Supplementary Material for
Disentangling Propagation and Generation for Video Prediction

A. Implementation Details

General architectural configuration We adapt our architectures from Zhu *et al*. [11] and Johnson *et al*. [5]. For all experiments described in the main paper, we use 5 blocks for the encoder and 5 blocks for the decoder. Below, we follow the naming convention used in their Github repositories to describe our general architectural configuration.

Let `cMsN-K` denote a $M \times M$ Conv-BN-Activation layer with stride N and K filters. We use Inplace-ABN [10] to reduce the memory consumption. Further, let us define a encoder basic block `eM-K` by cascading `cMs1-K` with another downsample convolution block `cMs2-K` where ReLU is used[1]. The basic decoder block `dM-K` consists of a nearest-neighbor upsample layer followed by two `cMs1-K` layers in which activation layers are chosen as LeakyReLUs of slope 0.2.

Flow Predictor Our flow predictor $\mathcal{F}$ could be defined as:

`e7-64, e5-128, e5-256, e3-512, e3-512, d3-512, d3-512, d3-256, d3-128, d3-2,`

where the last output layer has no activation, i.e., the flow prediction network regresses unconstrained displacement values for each coordinates. Raised by [4], we also empirically confirmed large kernel sizes, in first several layers, help the training to converge.

Occlusion Inpainter Our occlusion inpainter uses the same architectural parameters as in the flow predictor. The only differences here are that: (1) we replace the normal convolution operators with partial convolution operators in all `eM-K`'s and fusion convolution operators in all `dM-K`'s; (2) we replace `d3-2` with `d3-3`, where Tanh activation is used to bound the output value between -1 and 1.

B. Training Details

Here we specify more training details to supplement what we have described in the main paper. To train the flow predictor, we start from the learning rate at 10^{-4} and decay it by $1/10$ at the half of the training epochs, then repeat it again at the $3/4$ of the training epochs. The occlusion inpainter is trained from 10^{-3} and scheduled with the same decay strategy. We train our flow predictor, and occlusion inpainter for 200 epochs, and 800 epochs on CalTech Pedestrian dataset [2]. For KITTI Flow dataset [9], they are trained for 500 and 1000 epochs, respectively.

[1] All ReLU units are approximated by LeakyReLUs of slope 0.01 to be compatible with Inplace-ABN [10]

C. Supplementary Results

In this section, we include more results to supplement our main paper. We include more qualitative results for both Next-Frame Prediction and Multi-Frame Prediction. We also include the quantitative results for SSIM evaluations on Multi-Frame Prediction tasks on KITTI Flow dataset. To better assess our prediction results please refer to our website [2].

Next-Frame Prediction More qualitative results are shown in Figure S1 and S2, respectively. The experiment settings are consistent with the setups we established in the main paper.

Multi-Frame Prediction We here show comparison results for Multi-Frame Predictions in Figure S3. The experiment setting is 4-in 8-out prediction task. Compared to prior work [3] whose flow prediction degrades dramatically after a few time steps, our model can remain high fidelity even at the last several frames. We here also include the quantitative results for SSIM evaluations in S4.

D. Supplementary Ablations

In Table S1, we add one more ablation study to resolve the concern about the auxiliary losses we applied to our model. We build three groups of comparison experiment by removing perceptual and style losses, segmentation loss, or replacing our fusion decoder with normal partial convolutions as in [6]. All generators are trained using the same oracle model used in motion ablation studies. Removing perceptual and style losses does not hurt the performance of PSNR, but leads to large degeneration in structural and perceptual metrics. On the other hand, removing segmentation loss and our fusion decoding blocks results in performance drops in all metrics.

[2] https://sites.google.com/view/fgvp

(a) $\mathbf{x}_{t-5}$ (b) $\mathbf{x}_{t-2}$ (c) PredNet [8] (d) ContextVP [1] (e) DPG (Ours) (f) $\mathbf{x}_t$

Figure S1: More qualitative comparisons for 10-in 1-out Next-Frame Prediction on CalTech Pedestrian dataset.

(a) DVF [7] (b) CtrlGen [3] (c) DPG (Ours) (d) $\mathbf{x}_t$

Figure S2: More qualitative comparisons for 4-in 1-out Next-Frame Prediction on KITTI Flow dataset.

(a) $\mathbf{x}_{t-3}$ (b) $\mathbf{x}_{t-1}$ (c) $\mathbf{x}_{t+1}$ (d) $\mathbf{x}_{t+3}$ (e) $\mathbf{x}_{t+5}$ (f) $\mathbf{x}_{t+7}$

Figure S3: Qualitative comparisons for 4-in 8-out Multi-Frame Prediction on KITTI Flow dataset.

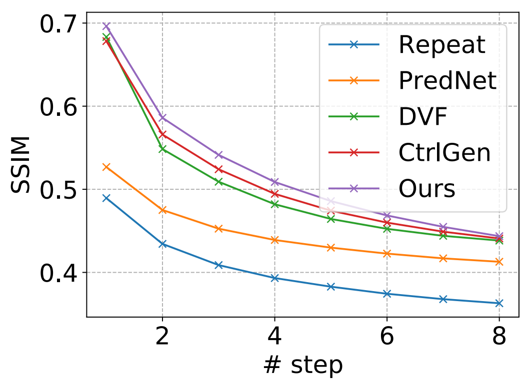

Figure S4: SSIM↑ quantitative results for 4-in 8-out Multi-Frame Prediction on KITTI Flow dataset.

Method	**PSNR**↑	**SSIM**↑	**LPIPS**↓ $(\times \mathbf{10^{-2}})$
w/o p+s	$\mathbf{23.3}$	0.748	17.9
w/o seg	22.6	0.766	10.2
w/o fus	22.42	0.760	10.8
all	$\mathbf{23.3}$	$\mathbf{0.786}$	$\mathbf{9.9}$

Table S1: Supplementary ablation study on auxiliary losses in our model.

References

[1] Wonmin Byeon, Qin Wang, Rupesh Kumar Srivastava, and Petros Koumoutsakos. Contextvp: Fully context-aware video prediction. In *ECCV*, 2018. 2

[2] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, 2012. 1

[3] Zekun Hao, Xun Huang, and Serge Belongie. Controllable video generation with sparse trajectories. In *CVPR*, 2018. 1, 2, 3

[4] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. *arXiv preprint arXiv:1712.00080*, 2017. 1

[5] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 1

[6] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. *arXiv preprint arXiv:1804.07723*, 2018. 1

[7] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, 2017. 2

[8] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *ICLR*, 2017. 2

[9] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015. 1

[10] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *CVPR*, 2018. 1

[11] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017. 1