ContextVP: Fully Context-Aware Video Prediction

Wonmin Byeon[1234], Qin Wang[1], Rupesh Kumar Srivastava[3], and Petros Koumoutsakos[1]

[1]ETH Zurich
[2]The Swiss AI Lab IDSIA
[3]NNAISENSE
[4]NVIDIA

Abstract

Video prediction models based on convolutional networks, recurrent networks, and their combinations often result in blurry predictions. We identify an important contributing factor for imprecise predictions that has not been studied adequately in the literature: blind spots, i.e., lack of access to all relevant past information for accurately predicting the future. To address this issue, we introduce a fully context-aware architecture that captures the entire available past context for each pixel using Parallel Multi-Dimensional LSTM units and aggregates it using blending units. Our model outperforms a strong baseline network of 20 recurrent convolutional layers and yields state-of-the-art performance for next step prediction. Moreover, it does so with fewer parameters than several recently proposed models, and does not rely on deep convolutional networks, multi-scale architectures, separation of background and foreground modeling, motion flow learning, or adversarial training. These results highlight that full awareness of past context is of crucial importance for video prediction.

1. Introduction

Blurry predictions are fundamentally a manifestation of model uncertainty, which increases if the model fails to sufficiently capture relevant past information. Unfortunately, this source of uncertainty has not received sufficient attention in the literature. Most current models are not designed to ensure that they can properly capture all possibly relevant past context. This paper attempts to address this gap. Quantitative improvements on metrics are accompanied by results of high visual quality showing sharper future predictions with reduced blur or other motion artifacts. Since the proposed models do not require separation of content and motion or novel loss functions to reach the state of the art, we find that full context awareness is the crucial ingredient for high quality video prediction.

2. Missing Contexts in Other Network Architectures

Blurry predictions can result from a video prediction model if it does not adequately capture all relevant information in the past video frames which can be used to reduce uncertainty. Figure 1 shows the recurrent connections of a pixel at time t with a 3×3 convolution between two frames (left) and the information flow of a ConvLSTM predicting the pixel at time $T + 1$ (right). The covering context grows progressively over time (depth), but there are also blind spots which cannot be used for prediction. In fact, as can be seen in Figure 1 (right, marked in gray color), frames in the recent past have larger blind areas. Due to this structural issue, the network is unable to capture the entire available context and is likely to miss important spatio-temporal dependencies leading to increased ambiguity in the predictions. The prediction will eventually fail when the object appearance or motion in videos changes dramatically within a few frames.

One possible way to address limited context, widely used in CNNs for image analysis, is to expand context by stacking multiple layers (sometimes with dilated convolutions [13]). However, stacking layers still limits the available context to a maximum as dictated by the network architecture, and the number of additional parameters required to gain sufficient context can be very large for high resolution videos. Another technique that can help is using a multi-scale architecture, but fixed scale factors may not generalize to all possible objects, their positions and motions.

3. Method

We introduce the Fully Context-aware Video Prediction model (ContextVP) — an architecture that avoids blind spots

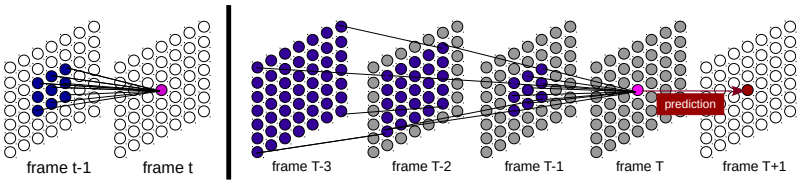

Figure 1: (left) The Convolutional LSTM (ConvLSTM) context dependency between two successive frames. (right) The context dependency flow in ConvLSTM over time for frame $t = T$. Blind areas shown in gray cannot be used to predict the pixel value at time $T + 1$. Closer time frames have larger blind areas.

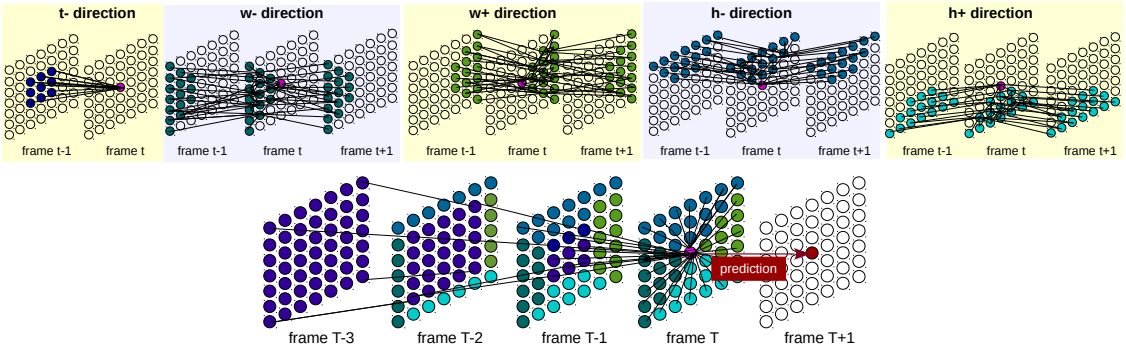

Figure 2: (top) Context dependency between two frames when using Parallel MD-LSTM (PMD) units for five directions: $t-$, $w-$, $w+$, $h-$, and $h+$, where h, w, and t indicate the current position for height, width, and time dimensions. (bottom) The combined context dependency flow for frame $t = T$ in the proposed architecture. All available context from past frames is covered in a single layer regardless of the input size.

by covering all the available context by design. Its advantages are:

- Since each processing layer covers the entire context, increasing depth is only used as necessary to add computation power, not more context. A priori specification of scale factors is also not required.
- Compared to models that utilize increased depth to cover larger context such as our baseline 20-layer models, more computations can be parallelized.
- Compared to state-of-the-art models from recent literature, it results in improved performance without the use of separation of motion and content, learning optical flow or adversarial training (although combinations with these strategies may further improve results).

Let $x_1^T = \{x_1, ..., x_T\}$ be a given input sequence of length T. $x_t \in \mathbb{R}^{H \times W \times C}$ is the t-th frame, where $t \in \{1, ..., T\}$, H is the height, W the width, and C the number of channels. For simplicity, assume $C = 1$, x_1^T is then a cuboid of pixels bounded by six planes. The task is to predict p future frame(s) in the sequence, $x_{t+1}^{t+p} = \{x_{t+1}, ..., x_{t+p}\}$ (next-frame prediction if $p = 1$). Therefore, our goal is to integrate information from the entire cuboid x_1^T into a representation at the plane where $t = T$, which can be used for predicting x_{t+1}^{t+p}. This is achieved in the proposed model by using fully context-aware layers, each consisting of two blocks. The first block is composed of *Parallel MD-LSTM units* (PMD) that sequentially aggregate information from different directions. The second block is the *Context Blending Block* that combines the output of PMD units for all directions. The context covered using PMD units for each direction (top) and the combined context from past frames (down) are visualized in Figure 2.

3.1. Parallel MD-LSTM Unit

Parallel computing units were used in the PyraMiD-LSTM [12] architecture and the idea of using LSTM to aggregate information from all directions was only explored in a limited setting (2D/3D image segmentation). They are mathematically similar to ConvLSTM units but our terminology highlights that it is **not** necessary to limit convolutional operations to spatial dimensions and LSTM connectivity to the temporal dimension as is conventional. PMD units can be used to aggregate context along any of the six directions available in a cuboid. Three directions are shown: t-, w+, and h+. At each plane, the local computation for each pixel is independent of other pixels in the same plane, so all pixels are processed as parallel using the convolution operation. The computational dependencies across planes are modeled using the LSTM operation. Computations for each PMD unit are explained mathematically below.

For any sequence of K two dimensional planes $x_1^K = \{x_1, ..., x_K\}$, the PMD unit computes the current cell and hidden state c_k, s_k using input, forget, output gates i_k, f_k, o_k, and the transformed cell $\tilde{c}_k$ given the cell and hidden state from the previous plane, c_{k-1}, s_{k-1}.

$$\begin{aligned} i_k &= \sigma(W_i * x_k + H_i * s_{k\text{-}1} + b_i), \\ f_k &= \sigma(W_f * x_k + H_f * s_{k\text{-}1} + b_f), \\ o_k &= \sigma(W_o * x_k + H_o * s_{k\text{-}1} + b_o) \\ \tilde{c}_k &= \tanh(W_{\tilde{c}} * x_k + H_{\tilde{c}} * s_{k\text{-}1} + b_{\tilde{c}}), \\ c_k &= f_k \odot c_{k\text{-}1} + i_k \odot \tilde{c}_k, \\ s_k &= o_k \odot \tanh(c_k). \end{aligned} \tag{1}$$

Here $(*)$ is the convolution operation, and $(\odot)$ the element-wise multiplication. W and H are the weights for input and the past state. The size of weight matrices are dependent only on the kernel size and number of units. If the kernel size is larger, more local context is taken into account.

As shown in Section 2, using a ConvLSTM would be equivalent to running a PMD unit along the time dimension from $k = 1$ to $k = T$, which would only integrate information from a pyramid shaped region of the cuboid and ignore several blind areas. For this reason, it is necessary to use four additional PMD units, for which the conditioning directions are aligned with the spatial dimensions, as shown in Figure 2 (top). We define the resulting set of five outputs at frame T as s^d where $d \in D = \{h\text{-}, h\text{+}, w\text{-}, w\text{+}, t\text{-}\}$ denotes the recurrence direction. Together this set constitutes a representation of the cuboid of interest x_1^T. Outputs at other frames in x_1^{T-1} are ignored.

3.2. Context Blending Block

This block captures the entire available context by combining the output of PMD units from all directions at frame T. This results in the critical difference from the traditional ConvLSTM: the context directions are aligned not only with the time dimension but also with the spatial dimensions. We consider two ways to combine the information from different directions.

Uniform blending (U-blending): this strategy was used in the traditional MD-LSTM [2, 6] and PyraMiD LSTM [12]. It simply sums the output of all directions along the channel dimension and then applies a non-linear transformation on the result:

$$m = f((\sum_{d \in D} s^d) \cdot W + b), \tag{2}$$

where $W \in \mathbb{R}^{N1 \times N2}$ and $b \in \mathbb{R}^{N2}$ are a weight matrix and a bias. $N1$ is the number of PMD units, and $N2$ is the number of (blending) blocks. f is an activation function.

Weighted blending (W-blending): the summation of PMD unit outputs in U-blending assumes that the information from each direction is equally important for each pixel. We propose W-blending to remove this assumption and learn the relative importance of each direction during training with the addition of a small number of additional weights compared to the overall model size. The block concatenates s from all directions:

$$S = \begin{bmatrix} s^{t-} & s^{h-} & s^{h+} & s^{w-} & s^{w+} \end{bmatrix}^T \tag{3}$$

The vector S is then weighted as follows:

$$m = f(S \cdot W + b), \tag{4}$$

where $W \in \mathbb{R}^{(5 \times N1) \times N2}$ (5 is the number of directions). Equations 2 and 4 are implemented using 1×1 convolutions. We found that W-blending is crucial for achieving high performance for the task of video prediction.

3.3. Directional Weight-Sharing (DWS)

Visual data tend to have structurally similar local patterns along opposite directions. This is the reason why horizontal flipping is a commonly used data augmentation technique in computer vision. We propose the use of a similarly inspired weight-sharing technique for regularizing the proposed networks. The weights and biases of the PMD units in opposite directions are shared i.e. weights for h- and h+ are shared, as are w- and w+. This strategy has several benefits in practice: 1) it lowers the number of parameters to be learned, 2) it incorporates knowledge about structural similarity into the model, and 3) it improves generalization.

4. Experiments

Network architecture: It consists of a stack of four context-aware layers with skip connections that directly predicts the scaled RGB values of the next frame. All results are reported for models using 3×3 convolutional kernels for all PMD units, identity activation function in Equations 2 and 4 and training using $\mathcal{L}_1$ with Image Gradient Difference Loss (GDL) [10].

Baseline: our baseline (ConvLSTM20) is a network consisting of a stack of 20 ConvLSTM layers with kernels of size 3×3. The number of layers was chosen to be 20 to cover a large context and also since each layer in our 4-layer model consists of 5 PMD units. Two skip connections similar to our model were also used. The layer sizes are chosen to keep the number parameters comparable to our best model (ContextVP4-WD-big). Surprisingly, **this baseline outperforms the state of the art models**.Note that it is **less amenable to parallelization** compared to ContextVP models where PMD units for different directions can be applied in parallel.

Car-mounted Camera Video Prediction (KITTI and CalTech Pedestrian dataset): The model is trained on

Figure 3: Qualitative comparisons from the test set among our best model (ContextVP4-WD-big), the baseline (ConvLSTM20), and the state-of-the-art model (PredNet). All models are trained for next-frame prediction given 10 input frames on the KITTI dataset, and tested on the CalTech Pedestrian dataset.

Table 1: Evaluation of Next frame prediction on the CalTech Pedestrian dataset (trained on the KITTI dataset). All models are trained on 10 frames and predicts the next frame. The results are averaged over test videos. ConvLSTM20 is our baseline containing 20 ConvLSTM layers. ContextVP4-WD-small has half the hidden units at each layer compared to ContextVP4-WD-big. Higher values of PSNR and SSIM, lower values of MSE indicate better results. (+) This score is provided by [7]. (*) The scores provided in Lotter et al. [9] are averaged over nine frames (time steps 2–10 in their study), but ours are computed only on the next predicted frame. We therefore re-calculated the scores of PredNet using their trained network. Our best models (ContextVP4-WD: 4 layers with weighted blending and DWS) outperform the baseline as well as current state-of-the-art methods with fewer number of parameters.

Method	MSE ($\times 10^{-3}$)	PSNR	SSIM	#param.
Copy-Last-Frame	7.95	23.3	0.779	-
+BeyondMSE [10]	3.26	-	0.881	-
*PredNet [9]	2.42	27.6	0.905	6.9M
Dual Motion GAN [7]	2.41	-	0.899	113M
ConvLSTM20	2.26	28.0	0.913	9.0M
ContextVP4-WD-small	2.11	28.2	0.912	2.0M
ContextVP4-WD-big	**1.94**	**28.7**	**0.921**	8.6M

the KITTI dataset [5] and tested on the CalTech Pedestrian dataset [3]. Every ten input frames from "City", "Residential", and "Road" videos are sampled for training resulting in ≈41K frames. Frames from both datasets are center-cropped and down-sampled to 128×160 pixels. We use the exact data preparation as PredNet [9] for direct comparison.

The car-mounted camera videos are taken from moving vehicles and consist of a wide range of motions. This dataset has diverse and large motion of cars at different scales and also has large camera movements. To make predictions for such videos, a model is required to learn not only small movement of pedestrians, but also relatively large motion of surrounding vehicles and backgrounds.

We compare our approach with the Copy-Last-Frame and ConvLSTM20 baselines as well as BeyondMSE, PredNet, and Dual Motion GAN [7] which are the current best models for this dataset. Note that the scores provided in Lotter et al. [9] are averaged over nine frames (time steps 2–10 in their study), but ours are computed only on the next predicted frame. Therefore, we re-calculated the scores of PredNet for the next frame using their trained network. As shown in Table 1, our four layer model with W-blending and DWS outperforms the state-of-the-art on all metrics. Once again, the smaller ContextVP network already matches the baseline while being much smaller and more suitable for parallelization. Some samples of the prediction results from the test set are provided in Figure 3. Our model is able to adapt predictions to the current scene and make sharper predictions compared to the baseline and PredNet.

References

[1] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki. Scene labeling with lstm recurrent neural networks. In *CVPR*, 2015. 4323

[3] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 304–311. IEEE, 2009. 4324

[4] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *Advances In Neural Information Processing Systems*, pages 64–72, 2016.

[5] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 4324

[6] A. Graves, S. Fernández, and J. Schmidhuber. Multi-dimensional recurrent neural networks. In *Proceedings of the 17th International Conference on Artificial Neural Networks*, September 2007. 4323

[7] X. Liang, L. Lee, W. Dai, and E. P. Xing. Dual motion gan for future-flow embedded video prediction. *arXiv preprint arXiv:1708.00284*, 2017. 4324

[8] Z. Liu, R. Yeh, X. Tang, Y. Liu, and A. Agarwala. Video frame synthesis using deep voxel flow. *arXiv preprint arXiv:1702.02463*, 2017.

[9] W. Lotter, G. Kreiman, and D. Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016. 4324

[10] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015. 4323, 4324

[11] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[12] M. F. Stollenga, W. Byeon, M. Liwicki, and J. Schmidhuber. Parallel multi-dimensional lstm, with application to fast biomedical volumetric image segmentation. In *Advances in Neural Information Processing Systems*, pages 2998–3006, 2015. 4322, 4323

[13] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 4321