Wide and Narrow:
Video Prediction from Context and Motion

Jaehoon Cho[1]
rehoon@yonsei.ac.kr

Jiyoung Lee[1]
easy00@yonsei.ac.kr

Changjae Oh[2]
c.oh@qmul.ac.uk

Wonil Song[1]
swonil92@yonsei.ac.kr

Kwanghoon Sohn[1]
khsohn@yonsei.ac.kr

[1] School of Electrical and
    Electronic Engineering
    Yonsei University
    Seoul, South Korea

[2] Centre for Intelligent Sensing
    Queen Mary University of London
    London, United Kingdom

Abstract

Video prediction, forecasting the future frames from a sequence of input frames, is a challenging task since the view changes are influenced by various factors, such as the global context surrounding the scene and local motion dynamics. In this paper, we propose a new framework to integrate these complementary attributes to predict complex pixel dynamics through deep networks. To capture the local motion pattern of objects, we devise local filter memory networks that generate adaptive filter kernels by storing the prototypical motion of moving objects in the memory. We further present global context propagation networks that iteratively aggregate the non-local neighboring representations to preserve the contextual information over the past frames. The proposed framework, utilizing the outputs from both networks, can address blurry predictions and color distortion. We conduct experiments on Caltech pedestrian and UCF101 datasets, and demonstrate state-of-the-art results. Especially for multi-step prediction, we obtain an outstanding performance in quantitative and qualitative evaluation.

1 Introduction

Video prediction, the task of generating high-fidelity future frames by observing a sequence of past frames, is paramount in numerous computer vision applications such as video object segmentation [57], robotics [6], anomaly detection [25], and autonomous driving applications [2, 4, 13].

Early works addressed the video prediction to directly predict raw pixel intensities from the observations through various deep learning architectures including 2D/3D convolutional neural networks (CNNs) [29, 49], recurrent neural networks (RNNs) [39, 44, 46, 51], and generative adversarial networks (GANs) [21, 24, 29, 49]. These methods have improved the perceptual quality by encoding the spatial contents, but they often overlook the complex
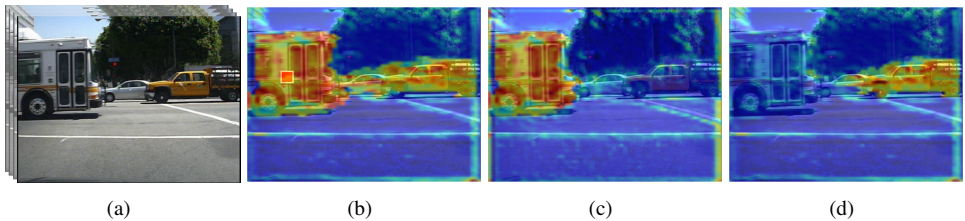
(a)   (b)   (c)   (d)

Figure 1: Examples of intermediate features in our network. Given a video sequence in (a), our model captures two complementary attributes, contextual information and motion dynamics. (b) Attention map for a query location (red point) from the global context propagation networks (GCPN), and (c)–(d) memory attention maps for different memory slots of the local filter memory networks (LFMN) respectively. The attention maps are upsampled to match the image size. Best viewed in color.

motion variations over time. Another approach is to explicitly model the motion dynamics by predicting a dense motion field (*e.g.* optical flow) [23, 24, 25, 26, 37, 40]. The dense motion field ensures the temporal consistencies between frames, but occlusion and large motion may limit precise motion estimation [8].

Humans perceive an entire image, recognize the regional information such as moving objects [41], and can predict the next scene. Likewise, recent approaches in video prediction have attempted to decompose high-dimensional videos into various factors, such as pose/content [16], motion/content [47, 48], motion context [22], multi-frequency [19], and object parts [53]. The decomposed factors are easier to predict since they address the prediction on lower-dimensional representations.

In this paper, we propose a video prediction framework that captures the global context and local motion patterns in the scene with two streams: local filter memory networks (LFMN) and global context propagation networks (GCPN). We present memory-based local filters that address the motion dynamics of objects over the frames. By leveraging the memory module [55], LFMN can generate filter kernels that contain the prototypical motion of moving objects on the scene. Namely, the filter kernels with memory facilitate learning the pixel motions (Fig. 1(c) and 1(d)) and predicting long-term future frames. Second, to capture the global context that lies over the consecutive frames, GCPN iteratively propagates the context from reference pixels to the entire image by computing the pairwise similarity in a non-local manner [50]. Through the propagation steps, GCPN aggregates the regions which have similar appearances to predict future frames consistent to the global context (Fig. 1(b)). Finally, we integrate the global and local information by filtering the global feature from GCPN with memory-based kernels containing motion patterns from LFMN (Fig. 4).

The proposed method can preserve the global context and refine the movements of dynamic objects simultaneously, resulting in more accurate video prediction. In experiment, we validate our method on the Caltech pedestrian [5] and UCF101 [45] datasets. Our method shows state-of-the-art performance and improves the prediction in the large-motion sequences and multi-step prediction.

2 Related Work

Video prediction. Early studies in video prediction adopt recurrent neural networks (RNNs) to consider the temporal information between frames [39]. To achieve better performance in long-term prediction, Long-Short-Term-Memory (LSTM) networks [46] and convLSTM [44]

is proposed. Recent study [53] exploits two cascaded adjacent recurrent states for spatio-temporal dynamics. Motivated by the recent success of 3D convolutions, a lightweight model is proposed to use a two-way autoencoder incorporating 3D convolutions [60], and a model is designed to integrate 3D convolutions into RNNs [52].Several recent methods have adopted GAN since it generates sharper results in image generation [21, 24, 29, 47, 49]. Since they focus on generating future frames at a global level through direct pixel synthesis, they show limited ability in capturing moving objects and large motions.

Another line of research computes the pixel-wise motion information from consecutive frames, and then, explicitly learning it or exploiting it as input [8, 23, 24, 25, 26, 37, 40, 53]. The estimated flow information is partially utilized to predict the future movement of an object [58]. Moreover, motion information can be extended to spatio-temporal domain [26] or multi-step flow prediction [23] for video prediction. However, these methods estimate the next frames with the local receptive fields while not fully considering the global context. Moreover, some studies [22, 35, 38, 47, 58] using motion achieved limited success on few simple datasets which have low variances such as the Moving MNIST [46], KTH action [43], and Human 3.6M [17]. Unlike previous works, we conduct experiments on more challenging datasets including Caltech and UCF101 and exploit two complementary attributes by incorporating global contextual information and local motion dynamics.

Future prediction errors in incomplete models can be classified as follows [53, 52]: (i) how to estimate the systematic errors owing to a lack of modeling ability for deterministic variations, (ii) how to model the probabilistic and inherent uncertainty about the future [7, 12, 54, 56]. Our paper belongs to video prediction, which focuses on the first factor.

Memory network. To address the long-term dependencies in video prediction, various methods have employed RNNs or LSTM as their backbone models [1, 11, 15]. However, their capacity is not large enough to accurately recover the features from the past information. To overcome the problems, the memory networks [55] introduce an external memory component that can be read and written for prediction. The external memory can be used to address various vision problems, including video object segmentation [51, 52], image generation [62], object tracking [59], and anomaly detection [10, 34]. In our work, we extend the idea of the memory networks to make it suitable for our task, video prediction. Specifically, the memory in local networks is dynamically recorded and updated with the new prototypical motion of moving objects, which helps in predicting motion dynamics in videos.

3 Proposed Method

3.1 Problem Statement and Overview

Let X_t be the tth frame in the video sequence $\mathbf{X} = (X_{t-(\delta-1)}, ..., X_t)$ of the last δ-frames. The goal of video prediction is to generate the next n frames $\mathbf{Y} = (Y_{t+1}, ..., Y_{t+n})$ from the input sequence $\mathbf{X}$. The major challenge in video prediction is to handle the complex evolution of pixels by composing two key attributes of the scene, *i.e.* context of the content and motion dynamics.

To this end, as shown in Fig. 2, we devise a new framework that takes advantages of two complementary components estimated from two sub-networks: local filter memory networks (LFMN) and global context propagation networks (GCPN). Following the previous works [21, 25, 40], our networks take the concatenated δ frames as inputs and feed them into the encoder to obtain high-dimensional embedded representation. LFMN takes the encoded representation to generate adaptive filter kernels that contain the prototype of moving
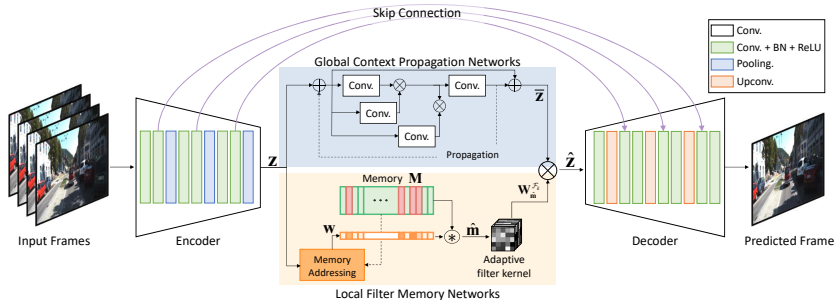
Figure 2: Overview of the proposed method. Our model consists of global context propagation networks (GCPN) and local filter memory networks (LFMN). $\otimes$, $\oplus$, and $\circledast$ are the convolution, addition, and multiplication operation, respectively.

objects. GCPN transforms the encoded representation to capture the contextual information by iteratively propagating all the points based on a non-local manner [50]. To incorporate global context and motion dynamics effectively, the intermediate features from GCPN are convolved with the filters generated from LFMN. Finally, the filtered features are fed into the decoder to predict the next frame.

3.2 Local Filter Memory Networks

To consider motion dynamics in video sequences, we introduce LFMN that captures and memorizes the prototypical motion of moving objects from the encoded representation $\mathbf{Z} \in \mathbb{R}^{W \times H \times C}$. For instance, as shown in 3, the objects in the sequence can move different directions. We thus aim at learning to update and address kernel parameters encoding prototypical motion patterns that transform global representation to be robust in dynamic changes.

We design the memory as a matrix $\mathbf{M} \in \mathbb{R}^{N \times C}$ containing N real-valued memory vectors of fixed channel dimension C. The prototypical items of the encoded future-relevant local motion features are updated to be written into the memory module. Given the encoded representation $\mathbf{Z}$, the memory networks retrieve a memory item most similar to $\mathbf{Z}$. In the memory addressing operator [55], it takes the encoding $\mathbf{Z}$ as query to obtain the addressing weights. We compute the similarity of the memory items and the encoded representation $\mathbf{Z}$. We compute each weight $\mathbf{w}_i$ for i^{th} memory vector $\mathbf{m}_i$ by applying softmax operations as follows:

$$\mathbf{w}_i = \frac{\exp(d(\mathbf{Z}, \mathbf{m}_i))}{\sum_{j=1}^{N} \exp(d(\mathbf{Z}, \mathbf{m}_j))}, \quad (1)$$

where $d(\cdot, \cdot)$ denotes a similarity measurement. We define $d(\cdot, \cdot)$ similar to [42] as follows:
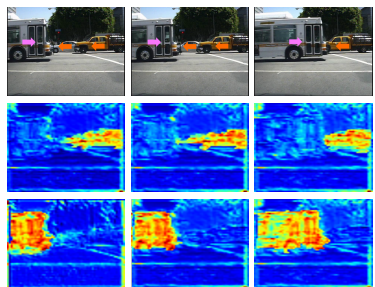

Figure 3: Visualization of memory attention maps in LFMN. In the sequence with moving objects (1st row), the attention maps are activated by different memory items with respect to the object motion (2nd, 3rd rows). The attention maps are upsampled to match the image size.

$$d(\mathbf{Z}, \mathbf{m}_i) = \frac{\mathbf{Z}\mathbf{m}_i^{\mathsf{T}}}{\|\mathbf{Z}\| \, \|\mathbf{m}_i\|}. \quad (2)$$

Given an encoded representation $\mathbf{Z}$, the memory networks obtain an aggregated memory feature $\hat{\mathbf{m}} \in \mathbb{R}^{W \times H \times C}$ relying a soft addressing weight $\mathbf{w} \in \mathbb{R}^{W \times H \times N}$ as follows

$$\hat{\mathbf{m}} = \mathbf{w}\mathbf{M} = \sum_{i=1}^{N} \mathbf{w}_i \mathbf{m}_i. \tag{3}$$

The $\hat{\mathbf{m}}$ is a combination of the memory items most similar to $\mathbf{Z}$. This allows our model to capture various local motion patterns using memory items. Finally, we generate the future-relevant local motion filter kernels using a filter generating network $\mathcal{F}_k$ such that

$$\mathbf{W}_{\hat{\mathbf{m}}}^{\mathcal{F}_k} = \mathcal{F}_k(\hat{\mathbf{m}}), \tag{4}$$

where k is the kernel size and $\mathbf{W}_{\hat{\mathbf{m}}}^{\mathcal{F}_k} \in \mathbb{R}^{W \times H \times C \times C \times k \times k}$ denotes the generated local motion filters. This generates filter kernels for each pixel on conditioned video frames. While the DFN [13] are limited on having to predict the next frame within the short-length input frame, our LFMN enables us to capture the long-term dynamics thanks to the memory. Fig. 3 shows a visualization of an attention map that is activated by memory items. The second row shows prototype patterns such as orange arrows which are activated in the movement of vehicles moving to the left. The third row shows that another type of prototype pattern is activated in the movement of objects moving to the right. We can observe that that each memory item addresses the particular movement pattern of objects.

3.3 Global Context Propagation Networks

The objective of GCPN is to capture a spatially wide range by propagating neighbor observations. To capture the content from different query locations far away within the image, GCPN is built upon non-local networks [50] which calculates the pairwise relationship regardless of the pixel position. However, non-local networks have a weakness to contain contextual relationships effectively [9], which may not always be effective to capture similar content information. Instead of directly using non-local networks, we construct a propagation step that aggregates more relevant elements along with the most discriminative parts.

Algorithm 1 summarizes GCPN. Given the encoded feature $\mathbf{Z}$, we compute the affinity matrix representing the relationship between all points. Unlike [50], we update the affinity matrix iteratively to propagate the future relevant context information. We conduct matrix multiplication between the updated affinity matrix (h^L) and a non-local block ($\mathbf{W}_o$) that denote linear transformation matrices (e.g., 1×1 convolution). At each step l, the feature is computed by aggregating the non-local neighboring representations [50]. The global context is propagated through GCPN that computes affinity matrix via self-attention operation, $\mathcal{G}$, such that

---

Algorithm 1 GCPN

Input: Intermediate feature $\mathbf{Z}$
Output: Global contextual feature $\bar{\mathbf{Z}}$
Parameters : $\mathbf{W}_\theta$, $\mathbf{W}_\phi$, $\mathbf{W}_g$, $\mathbf{W}_o$

1: **procedure** PROPAGATION($\mathbf{Z}$)
2:     Initialize $\mathbf{h}^0 = \mathbf{Z}$
3:     **for** $l = 1$ to L **do**
4:         $\mathbf{c} = (\mathbf{h}^{l-1}\mathbf{W}_\theta) \cdot (\mathbf{h}^{l-1}\mathbf{W}_\phi)^\top$
5:         $\bar{\mathbf{c}} = \text{softmax}(\mathbf{c})$
6:         $\mathbf{h}^l = \bar{\mathbf{c}} \cdot (\mathbf{h}^{l-1}\mathbf{W}_g)$
7:     **end for**
8:     $\bar{\mathbf{Z}} = \mathbf{Z} + \mathbf{h}^L\mathbf{W}_o$
9: **end procedure**

$$\mathbf{h}^l = \mathcal{G}(\mathbf{h}^{l-1}), \tag{5}$$

where $l = 1, ..., L$ and $\mathbf{h}^0 = \mathbf{Z}$.

At the final propagation step L, globally enhanced feature $\bar{\mathbf{Z}}$ is calculated by the sum of initial feature $\mathbf{Z}$ and propagated feature:

$$\bar{\mathbf{Z}} = \mathbf{Z} + \mathbf{h}^L\mathbf{W}_o, \tag{6}$$

with a weight matrix $\mathbf{W}_o \in \mathbb{R}^{C \times C}$. This procedure allows GCPN to consider all positions for each location at each step and produce non-locally enhanced feature representations by propagating the future-relevant contextual information.

To incorporate global information and local information, we merge the output features and generated filter kernels from GCPN and LFMN by the convolutional filtering operation, respectively. Given the enhanced feature $\bar{\mathbf{Z}}$ from GCPN, this is convolved with the prediction of the generated filter kernels $\mathbf{W}_{\hat{\mathbf{m}}}^{\mathcal{F}_k}$ from LFMN as follows:

$$\hat{\mathbf{Z}} = \bar{\mathbf{Z}} \otimes \mathbf{W}_{\hat{\mathbf{m}}}^{\mathcal{F}_k}, \tag{7}$$

where $\otimes$ is the convolution operation. In this process, we generate an adaptive filter that contains motion dynamic information for every pixel. The resulting features are fed into the decoder and then we estimate next future frame $\hat{Y}_{t+1}$.

3.4 Loss Function

To train the proposed model, we minimize an overall objective function $\mathcal{L}$, that includes a reconstruction loss $\mathcal{L}_r$ and a gradient loss $\mathcal{L}_g$:

$$\mathcal{L} = \mathcal{L}_r + \lambda_g \mathcal{L}_g, \tag{8}$$

where λ_g is a weighting factor. The reconstruction loss, $\mathcal{L}_r$, measures the difference between an estimated future frame $\hat{Y}_{t+1}$ and its corresponding ground-truth frame Y_{t+1}:

$$\mathcal{L}_r = \sum_t \left\| \hat{Y}_{t+1} - Y_{t+1} \right\|_1, \tag{9}$$

where $\|\cdot\|_1$ is L_1 norm which does not over-penalize the error and thus enables sharper predicted images [30, 40]. We also use the gradient loss $\mathcal{L}_g$, similar to [25, 29], that computes the differences of image gradient predictions and enforces to preserve image edges effectively:

$$\mathcal{L}_g = \sum_t \sum_{u,v} \left\| \left| Y_{t+1}^{u,v} - Y_{t+1}^{u-1,v} \right| - \left| \hat{Y}_{t+1}^{u,v} - \hat{Y}_{t+1}^{u-1,v} \right| \right\|_1 + \left\| \left| Y_{t+1}^{u,v-1} - Y_{t+1}^{u,v} \right| - \left| \hat{Y}_{t+1}^{u,v-1} - \hat{Y}_{t+1}^{u,v} \right| \right\|_1, \tag{10}$$

$\hat{Y}_{t+1}^{u,v}$ and $Y_{t+1}^{u,v}$ are the pixel elements from the estimated future frame $\hat{Y}_{t+1}$ and its corresponding ground-truth Y_{t+1}, respectively. u, v denotes the coordinates of the pixel for the width and height. $|\cdot|$ indicates the absolute value function.

3.5 Implementation Details

All models were trained end-to-end using PyTorch [36], taking about 2 days, with an Nvidia RTX TITAN. The network is trained using Adam [20] with an initial learning rate of 0.0002 and batch size of 16 for 60 epochs. During the training, the learning rate is reduced using a cosine annealing method [27] and the memory $\mathbf{M}$ is randomly initialized. At the testing phase, we read the memory items based on the input query. All training video frames were normalised to the range of [-1, 1]. We set the height, H, width, W, channels, C, and memory items, N, to 64, 64, 64, and 20, respectively. For the time index $t < \delta$, we copy the first frame $\delta - t$ times to get the δ frames for predicting the future frames. We achieved the best results by setting the generated kernel size k to 5. We used a grid search to set the parameter ($\lambda_g = 0.01$) on the validation set of the KITTI dataset. Details of the network architecture are provided in the supplementary material.

4 Experiment

4.1 Experimental Settings

Baselines. In the experiment, we consider representative baselines of the most relevant methods to our method such as PredNet [28], BeyondMSE [29], Dual-GAN [24] MC-Net [48], SDC-Net [40], Liu *et al*. [25], CtrlGen [14], DPG [8], Kwon *et al*. [21], Jin *et al*. [19], and CrevNet [60]. We used the pre-trained models provided by authors for visual comparison. We additionally obtained the results from ContextVP [2].

Metrics. For quantitative comparison, we employ several evaluation metrics that have been used most widely for video prediction such as, Mean-Squared Error (MSE), Structural Similarity Index Measure (SSIM), and Peak Signal to Noise Ratio (PSNR). Since these metrics are mostly focused on the pixel-level image quality, we additionally measure Learned Perceptual Image Patch Similarity (LPIPS) [61] as an evaluation metric for perceptual dissimilarity. Higher values of SSIM/PSNR and lower values of LPIPS indicate better quality.

Datasets. We evaluate our method on two different datasets such as Caltech pedestrian [5] and UCF101 [45] datasets. The Caltech pedestrian dataset consists of 640×480 videos taken from various driving places using vehicle-mounted cameras. To validate the generalization performance, we followed experimental protocols of [2, 21, 28], where the KITTI dataset with 41K images is used of training and the Caltech pedestrian dataset is used for testing. The KITTI dataset contains 375×1242 image sequences for driving scenes. Both datasets also contain dynamic scenes since they were recorded from the moving vehicles.

In addition, we use the UCF101 dataset that is generally used for action recognition and contains 320×240 videos focusing on human activities. This dataset mainly contains the moving objects in various environments. Since it contains a large amount of videos (13K videos), we use 10% of the videos, similar to the previous works [2, 21, 29].

4.2 Ablation study

We compare the performance of our network trained with and without GCPN in Table 1(left). The results show improved performance with GCPN as it aggregates non-local neighboring features to consider the global context effectively. As shown in the third and fourth row Table 1(left), GCPN shows greater performance improvement than LFMN in terms of the PSNR results as GCPN improves entire images while LFMN focuses on partial details. The proposed model, GCPN combined with LFMN, thus shows the best performance thanks to the benefit from both networks.

We compare the proposed model trained with and without LFMN as shown in Table 1(left). Compared to the basic U-net

Figure 4: Visualization of feature maps on the UCF101 dataset. (From top to bottom) Input frames, output features of GCPN, and output features of GCPN convolved with filter kernels from LFMN. The bottom figures show higher response around the moving objects.

LFMN	GCPN	PSNR ↑	SSIM ↑	LPIPS ↓ ($\times 10^{-2}$)
✗	✗	28.2	0.908	5.08
✓	✗	29.1	0.917	4.84
✗	✓	29.5	0.919	4.83
✓	✓	**30.1**	**0.927**	**4.81**

Number of memory items	0	5	10	20	30
PSNR ↑	29.5	29.87	29.92	**30.1**	30.1
SSIM ↑	0.919	0.920	0.923	**0.927**	0.923
LPIPS ↓	4.83	4.83	4.82	**4.81**	4.81

Table 1: Ablation studies of ours on the Caltech dataset. (Left) Results with and without LFMN and GCPN and (right) results with the different number of memory items in LFMN.

network, a baseline, without LFMN and GCPN, our model with LFMN yields improved prediction performance in both Caltech and UCF101 datasets. Fig. 4 shows the examples to validate that LFMN can record the local motion patterns and generate adaptive filter kernels by combining the prototypical elements of the encoded future-relevant features. This shows the output features of GCPN convolved with and without filter kernels from LFMN. In the 3rd row, we show that the integration of adaptive filter kernels with the output features from GCPN. With LFMN, the feature response around the moving objects becomes higher while that around the static region decreases. The results show that the output features convolved with filter kernels capture moving objects effectively.

Table 1(right) shows the evaluation by varying the number of memory items (N), from 0 to 30 to verify the effect of the memory items. The model with a larger N generally shows better prediction results since diverse prototypical motions of moving objects can be more recorded in the memory items. The effect of LFMN starts to converge when N becomes 20 or larger.

Figure 5: Visualization of feature maps of GCPN at each step l. The feature maps are color-coded, where warmer colors indicate higher values.

To demonstrate the changes in each step l in GCPN, we visualize the feature maps on the Caltech dataset, as shown in Fig. 5. Fig. 5 shows that the input sequence (left) of the Caltech dataset, and the 4th and 16th feature map of GCPN (right). At each step l, GCPN aggregates non-local neighbors, propagating global spatial context. In addition, in terms of PSNR/SSIM, we obtained the 29.7/0.921 at $l = 1$ and 30.1/0.927 at $l=2$, respectively. The results show the improved performance with GCPN because it aggregates non-local neighboring features to consider the global context effectively.

In addition, as in [21], we present an experiment to evaluate the effect of the number of input frames when predicting the next frame. Table 2 shows the quantitative results according to the number of input images. Similar to [21], we achieve good performance when using the inputs from 2 to 6 frames, and the performance starts to decrease when using more inputs.
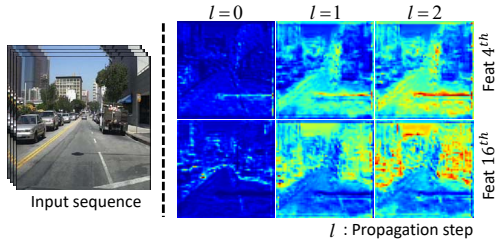
	# of frames	2	4	6	8	10
Kwon *et al.* [21]	PSNR ↑	29.17	29.22	29.01	28.94	29.01
	SSIM ↑	0.919	0.918	0.920	0.919	0.918
Ours	PSNR ↑	29.73	**30.1**	29.51	29.02	28.93
	SSIM ↑	0.920	**0.927**	**0.927**	0.921	0.918

Table 2: Video prediction results on the Caltech pedestrian dataset by changing the number of input frames.

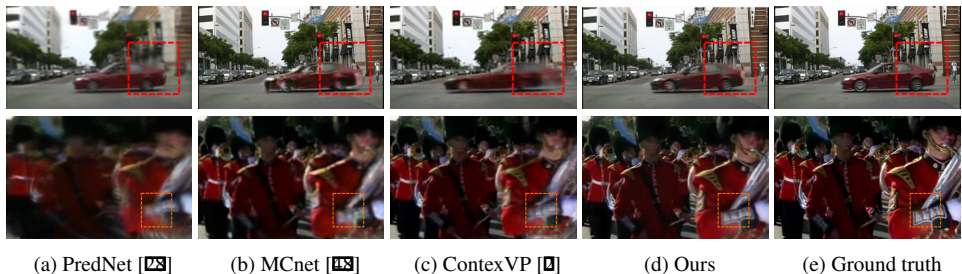| (a) PredNet [23] | (b) MCnet [43] | (c) ContexVP [2] | (d) Ours | (e) Ground truth |

Figure 6: Examples of the next-frame prediction on the Caltech Pedestrian (1st row) and the UCF101 dataset (2nd row).

4.3 Comparison with state-of-the-art methods

4.3.1 Next-frame prediction.

To evaluate the performance of the proposed method, we compare the accuracy of the next-frame predictions with several state-of-the-art video prediction methods. Sample results on thez Caltech pedestrian [5] dataset are shown in Fig. 6 in the first row. Each model is only trained on the KITTI [9] dataset and evaluated on the Caltech pedestrian dataset without additional fine-tuning process. Our method uses 4 past frames as input images, while PredNet [23] and ContextVP [2] use 10 past frames as input but showing blurry results. MCnet [43] also shows blurry results and unnatural deformations on the highlighted car in the first row. ContextVP [2] shows a lot of artifacts in moving parts of objects such as the backside of a car.

For capturing the motion dynamics, MCnet [43] takes the difference between two consecutive frames. Because they are mainly focused on short-term information, there is a limit to their performance in estimating dynamic motion. In contrast, our method mitigates the above problems by considering both the global contextual information and local motion dynamics with the memory networks. The second row shows the results of the UCF101 [45] dataset that contains human actions taken in the wild and exhibits various challenges. The results show that our method produces sharp predictions and visually pleasing results compared to the state-of-the-art methods. More qualitative results are provided in supplementary materials.

Method	Caltech ped.			UCF101		
	PSNR ↑	SSIM ↑	MSE ↓	PSNR ↑	SSIM ↑	MSE ↓
Last Frame	23.3	0.779	7.95	30.2	0.89	4.09
PredNet [23]	27.6	0.91	2.42	19.87	-	15.50
BeyondMSE [24]	-	0.881	3.26	22.78	-	9.26
Dual-GAN [21]	-	0.899	2.41	30.5	0.94	-
MCnet [43]	-	0.879	2.50	31.0	0.91	-
ContexVP [2]	28.7	0.92	1.94	34.9	0.92	-
SDC-Net [30]	-	0.918	1.62	-	-	-
CtrlGen [12]	26.5	0.90	-	28.8	0.92	-
DPG [8]	28.2	0.923	1.62	-	-	-
Kwon et al. [20]	29.2	0.919	1.61	35.0	0.94	1.37
Jin et al. [13]	29.1	**0.927**	-	-	-	-
CrevNet [50]	29.3	0.925	-	-	-	-
Ours w/o $\mathcal{L}_g$	29.8	0.921	1.60	35.2	0.94	1.35
Ours	**30.1**	**0.927**	**1.58**	**35.5**	**0.95**	**1.32**

Table 3: Quantitative evaluation of video prediction. The table shows the performance of state-of-the-arts methods for Caltech and UCF101 datasets, respectively. MSE is averaged per pixel ($\times 10^{-3}$). The scores of other methods are adopted from original papers. The best results are presented in bold.

Table 3 shows a quantitative comparison with several state-of-the-art methods in both datasets. We also report the results obtained by copying the last frame, which is the trivial baseline that uses the most current past frame as the prediction. The last two rows of Table 3 show that $\mathcal{L}_g$ improves the performance by preserving the image edges. Our method signif-

icantly outperforms the baselines with decomposition (MCNet [48], DPG [8], CtrlGen [14]) or without disentanglement. For DPG [8] and CtrlGen [14], training with estimated optical flows may lead to erroneous supervision signals. Contrarily, our method outperforms the state-of-the-art methods thanks to the GCPN, LFMN, and integration of each network.

Table 4 compares the running time between ours and existing methods. We follow the original setting of all the released codes. In GCPN and LFMN, we obtain 0.161s and 0.112s on average on the Caltech dataset, respectively. On average, our method takes about 0.451.

	PredNet [28]	MCnet [48]	Ours
Time	0.321	0.304	0.451

Table 4: Comparison of running time.

4.3.2   Multi-step prediction

To validate the time and spatial consistency in the long-term future, we present the experiment on multi-step prediction [8, 14, 21, 28, 48]. The experimental scheme is as follows. For example, the networks that require 4 input frames take the first 4 consecutive frames as input to find the 5th frame. The 6th frame is then predicted by using the 2nd to 4th frames and the 5th predicted frame as input. We continuously find the next future frame using the predicted results. Fig. 7 shows the quantitative results for multi-step predictions. All the models take in 4 frames as input except PredNet [28] and recursively predict the next 15 frames. Our method consistently outperforms recent approaches on all metrics over time. In terms of LPIPS, while the Kwon *et al.* [21] and

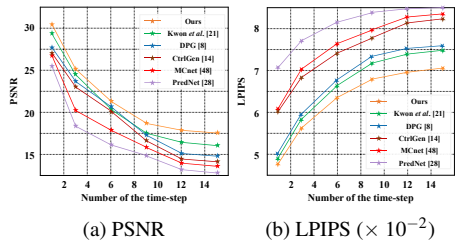(a) PSNR   (b) LPIPS ($\times 10^{-2}$)

Figure 7: Quantitative comparison of multi-step prediction results with state-of-the-art methods and the proposed method on the Caltech pedestrian dataset.

DPG [8] face a performance degradation significantly after 6 steps, our method does not quickly drop thanks to the previously well-predicted results. The proposed method achieves outstanding results at anticipating the far future frames compared to state-of-the-art methods using with/without explicit motion estimation. Due to the page limitation, we put the qualitative comparisons with state-of-the-art methods to the supplementary materials.

5   Conclusion

In this paper, we presented the video prediction framework that includes GCPN and LFMN, two complementary convolutional neural networks, to capture global contextual information and local motion patterns of objects, respectively. GCPN considers the global context by iteratively aggregating the non-local neighboring representations. LFMN generates adaptive filter kernels using memory items that have the prototypical motion of moving objects. The integration of the two networks preserves the global context and refines the movements of dynamic objects simultaneously. Our in-depth analysis shows that the proposed method significantly outperforms state-of-the-art methods on the next frame prediction as well as the multi-step prediction. In future work, we will investigate the applicability of our model to other applications such as video semantic segmentation and anomaly detection.

Acknowledgements

References

[1] C. G Atkeson and S. Schaal. Memory-based neural networks for robot learning. In *Neurocomputing*, 1995.

[2] W. Byeon, Q. Wang, R. K Srivastava, and K. Petros. Contextvp: Fully context-aware video prediction. In *ECCV*, 2018.

[3] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *CVPRW*, 2019.

[4] J. Cho, Y. Kim, H. Jung, C. Oh, J. Youn, and K. Sohn. Multi-task self-supervised visual representation learning for monocular road segmentation. In *ICME*, 2018.

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

[6] C. Finn, I. J Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *NIPS*, 2016.

[7] JY. Franceschi, E. Delasalles, M. Chen, S. Lamprier, and P. Gallinari. Stochastic latent residual video prediction. In *ICML*, 2020.

[8] H. Gao, H. Xu, O. Z Qi, R. Wang, F. Yu, and T. Darrell. Disentangling propagation and generation for video prediction. In *ICCV*, 2019.

[9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. In *IJRR*, 2013.

[10] D. Gong, L. Liu, V. Le, B. Saha, M. R Mansour, S. Venkatesh, and A. V D Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *ICCV*, 2019.

[11] A. Graves. Generating sequences with recurrent neural networks. In *arXiv preprint arXiv:1308.0850*, 2013.

[12] VL. Guen and N. Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *CVPR*, 2020.

[13] L. Y Gui, Y. X Wang, D. Ramanan, and J. MF Moura. Few-shot human motion prediction via meta-learning. In *ECCV*, 2018.

[14] Z. Hao, X. Huang, and S. Belongie. Controllable video generation with sparse trajectories. In *CVPR*, 2018.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. In *Neural computation*, 1997.

[16] J. T Hsieh, B. Liu, D. A Huang, L. Fei-Fei, and J. C Niebles. Learning to decompose and disentangle representations for video prediction. In *arXiv preprint arXiv:1806.04166*, 2018.

[17] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. In *ICPR*, 2013.

[18] X. Jia, BD. Brabandere, T. Tuytelaars, and LV. Gool. Dynamic filter networks. In *NIPS*, 2016.

[19] B. Jin, Y. Hu, Q. Tang, J. Niu, Z. Shi, Y. Han, and X. Li. Exploring spatial-temporal multi-frequency analysis for high-fidelity and temporal-consistency video prediction. In *CVPR*, 2020.

[20] D. P Kingma and J. Ba. Adam: A method for stochastic optimization. In *arXiv preprint arXiv:1412.6980*, 2014.

[21] Y. Kwon and M. Park. Predicting future frames using retrospective cycle gan. In *CVPR*, 2019.

[22] S. Lee, H. Kim, D. Choi, H. Kim, and Y. Ro. Video prediction recalling long-term motion context via memory alignment learning. In *CVPR*, 2021.

[23] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M. H Yang. Flow-grounded spatial-temporal video prediction from still images. In *ECCV*, 2018.

[24] X. Liang, L. Lee, W. Dai, and E. P Xing. Dual motion gan for future-flow embedded video prediction. In *ICCV*, 2017.

[25] W. Liu, W. Luo, D. Lian, and S. Gao. Future frame prediction for anomaly detection–a new baseline. In *CVPR*, 2018.

[26] Z. Liu, R. A Yeh, X. Tang, Y. Liu, and A. Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, 2017.

[27] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *arXiv preprint arXiv:1608.03983*, 2016.

[28] W. Lotter, G. Kreiman, and D. Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *ICLR*, 2017.

[29] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016.

[30] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive separable convolution. In *ICCV*, 2017.

[31] S. Oh, J. Lee, N. Xu, and S. Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019.

[32] S. Oh, J. Lee, N. Xu, and S. Kim. Kernelized memory network for video object segmentation. In *arXiv preprint arXiv:2007.08270*, 2020.

[33] S. Oprea, P. Martinez-Gonzalez, A. Garcia-Garcia, J. Castro-Vargas, A. John, S. Orts-Escolano, J. Garcia-Rodriguez, and A. Argyros. A review on deep learning techniques for video prediction. In *TPAMI*, 2020.

[34] H. Park, J. Noh, and B. Ham. Learning memory-guided normality for anomaly detection. In *CVPR*, 2020.

[35] S. Park, K. Kim, J. Lee, J. Choo, J. Lee, and S. Kim. Vid-ode: Continuous-time video generation with neural ordinary differential equation. In *AAAI*, 2021.

[36] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

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

[38] R. Pottorff, J. Nielsen, and D. Wingate. Video extrapolation with an invertible linear embedding. In *arXiv preprint arXiv:1903.00133*, 2019.

[39] M. A Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. In *arXiv preprint arXiv:1412.6604*, 2014.

[40] F. A Reda, G. Liu, K. J Shih, R. Kirby, J. Barker, D. Tarjan, A. Tao, and B. Catanzaro. Sdc-net: Video prediction using spatially-displaced convolution. In *ECCV*, 2018.

[41] R. A Rensink. The dynamic representation of scenes. In *Visual cognition*, 2000.

[42] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. One-shot learning with memory-augmented neural networks. In *arXiv preprint arXiv:1605.06065*, 2016.

[43] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR*, 2004.

[44] X. Shi, Z Chen, H Wang, D. Y Yeung, W. K Wong, and W. C Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*, 2015.

[45] K. Soomro, A. R Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. In *arXiv preprint arXiv:1212.0402*, 2012.

[46] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015.

[47] S. Tulyakov, M. Liu, X. Yang, and J. Kautz. Mocogan: Decomposing motion and content for video generation. In *CVPR*, 2018.

[48] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee. Decomposing motion and content for natural video sequence prediction. In *ICLR*, 2017.

[49] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NIPS*, 2016.

[50] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *CVPR*, 2018.

[51] Y. Wang, M. Long, J. Wang, Z. Gao, and P. S Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *NIPS*, 2017.

[52] Y. Wang, L. Jiang, MH. Yang, LJ. Li, M. Long, and L. Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *ICLR*, 2018.

[53] Y. Wang, J. Zhang, H. Zhu, M. Long, J. Wang, and PS. Yu. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In *CVPR*, 2019.

[54] Y. Wang, J. Wu, M. Long, and JB. Tenenbaum. Probabilistic video prediction from noisy data with a posterior confidence. In *CVPR*, 2020.

[55] J. Weston, S. Chopra, and A. Bordes. Memory networks. In *arXiv preprint arXiv:1410.3916*, 2014.

[56] J. Xu, B. Ni, and X. Yang. Video prediction via selective sampling. In *NIPS*, 2018.

[57] K. Xu, L. Wen, G. Li, L Bo, and Q. Huang. Spatiotemporal cnn for video object segmentation. In *CVPR*, 2019.

[58] Z. Xu, Z. Liu, C. Sun, K. Murphy, W.T. Freeman, J. B. Tenenbaum, and J. Wu. Unsupervised discovery of parts, structure, and dynamics. In *ICLR*, 2019.

[59] T. Yang and A. B Chan. Learning dynamic memory networks for object tracking. In *ECCV*, 2018.

[60] W. Yu, Y. Lu, S. Easterbrook, and S. Fidler. Efficient and information-preserving future frame prediction and beyond. In *ICLR*, 2019.

[61] R. Zhang, P. Isola, A. A Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[62] M. Zhu, P. Pan, W. Chen, and Y. Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *CVPR*, 2019.