Appendix: Multi-task Learning with Future States for Vision-based Autonomous Driving

Inhan Kim[1[0000−0002−1426−108X]], Hyemin Lee[1[0000−0002−1899−7211]], Joonyeong Lee[1[0000−0003−3217−2168]], Eunseop Lee[1[0000−0002−6027−2863]], and Daijin Kim[1[0000−0002−8046−8521]]

{kiminhan, lhmin, joonyeonglee, eunseop90, dkim}@postech.ac.kr

Department of Computer Science and Engineering, POSTECH, Korea[1]

S.1 Conditional Predictive-coding Networks

In this section, we will introduce the Conditional Predictive-coding Networks (CPredNet), which is the PredNet [1] extension with high-level navigational command. Similar to the PredNet each generative module has four parts: a recurrent representation layer (R_l), an input convolutional layer (A_l), a prediction layer ($\hat{A}_l$), and an error representation unit (E_l). Compared to the PredNet, the conditional branches in the CPredNet are applied to the first generative module. Because the best generative performing models tend to have a loss solely concentrated at the lowest layer, we set $\lambda_0 = 1$ and $\lambda_{l>0} = 0$ [1]. The full set of update rules at time t are shown in Equations (1)-(5).

$$A_l^t = \begin{cases} I_t & if\, l = 0 \\ MAXPOOL(RELU(CONV(E_{l-1}^t))) & l > 0 \end{cases} \tag{1}$$

$$\hat{A}_{l,c}^t = \begin{cases} RELU(CONV(R_l^t), G(c_t))) & if\, l = 0 \\ RELU(CONV(R_l^t))) & l > 0 \end{cases} \tag{2}$$

$$E_l^t = \begin{cases} [RELU(A_l^t - \hat{A}_{l,c}^t); RELU(\hat{A}_{l,c}^t - A_l^t)] & if\, l = 0 \\ [RELU(A_l^t - \hat{A}_l^t); RELU(\hat{A}_l^t - A_l^t)] & l > 0 \end{cases} \tag{3}$$

$$R_l^t = CONVLSTM(E_l^{t-1}, R_l^{t-1}, DECONV(R_l^{t+1})) \tag{4}$$

$$L_c = \sum_t \lambda_t \sum_l \lambda_l E_l^t \tag{5}$$

Based on the command, the representation layer is fed into the selected convolution layer by $G(c_t)$ in Equation (2). To avoid the drawback of the upscaling by interpolation, which only uses neighborhood values, deconvolution is used to reconstruct a larger representation in Equation (4) [2]. The training loss is defined in Equation (5) with weighting factors by time, λ_t, and layer, λ_l. As an enhanced generation mechanism, we employ two CPredNets. CPredNet$_{next}$ predicts the next frame with actual frames as input, and CPredNet$_{extra}$ uses previous prediction as input for extrapolation frames [1]. The first input to CPredNet$_{extra}$ is $\hat{I}_{t+1}$, which is generated by CPredNet$_{next}$ and the other inputs are previous frames generated by CPredNet$_{extra}$. The learned cell states of the CPredNet$_{next}$ are shared into CPredNet$_{extra}$ to inject temporal dynamics of actual frames.

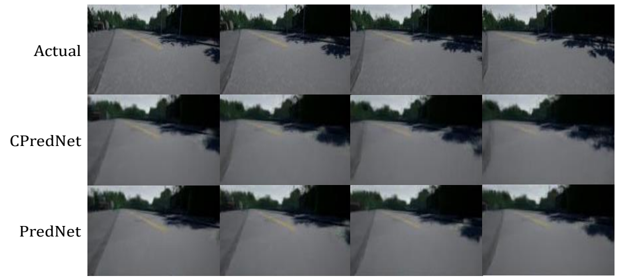

Fig. 1. Extrapolation sequences generated by CPredNet$_{extra}$ and PredNet$_{extra}$.

Table 1. Evaluation of next and extrapolation frame predictions on CARLA dataset.

Model	**MSE**$_{next}$	**MSE**$_{extra}$	**SSIM**$_{next}$	**SSIM**$_{extra}$
CPredNets	$\mathbf{3.17{\times}10^{-3}}$	$\mathbf{4.81{\times}10^{-3}}$	**0.918**	**0.873**
PredNets	$3.31{\times}10^{-3}$	$5.06{\times}10^{-3}$	0.909	0.851

S.2 Comparison between the CPredNet and PredNet

In Fig. 1, we show the results of PredNets and CPredNets in a curve scenario. For this comparison, we modified the PredNet to the have same overall architecture and training scheme as the CPredNet. The second and third row show the generated frames by CPredNet$_{extra}$ and PredNet$_{extra}$ network respectively. Despite blurriness, both models capture some key structure, such as lane, road, and curb. However, the CPredNet results have more detailed information. For example, in the second sequence, the shape of the curb is more accurately generated than the sequence shown in the third row.

To prove this quantitatively, we evaluated the prediction error in terms of Mean Squared Error (MSE) and the Structural Similarity Index Measure (SSIM) (Table 1). MSE$_{next}$ and SSIM$_{next}$ are evaluated with frames generated from the PredNet$_{next}$ and the CPredNet$_{next}$. In addition, MSE$_{extra}$ and SSIM$_{extra}$ are evaluated using extrapolation frames generated from the PredNet$_{extra}$ and the CPredNet$_{extra}$. As expected, both models slightly outperformed the baselines on both measures in terms of evaluating pixel-level predictions.

S.3 Driving Video Clips

We record the driving video clips for the "Dense Traffic" tasks on *NoCrash* benchmark. Due to a limitation of the size of files, the video clips are can be seen at [3–8].

References

1. Lotter, W., Kreiman, G., Cox, D.: Deep predictive coding networks for video prediction and unsupervised learning. arXiv preprint arXiv:1605.08104 (2016)
2. Zhong, J., Cangelosi, A., Zhang, X., Ogata, T.: Afa-prednet: The action modulation within predictive coding. In: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE (2018) 1–8
3. Kim, I.: Fasnet carla town02 failure due to accident. https://youtu.be/Dlx_N7ZOdAY (2020)
4. Kim, I.: Fasnet carla town02 failure due to traffic congestion. https://youtu.be/-ptyqjaWU6Q (2020)
5. Kim, I.: Fasnet carla town02 success. https://youtu.be/0b0Z730QK40 (2020)
6. Kim, I.: Fasnet carla town01 failure due to traffic congestion. https://youtu.be/Sdz-Pvyjy5M (2020)
7. Kim, I.: Fasnet carla town01 failure due to accident. https://youtu.be/ankz35TjiXo (2020)
8. Kim, I.: Fasnet carla town01 success. https://youtu.be/8ydPhNH11V4 (2020)