

Location Dependency in Video Prediction

Niloofar Azizi*, Hafez Farazi*, and Sven Behnke

Bonn University, Computer Science Department,
Endenicher Allee 19a, 53115 Bonn, Germany
{farazi, behnke}@ais.uni-bonn.de, niloofarazizi37@gmail.com

Abstract. Deep convolutional neural networks are used to address many computer vision problems, including video prediction. The task of video prediction requires analyzing the video frames, temporally and spatially, and constructing a model of how the environment evolves. Convolutional neural networks are spatially invariant, though, which prevents them from modeling location-dependent patterns. In this work, the authors propose location-biased convolutional layers to overcome this limitation. The effectiveness of location bias is evaluated on two architectures: Video Ladder Network (VLN) and Convolutional Predictive Gating Pyramid (Conv-PGP). The results indicate that encoding location-dependent features is crucial for the task of video prediction. Our proposed methods significantly outperform spatially invariant models.

Keywords: Video prediction, Deep learning, Location-dependent bias

1 Introduction

The task of video prediction consists of predicting a set of successor frames, given a sequence of video frames. It is challenging, because the predictor needs to understand both contents and motion of the scene in order to make good predictions. In recent years, deep learning approaches became popular for video prediction. They analyze the video both spatially and temporally and learn hierarchical representations, which model the image evolution in terms of its content and dynamics ([1], [2]). The learned representations can be used for a variety of applications, including action recognition and anticipating future actions, which can be utilized for instance in human-robot interaction scenarios.

Convolutional deep learning architectures cannot recognize location-dependent features, however, due to the location-invariant nature of convolutions. In the task of the video prediction, for instance, learning the location of static obstacles in the environment leads to better frame forecasting. In this work, the authors propose three different methods to overcome this limitation:

- a) encoding location features in separate channels of the input,
- b) convolutional layers with learnable location-dependent biases, and
- c) convolutional layers with learnable location-dependent biases and predefined location encodings.

* Both authors contributed equally to this work.

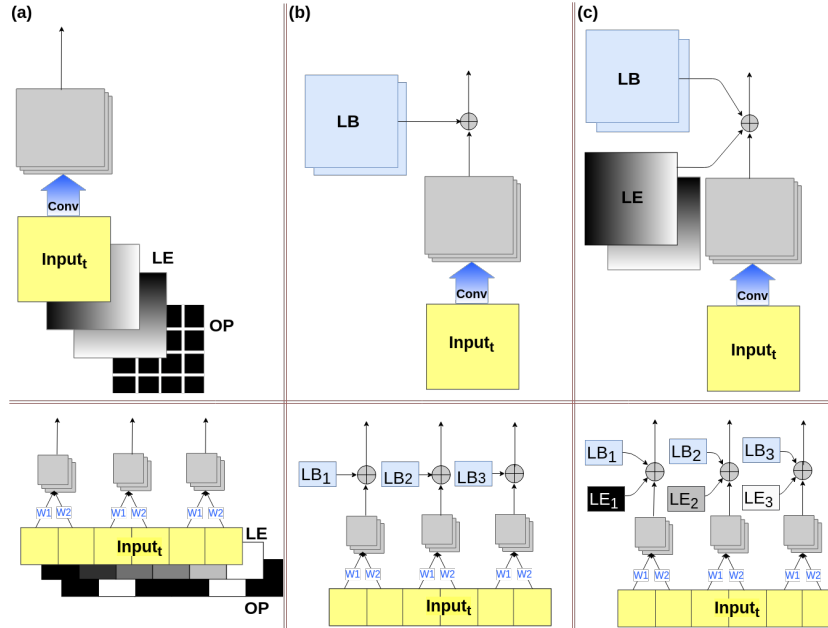


Fig. 1 Proposed methods for location-dependency. Top row 2D and bottom row 1D. a) three additional input channels, two of which encode location by gradients in x and y directions (LE); the third contains the occlusion pattern (OP). b) learnable location-dependent biases (LB) are added to the output of convolutions. c) learnable location-dependent biases and predefined location encodings use combined.

These methods are illustrated in Fig. 1 for 1D and two-dimensional convolutions.

We demonstrate the utility of our approach using two datasets that contain location dependencies. The code and datasets of this paper are publicly available.¹

2 Related Work

Convolutional deep learning architectures are spatially invariant, which leads to the constraint of not being able to model location-dependent patterns.

To address this issue in various computer vision tasks, different approaches have been explored. Utilizing fully connected layers leads to learning location-dependent features, but this has the drawbacks of many parameters and no spatial weight sharing. In the PixelCNN architecture for conditional image generation, Oord et al. [3] applied 1×1 convolutions to map a hidden representation into a spatial representation. The disadvantage of this approach is that to extract the spatial features, a very large number of parameters is needed. In saliency prediction, Kruthiventi et al. [4] proposed adding another set of convolutional

¹ <https://github.com/AIS-Bonn/LocDepVideoPrediction>

weights with the same size of the original filters. They convolved these additional weights with predefined fixed channels that encode the image center using Gaussian blobs with different horizontal and vertical extent. Ghafoorian et al. [5] applied specific location features to train the model and utilized location dependency for the task of brain MRI image segmentation. They showed that the results improve in comparison to CNNs that do not use location information. The above approaches depend all on predefined location feature structures.

For the task of video prediction, different approaches have been explored. The most successful ones utilize deep learning methods. Cricri et al. [6] proposed Video Ladder Networks (VLN) by adding recurrent connections to the ladder network [7]. Similar to ladder networks, VLN employs shortcut connections from the encoder to the respective decoder part, whereby it relieves the deeper layers from modeling details. The VLN architecture achieves a result competitive to VPN [8] which is the state-of-the-art on the synthetic dataset of Moving MNIST. However, the VLN architecture due to its convolutional layers, cannot deal with location-dependent features. Another recurrent network for the task of video prediction was proposed by Michalski et al. [9]. Their PGP network is based on a gated autoencoder and a bilinear transformation model, to learn transformations between pairs of consecutive images ([10], [11]). PGP is fully connected, which results in a large number of parameters. Its convolutional variant Conv-PGP reduces the number of parameters significantly [12], but loses the ability to learn location-dependent features. For the evaluation of Conv-PGP, the authors augmented one-pixel padding to the input to learn a bouncing ball motion in their synthetic dataset.

While VLN and Conv-PGP have shown impressive performance in the task of video prediction, the above analysis shows that the effect of location-dependent features on these two architectures requires further investigation.

3 Location Dependency in VLN Model

The VLN model [6] is a neural network architecture that predicts future frames by encoding the temporal and spatial features of a video. Although it achieves a competitive result in comparison to the state-of-the-art on Moving MNIST, due to the location invariant property of convolution operation, it cannot learn location-dependent features present in the dataset. The network would become unreasonably huge if we wanted to utilize a fully connected layer to allow for learning location-dependent features. Using a fully connected layer would also violate the assumption of weight sharing in the VLN architecture. The same-padding property around the border, which is not analyzed in the original paper, is the reason which allows the network to learn where to mirror digit velocity despite using only convolutional operations. Such a behavior is accidental, though, and should not be treated as a feature.

To demonstrate this limitation of the VLN architecture, we modified the Moving MNIST dataset to Occluded Moving MNIST, similar to what is used by Prémont-Schwarz et al. [13]. As demonstrated in the experiment section, we tested the original one-layer VLN with this dataset and it did not achieve an acceptable result.

To solve this issue, we propose three methods for providing location information to the network. In the first method illustrated in Fig. 1(a), we provide three additional input channels to the network: two gradient channels in x and y direction, starting from 0 and ending with 1, as well as one channel containing the occlusion grid pattern. The occlusion channel is 1 in the occlusion areas and 0 elsewhere. These additional input channels allow the network to infer the location-dependent feature of the border and to utilize the occlusion pattern. In contrast to encoding location features in the original input channel, having additional channels does not alter the original input. Encoding occlusions in a separate channel can be useful, for example, when they are inferred from modalities other than a camera, like a laser scanner.

In the second method (Fig. 1(b) and Fig. 2), we replace the first convolutional layer in the encoder block with a location-dependent convolutional layer:

$$LC(x, y) = A \left(\sum_{i,j} (I(x+i, y+j) * W(i, j) + b) + W'_1(x, y) + W'_2(x, y) \right) \quad (1)$$

where A is the activation function. W and b are the weight and bias of the specified layer, respectively. Note that b can be omitted, but we kept it to make the proposed layer easy to implement on top of an existing convolution layer. $I(x, y)$ is the input vector at the Cartesian position (x, y) and $*$ represents the convolution operator. Note that W'_1 and W'_2 are location-dependent weights that are learned through the training procedure. W'_1 and W'_2 are shared for all convolutional filters, which is done by broadcasting over channel dimension.

In the third method, illustrated in Fig. 1(c), we added location-dependent gradients to the W'_1 and W'_2 :

$$LC(x, y) = A \left(\sum_{i,j} (I(x+i, y+j) * W(i, j) + b) + (L_x(x, y) + W'_1(x, y)) + (L_y(x, y) + W'_2(x, y)) \right) \quad (2)$$

where similar to additional input channels, $L_x(x, y)$ and $L_y(x, y)$ encode location by gradients in x and y directions, respectively. Providing these facilitates the learning of more complex location-dependent biases.

4 Location Dependency in Conv-PGP Model

PGP [9] is designed based on the assumption that two temporally consecutive frames can be described as a linear transformation of each other. In the PGP architecture, by using a Gated AutoEncoder (GAE) as bi-linear model, the hidden layer of mapping units m encodes the transformation.

The fully connected PGP architecture contains a significant number of parameters. To deal with this issue, we utilized its convolutional variant (Conv-PGP), similar to [12], where fully connected layers are replaced by convolutions.

While Conv-PGP reduces the number of parameters significantly, it cannot learn location-dependent features such as the image border anymore. Using valid convolutions prevents, e.g., learning the mirroring motion in the Bouncing Ball dataset. As shown in the experiment section, in the Conv-PGP model, the balls disappear instead of being reflected at the border which indicates that the model is incapable of predicting location-dependent motions.

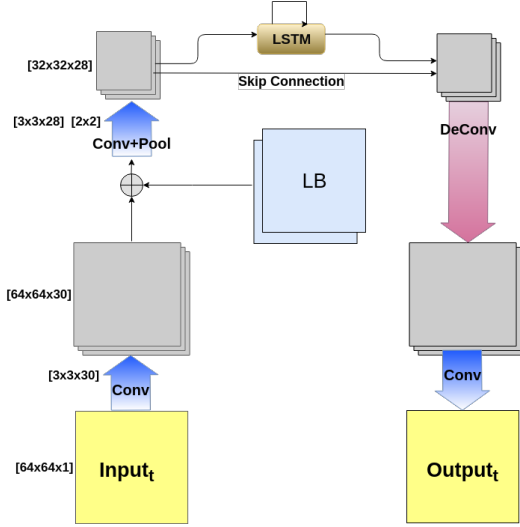


Fig. 2 One-layer location-dependent VLN architecture consisting of two convolution layers in the encoder block, a Conv-LSTM block, and one deconvolution followed by a convolution layer in the decoder part. The trainable location-dependent bias (LB) is applied after the first convolution layer.

To demonstrate this limitation more clearly, we modified the Bouncing Ball dataset. In the Occluded Bouncing Ball dataset, we augmented fixed strides of three pixels to occlude the moving balls as well as invisible lines to mirror the velocity. As shown in the following section, we trained the Conv-PGP with this dataset, and it did not achieve a satisfactory result. To resolve this issue, we applied the three proposed methods for modeling location dependency to Conv-PGP.

5 Experiment

We tested our modified VLN architectures on the Occluded Moving MNIST dataset. Each video in the Occluded Moving MNIST dataset contains 10 frames, with one MNIST digit moving inside a 64×64 patch. Digits are chosen randomly from the training set and placed initially at random locations inside the patch with a random velocity. The frames are filled with occluding vertical and horizontal bars; the distance between them is eight pixels. In addition to that, we added invisible lines to mirror the velocity at a distance of ten pixels from the border.

In our first experiment, we compare the one-layer original VLN architecture on Occluded Moving MNIST with our three proposed solutions:

- VLN-AI: Two location gradient channels and one occlusion channel as additional location encoding inputs (Fig. 1(a)),
- VLN-LDC: Location-dependent bias in the encoder block (Fig. 1(b)), and
- VLN-LDCAI: Location-dependent bias in the encoder block and location gradient channels (Fig. 1(c)).

In our experiment, the first eight frames are predicted using the given frame from the dataset. The last two frames are predicted using the previous network output. Sample results of one-layer original VLN and VLN-LDCAI are depicted

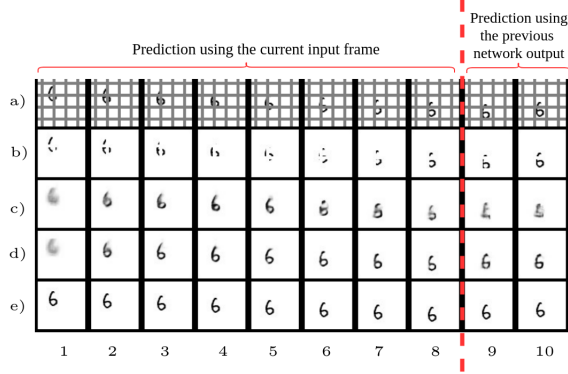


Fig.3 Ocluded Moving MNIST. a) Input frames with visualized occlusion. b) Frames given to the network. c) Predicted frames with the one-layer original VLN. d) Predicted frames with VLN-LDCAI. e) Expected ground truth frames.

in Fig. 3. Sample activations of the Conv-LSTM and the encoder block for both the original VLN and the VLN-LDCAI are shown in Fig. 4. These activations demonstrate that the original VLN cannot infer the location-dependent features while the VLN-LDCAI can learn location-dependent features including the border and the occlusion grid.

Table 1 reports the prediction loss and the number of parameters for the evaluated model variant. It can be observed that all methods to model location dependencies improve performance.

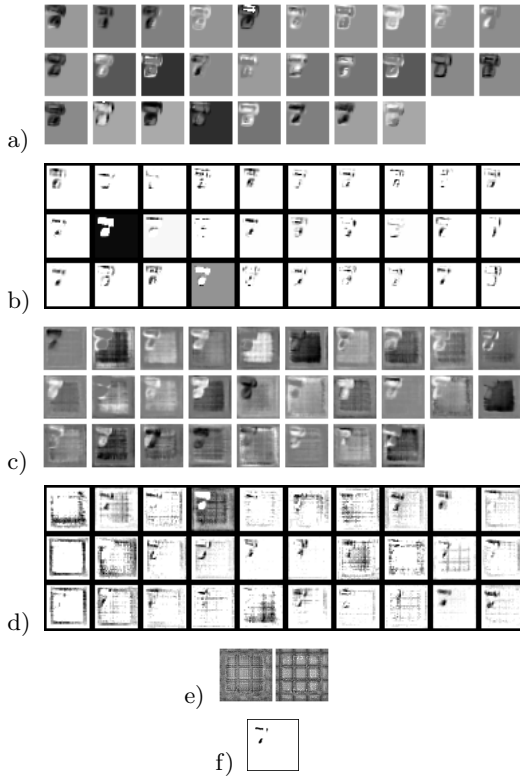


Fig.4 Ocluded Moving MNIST activities. a) Activation layers of Conv-LSTM block in original VLN. b) Activation layers of the encoder block in original VLN. Note that none of the channels can detect the location-dependent features. c) Activation layers of Conv-LSTM block in VLN-LDCAI. d) Activation layers of the encoder block in VLN-LDCAI. e) Learned location-bias channels in VLN-LDCAI. f) Input frame. Note that the VLN-LDCAI automatically inferred location-dependent features.

Table 1 Results of VLN models on Occluded Moving MNIST test dataset.

Model	Prediction test loss (BCE)	Number of parameters
VLN	165.9	90K
VLN-AI	150.7	91K
VLN-LDC	154.7	103K
VLN-LDCAI	153.2	103K

In a second experiment, we compared a one-layer Conv-PGP network with and without the border on the Occluded Bouncing Ball dataset, which is constructed similar to Occluded Moving MNIST. In our experiment, the first three frames are predicted using the given frame from the dataset. The last seven frames are predicted using the previous network output. As illustrated in Figure 5, learning the location-dependent features is crucial for the prediction task. The prediction losses reported in Table 2 show that our proposed one-layer location-dependent Conv-PGP can solve the Occluded Bouncing Ball dataset and yields a much better result than one-layer Conv-PGP.

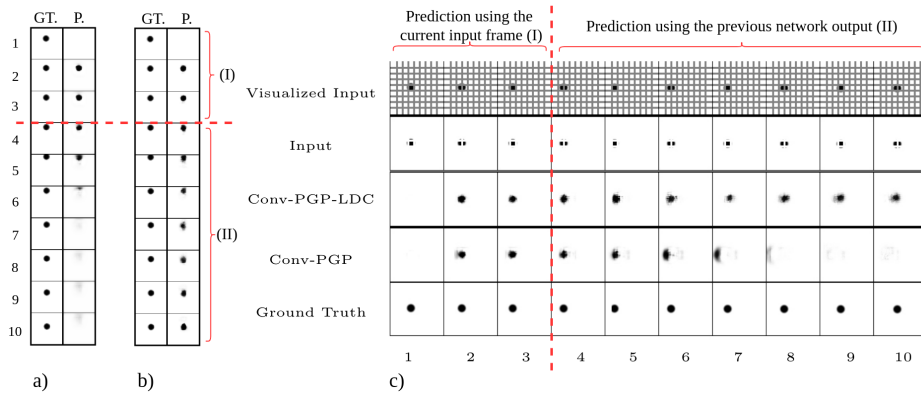


Fig. 5 Bouncing Ball results. a) Conv-PGP. b) Conv-PGP-AI. c) Conv-PGP-LDC and Conv-PGP on Occluded Bouncing Ball dataset.

Table 2 Results of Conv-PGP models on Occluded Bouncing Ball test dataset.

Model	Prediction test loss (BCE)	Number of parameters
Conv-PGP	266.9	39k
Conv-PGP-AI	148.7	40k
Conv-PGP-LDC	139.4	56k
Conv-PGP-LDCAI	143.3	56k

6 Conclusion

Our experiments indicate that location information is a necessity in convolutional architectures for video prediction tasks as, for example, dealing with occlusions in the environment is challenging. To test three proposed variants of learning location-dependent features, we utilized the Occluded Moving MNIST and Occluded Bouncing Ball datasets which mimic occlusions in the real world. The proposed location-dependent inputs and biases allow the VLN and Conv-PGP models to learn more complex location-dependent features than just mirroring velocity at the borders. In contrast to previous approaches, our proposed

learnable location-dependent biases do not assume any predefined underlying feature structure. Our proposed location-dependent convolution layers significantly improve on the results of both one-layer VLN and one-layer Conv-PGP architectures.

In future work, we will explore the proposed methods for general deep convolutional neural network architectures, and test the performance on real-world datasets.

Acknowledgment

This work was funded by grant BE 2556/16-1 (Research Unit FOR 2535 Anticipating Human Behavior) of the German Research Foundation (DFG).

References

1. Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *Preprint arXiv:1511.05440*, 2015.
2. Jörg Wagner, Volker Fischer, Michael Herman, and Sven Behnke. Learning Semantic Prediction using Pretrained Deep Feedforward Networks. *26th European Symposium on Artificial Neural Networks (ESANN)*, 2017.
3. Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with PixelCNN decoders. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.
4. Srinivas SS Kruthiventi, Kumar Ayush, and R Venkatesh Babu. DeepFix: A fully convolutional neural network for predicting human eye fixations. *Preprint arXiv:1510.02927*, 2015.
5. Ghafoorian, Karssemeijer, Heskes, Uden, Sanchez, Litjens, Leeuw, Ginneken, Marchiori, and Platel. Location sensitive deep convolutional neural networks for segmentation of white matter hyperintensities. *Scientific Reports*, 7(1):5110, 2017.
6. Francesco Cricri, Xingyang Ni, Mikko Honkala, Emre Aksu, and Moncef Gabbouj. Video ladder networks. *Preprint arXiv:1612.01756*, 2016.
7. Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3546–3554, 2015.
8. Nal Kalchbrenner, Aaron van den Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. *Preprint arXiv:1610.00527*, 2016.
9. Vincent Michalski, Roland Memisevic, and Kishore Konda. Modeling deep temporal dependencies with recurrent grammar cells. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1925–1933, 2014.
10. Roland Memisevic. Learning to relate images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1829–1846, 2013.
11. Roland Memisevic and Geoffrey E Hinton. Learning to represent spatial transformations with factored higher-order Boltzmann machines. *Neural computation*, 22(6):1473–1492, 2010.
12. Filip De Roos. Modeling spatiotemporal information with convolutional gated networks. Master thesis, Chalmers University of Technology, 2016.
13. Alexander Ilin, Isabeau Prémont-Schwarz, Tele Hotloo Hao, Antti Rasmus, and Harri Valpola. Recurrent ladder networks. *Preprint arXiv:1707.09219*, 2017.