Supplementary Material: MSPred: Video Prediction at Multiple Spatio-Temporal Scales with Hierarchical Recurrent Networks

A Datasets

We evaluate MSPred for different prediction tasks on three video datasets of different levels of complexity, namely Moving MNIST [13], KTH-Actions [10], and SynpickVP. Table 1 summarizes the three datasets used in our work.

Moving MNIST is a standard video prediction dataset containing sequences of two random digits from the MNIST dataset [9] moving with constant speed in a 64×64 grid, and bouncing off the image boundaries. In our experiments, we treat Moving MNIST frames as RGB images, i.e., repeating the MNIST digits across the RGB channels. For the high-level representations, we use Gaussian blobs centered at the digit locations. Despite its simplicity, this dataset is commonly used as a benchmark for video prediction. For training, we randomly generate sequences of 49 frames by sampling two random MNIST digits, a starting position and speed; whereas for testing we use a fixed set containing 10,000 sequences.

KTH-Actions is a dataset consisting of real videos of humans performing one out of six possible actions, namely boxing, hand-clapping, hand-waving, walking, running and jogging. The dataset includes 600 videos of 25 different human actors performing the actions in various indoor and outdoor environments. In our experiments, we downsample the images to a resolution of 64×64. We use 1436 training sequences of length 49 from 16 different actors, whereas for testing we use 824 sequences from the remaining nine actors. In addition to video frames, we use nine human keypoints as intermediate level representations, and a center-point of the person a high-level representation. We generate the ground-truth keypoints using a pretrained OpenPose [1] model for human pose estimation.

SynpickVP is a new synthetic video prediction dataset, consisting of videos of various bin-picking scenarios in which a suction-cap gripper robot moves in arbitrary directions in a box containing different objects. We generate the dataset by selecting sequences from the recently proposed SynPick [6] dataset. We use 1975 training and 200 evaluation sequences containing 29 RGB video frames of size 64×112. This is a challenging video prediction benchmark, since the model needs to capture the motion of the robotic gripper, as well as predict the future arrangement of displaced objects, while still representing a complex and cluttered background. In our experiments on SynpickVP, we train our model to predict image frames at the lowest level in the hierarchy, semantic segmentation maps from the 22 different classes at the intermediate level, and a single-keypoint heatmap for the robotic gripper position at the highest level. Due to the synthetic nature of the dataset, semantic segmentation and object localization annotations are readily available. When evaluating semantic segmentation forecasting, we average the class-wise results into three different categories: *gripper* corresponds to the robot gripper, *static* includes the different objects contained in the box, and *background* corresponds to the red box where objects are placed.

Table 1: Summary of the datasets used in our experiments, including the size of the dataset splits and the type of high-level representations used for each dataset.

Dataset Name	Img. Size	# Train	# Test	Mid-Level Rep.	High-Level Rep.
Moving MNIST [13]	(3, 64, 64)	-	10.000	Digit Blob	Digit Position
KTH-Actions [10]	(3, 64, 64)	1.436	824	Human Pose	Person Position
SynpickVP [6]	(3, 64, 112)	1.975	200	Segmentation Maps	Gripper Position

B Evaluation Metrics

We employ several evaluation metrics designed for different tasks in order to evaluate the predictions from the different MSPred decoder heads. For future frame prediction, we compute several popular metrics which measure the visual similarity between the predicted and ground-truth video frames. Furthermore, we employ different metrics to evaluate the ability of our model to make high-level structured predictions, such as human poses or semantic segmentation maps. For all metrics, we average the results across all predicted frames or high-level structured representations.

Image Similarity Metrics: We evaluate our models for future frame prediction using four popular metrics, namely MSE, PSNR, SSIM [15], and LPIPS [16]. MSE, PSNR and SSIM measure pixel or statistical differences between predicted and target images. However, they have been proven to correlate poorly with human perception, favoring blurred predictions over more detailed, though imperfect, generations [9, 16]. Therefore, we favor LPIPS in our experiments, which measures the distance between CNN feature maps, and has been shown to better correlate with human judgment.

Pose and Keypoint Prediction Metrics: MSPred forecasts future human poses at its intermediate level on the KTH-Actions dataset. Given a predicted heatmap representing the location of a body joint, we extract the position coordinates by taking the location with maximum value of the heatmap, provided that the maximum value exceeds a certain threshold. Through empirical validation, we set the threshold value to 0.05.

In order to assess our model's performance for human pose forecasting, we employ three popular metrics. *Mean Per Joint Position Error* (MPJPE) calculates the average $\ell 2$-norm across predicted and target joints. *Percentage of Detected Joints* (PDJ) measures the fraction of the correctly estimated joints among the joints present in the ground-truth pose. A predicted keypoint is marked as a correct detection if its distance from the respective target keypoint does not exceed a certain threshold. We select this threshold as 20% of the ground-truth person's height [8]. Similarly, *Percentage of Correct Keypoints* (PCK) measures the fraction of correctly detected joints among the overall predicted joints. Additionally, we also compute summary statistics for the PCK metric over a range of thresholds. We evaluate the *Average Precision* (AP) as the mean PCK values computed over a range of thresholds 0.1, 0.2, ..., 0.5.

Segmentation Metrics: We predict semantic segmentation maps as the intermediate-level representation on the SynpickVP dataset. We evaluate our predicted segmentation maps using two popular evaluation metrics. *Pixel accuracy* (Acc) measures the fraction of correctly

Table 2: DCGAN Discriminator Encoder

Layer	Size	Activation	Comment
Conv 4x4	64	LeakyReLU	Stride 2
Conv 4x4	128	LeakyReLU	Stride 2
Conv 4x4	256	LeakyReLU	Stride 2
Conv 4x4	512	LeakyReLU	-

Table 3: VGG16-Like Encoder

Layer	Size	Activation
2x Conv 3x3	64	LeakyReLU
MaxPool 2x2	-	-
2x Conv 3x3	128	LeakyReLU
MaxPool 2x2	-	-
2x Conv 3x3	256	LeakyReLU
MaxPool 2x2	-	-
2x Conv 3x3	512	LeakyReLU

classified pixels in the image, whereas *Intersection over Union* (IoU) is computed by dividing the corresponding number of correctly estimated pixels, i.e. the area of overlap between predicted and ground-truth segments, by the area of union of the very segments.

We compute the average Acc and IoU metrics for three subsets of the classes. More precisely, we average the metrics separately for three object categories: robot gripper, static objects placed on the box, and the red box itself.

C Implementation Details

In this section we provide further implementation details of MSPred (Section C.1), and the hyper-parameter values used in our experiments (Section C.2). Additionally we discuss the implementation of the SVG' baseline (Section C.3). Our codebase is implemented using the PyTorch [5] deep learning framework. We run our experiments on an NVIDIA A6000 GPU with 48 GiB RAM.

C.1 MSPred Architecture Details

Encoder: In order to ensure a fair comparison with baseline methods, the encoder is implemented following the SVG [2] architecture. For the Moving MNIST dataset, the encoder follows the DCGAN discriminator [7] architecture, whereas for KTH-Actions and SynpickVP we employ VGG16-like [12] modules. The architectures of both encoders are depicted in Tables 2 and 3, respectively. All convolutional layers use padding 'SAME', include a bias weight, and are followed by batch normalization [3].

Decoder: The decoder in MSPred is implemented as a mirrored version of the corresponding encoder. In the DCGAN-like decoder, feature maps are upsampled via transposed convolutions, whereas in the VGG-like decoder upsampling is achieved via nearest neighbor interpolation. The higher-level decoder heads are each composed of two convolutional blocks with the same structure as the decoder.

Predictor: Our predictor module uses four ConvLSTM [11] cells for each of the three levels in the hierarchy, each with 128 kernels of size 3×3. The lowest-level LSTM processes all inputs, whereas the higher-level LSTMs process one out of every T_1 and T_2 inputs respectively.

Table 4: Hyper-parameter values used for each dataset in our experiments

Hyper-Parameter	Moving MNIST	KTH-Actions	SynpickVP
C	9	9	9
T_1	4	4	2
T_2	8	8	4
Learning rate	10^{-4}	$3 \cdot 10^{-4}$	$5 \cdot 10^{-4}$
Batch size	16	12	12
Num. Epochs	350	800	200
λ_1	2.5	1.4	2.0
λ_2	2.5	0.2	0.3
β	10^{-4}	$5 \cdot 10^{-5}$	10^{-4}

Stochastic Component: The prior ($LSTM_\psi$) and posterior ($LSTM_\phi$) modules are implemented as a single-cell ConvLSTM with 64 kernels of size 3×3, followed by a convolutional layer mapping the feature maps into the desired latent dimensionality. Inspired by SVG [14], we sample latent tensors with 10, 24 and 32 channels for the Moving MNIST, KTH-Actions, and SypickVP datasets, respectively.

C.2 Hyper-Parameters

The hyper-parameters used in our experiments are reported in Table 4. We report the specific values for the experiments on each of the datasets.

C.3 SVG'

As described in Section 4.4 of the paper, we train a specialized baseline *SVG'*, based on a modified SVG-LP [2] model, which predicts high-level representations (e.g. human poses or semantic segmentation) conditioned on input video frames. SVG' follows the same architecture as SVG-LP, but we apply some modifications to adapt the model for the tasks of pose and semantic segmentation forecasting, and for a fair comparison with MSPred. First, the linear LSTM recurrent blocks are replaced by ConvLSTMs operating with a period of T_1, i.e., processing every T_1-th input. Second, the number of output channels is changed from three to nine for KTH-Actions, and to 22 for SynpickVP. Finally, since there are no predicted image frames to be fed back into the model, we design SVG' to be autoregressive in the feature space, i.e., the output of the predictor module becomes its input at the subsequent time step.

D Qualitative Results

In Figures 1–4, we qualitatively compare several video prediction models for the task of future frame prediction on the Moving MNIST, KTH-Actions and SynpickVP datasets, respectively. Figures 5 and 6 depict additional examples of multi-scale prediction on the KTH and SynpickVP datasets, respectively. Further images and animations can be found in the project website[1].

[1] https://sites.google.com/view/mspred/home

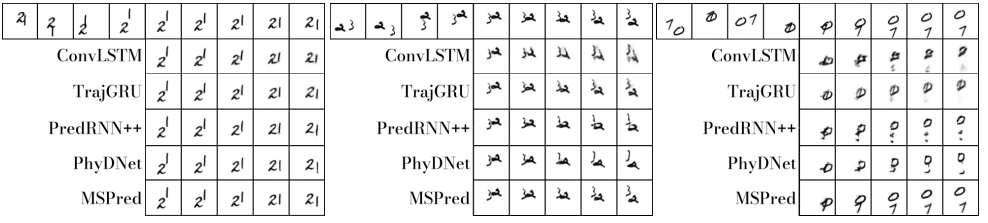

Figure 1: Qualitative results on the Moving MNIST dataset. Top row corresponds to ground truth frames. We display four seed frames and five predictions for three test-set sequences. In general, all compared methods achieve good frame predictions. However, only MSPred accurately resolves challenging cases in which digits overlap. Colors are inverted to improve the visualization.

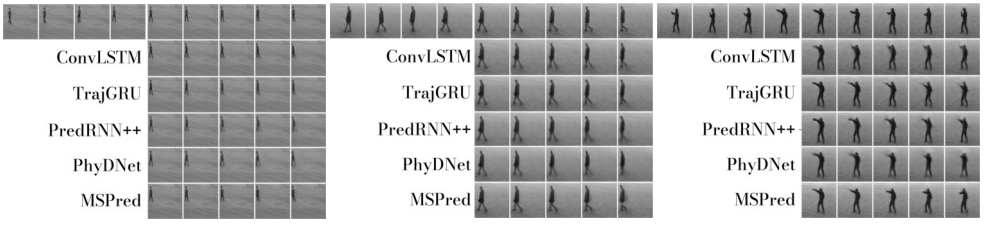

Figure 2: Qualitative results on the KTH-Actions dataset. Top row corresponds to ground truth frames. We display four seed frames and five predictions for three test-set sequences. MSPred achieves the sharpest and more accurate predictions among the compared methods.

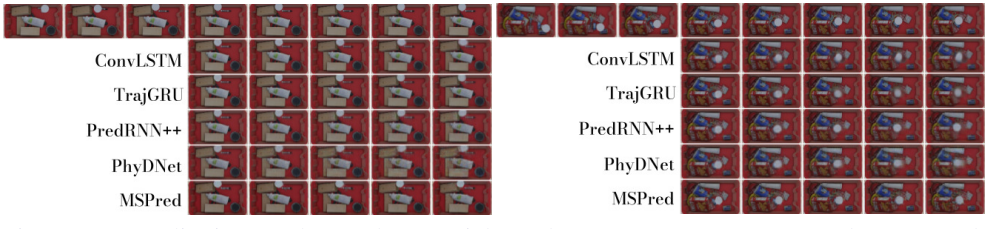

Figure 3: Qualitative results on the SynpickVP dataset. Top row corresponds to ground truth frames. We display three seed frames and five predictions for two test-set sequences. MSPred qualitatively outperforms the compared methods, achieving sharp reconstructions, whereas the baseline methods tend to blur the predictions.

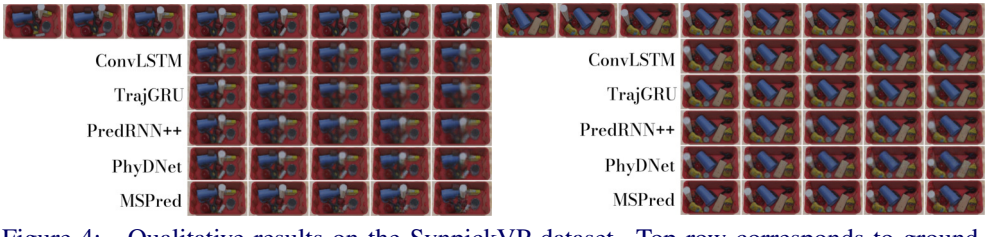

Figure 4: Qualitative results on the SynpickVP dataset. Top row corresponds to ground truth frames. We display three seed frames and five predictions for two test-set sequences. MSPred qualitatively outperforms the compared methods, achieving sharp reconstructions, whereas the baseline methods tend to blur the predictions.

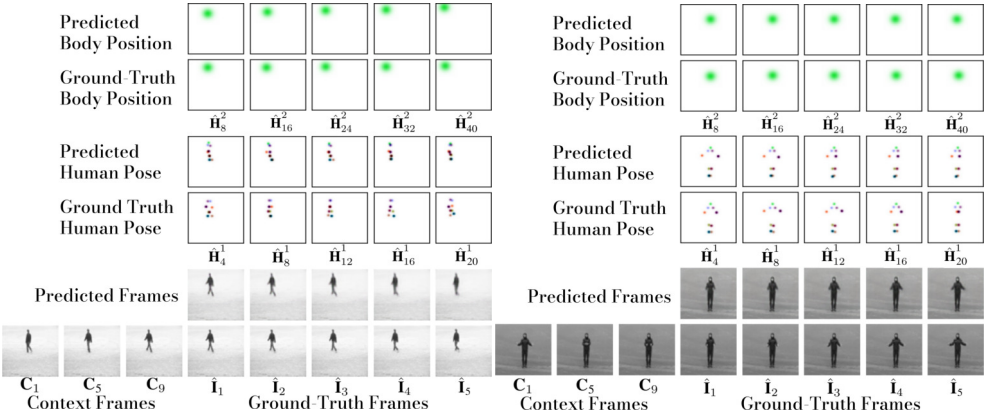

Figure 5: Predictions of different level of abstraction on the KTH-Actions dataset. We display three seed frames and five targets and predictions for each decoder head. MSPred forecasts frames on short time horizons, while also predicting human poses and person locations longer into the future using coarser temporal resolutions.

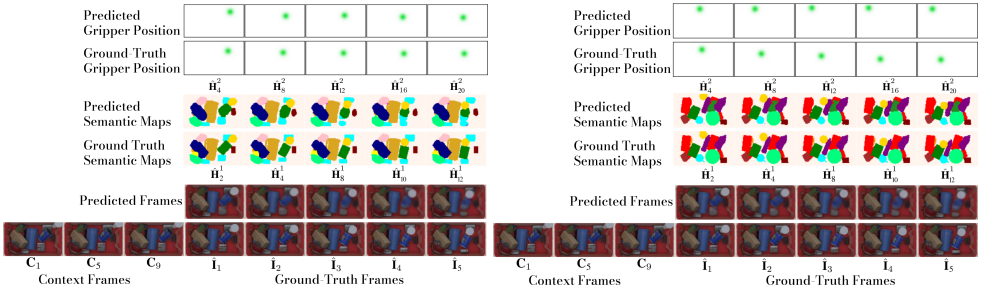

Figure 6: Predictions of different levels of abstraction on the SynpickVP dataset. We display three seed frames, and five targets and predictions for each decoder head. MSPred forecasts frames on short time horizons, while also predicting the semantic segmentation of the scene and the gripper location long into the future using coarser temporal resolutions.

References

[1] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7291–7299, 2017.

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

[3] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[4] Yann LeCun, Corinna Cortes, and Christopher JC Burges. The MNIST database of handwritten digits. *URL http://yann. lecun. com/exdb/mnist*, 10:34, 1998.

[5] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *International Conference on Neural Information Processing Systems Workshops (NeurIPS-W)*, 2017.

[6] Arul Selvam Periyasamy, Max Schwarz, and Sven Behnke. SynPick: A dataset for dynamic bin picking scene understanding. In *IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 488–493, 2021.

[7] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Learning Representations, (ICLR)*, 2016.

[8] Ben Sapp and Ben Taskar. Modec: Multimodal decomposable models for human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3674–3681, 2013.

[9] Umme Sara, Morium Akter, and Mohammad Shorif Uddin. Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study. *Journal of Computer and Communications*, 7(3):8–18, 2019.

[10] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local SVM approach. In *IEEE International Conference on Pattern Recognition (ICPR)*, volume 3, pages 32–36, 2004.

[11] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wangchun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[12] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations, (ICLR)*, 2015.

[13] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using LSTMs. In *International Conference on Machine Learning (ICML)*, pages 843–852, 2015.

[14] Ruben Villegas, Arkanath Pathak, Harini Kannan, Dumitru Erhan, Quoc V Le, and Honglak Lee. High fidelity video prediction with large stochastic recurrent neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[15] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[16] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.