, X. 2020. Exploring spatial-temporal multi-frequency analysis for high-fidelity and temporal-consistency video prediction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 4554–4563.

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

Lai, J.; Gan, L.; Zhu, J.; Liu, H.; and Gao, L. 2024. Exploring Spatial Frequency Information for Enhanced Video Prediction Quality. *IEEE Transactions on Multimedia*, 1–14.

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

Lee, S.; Kim, H. G.; Choi, D. H.; Kim, H.-I.; and Ro, Y. M. 2021. Video prediction recalling long-term motion context via memory alignment learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 3054–3063.

Li, M.; Zhou, Z.; and Liu, X. 2019. Multi-person pose estimation using bounding box constraint and LSTM. *IEEE Trans. Multimedia*, 21(10): 2653–2663.

Li, P.; Zhang, C.; and Xu, X. 2024. Fast Fourier Inception Networks for Occluded Video Prediction. *IEEE Transactions on Multimedia*, 26: 3418–3429.

Li, S.; Fang, J.; Xu, H.; and Xue, J. 2020. Video frame prediction by deep multi-branch mask network. *IEEE Trans. Circuit Syst. Video Technol.*, 31(4): 1283–1295.

Liang, X.; Lee, L.; Dai, W.; and Xing, E. P. 2017. Dual motion GAN for future-flow embedded video prediction. In *Int. Conf. Comput. Vis.*, 1744–1752.

Lin, X.; Zou, Q.; Xu, X.; Huang, Y.; and Tian, Y. 2020. Motion-aware feature enhancement network for video prediction. *IEEE Trans. Circuit Syst. Video Technol.*, 31(2): 688–700.

Ning, S.; Lan, M.; Li, Y.; Chen, C.; Chen, Q.; Chen, X.; Han, X.; and Cui, S. 2023. MIMO is all you need: a strong multi-in-multi-out baseline for video prediction. In *AAAI*, volume 37, 1975–1983.

Reichstein, M.; Camps-Valls, G.; Stevens, B.; Jung, M.; Denzler, J.; Carvalhais, N.; and Prabhat, f. 2019. Deep learning and process understanding for data-driven Earth system science. *Nature*, 566(7743): 195–204.

Schuldt, C.; Laptev, I.; and Caputo, B. 2004. Recognizing human actions: a local SVM approach. In *Int. Conf. Pattern Recog.*, volume 3, 32–36. IEEE.

Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Adv. Neural Inform. Process. Syst.*, 28.

Srivastava, N.; Mansimov, E.; and Salakhudinov, R. 2015. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, 843–852. PMLR.

Tan, C.; Gao, Z.; Wu, L.; Xu, Y.; Xia, J.; Li, S.; and Li, S. Z. 2023. Temporal attention unit: Towards efficient spatiotemporal predictive learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 18770–18782.

Tang, S.; Li, C.; Zhang, P.; and Tang, R. 2023. SwinLSTM: Improving Spatiotemporal Prediction Accuracy using Swin Transformer and LSTM. In *Int. Conf. Comput. Vis.*, 13470–13479.

Villegas, R.; Pathak, A.; Kannan, H.; Erhan, D.; Le, Q. V.; and Lee, H. 2019. High fidelity video prediction with large stochastic recurrent neural networks. *Advances in Neural Information Processing Systems*, 32.

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

Wang, Y.; Jiang, L.; Yang, M.-H.; Li, L.-J.; Long, M.; and Fei-Fei, L. 2018b. Eidetic 3D LSTM: A model for video prediction and beyond. In *Int. Conf. Learn. Represent.*

Wang, Y.; Long, M.; Wang, J.; Gao, Z.; and Yu, P. S. 2017. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Adv. Neural Inform. Process. Syst.*, 30.

Wang, Y.; Wu, H.; Zhang, J.; Gao, Z.; Wang, J.; Philip, S. Y.; and Long, M. 2022. Predrnn: A recurrent neural network for spatiotemporal predictive learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(2): 2208–2225.

Wu, H.; Yao, Z.; Wang, J.; and Long, M. 2021. MotionRNN: A flexible model for video prediction with spacetime-varying motions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 15435–15444.

Ye, X.; and Bilodeau, G.-A. 2024. STDiff: Spatio-Temporal Diffusion for Continuous Stochastic Video Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 6666–6674.

Yu, W.; Lu, Y.; Easterbrook, S.; and Fidler, S. 2019. Efficient and Information-Preserving Future Frame Prediction and Beyond. In *Int. Conf. Learn. Represent.*

Zhang, Z.; Hu, J.; Cheng, W.; Paudel, D.; and Yang, J. 2024. Extdm: Distribution extrapolation diffusion model for video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19310–19320.

Zheng, X.; Peng, Z.; Cao, Y.; Shan, H.; and Zhang, J. 2024. SIAM: A Simple Alternating Mixer for Video Prediction. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, 1–10. IEEE.

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