Supplementary Materials: Online Adaptive Temporal Memory with Certainty Estimation for Human Trajectory Prediction

Anonymous WACV submission

Paper ID 1252

In this supplementary material, we provide the descriptions of PIE [3] and JAAD [1] datasets, used in our experiments (Section 1). We then present the training and testing procedure in Section 2, followed by the implementation details in Section 3. Lastly, we provide additional qualitative results in Section 4.

1. Datasets

PIE [3] datasets consist of long continuous sequences 1800 pedestrian trajectories. The pedestrian bounding boxes are annotated at 30Hz. The dataset covers 6 hours of driving footage captured with calibrated monocular dashboard camera. The entire dataset was recorded in downtown Toronto, Canada during daytime under sunny/overcast weather conditions. JAAD[1] dataset contains sequence of 5-10 second long. Videos were recorded in several locations in North America and Europe under different weather conditions. 2800 pedestrian trajectories captured from dash cameras annotated at 30Hz. Wide range of pedestrian behaviors in different locations: street intersections with high foot-traffics. narrow streets, wide boulevards with fewer pedestrians. Since JAAD dataset consists of multiple sequences, we concatenate these sequences as a continuous stream for testing. Although this could result in abrupt scene context changes, which potentially happen in realistic driving scenarios, it does not alternate the original results of the native predictors. We used the same train/test splits provided by these datasets.

2. Training and Testing Procedure

The training process is divided into several steps as follows:
(1) We first train a predictor on a train dataset following the common experiment setups [5, 3] without modifying the predictor.
(2) We then train our motion encoder, prediction encoder and motion decoder jointly for reconstruction task using the following $L2$ loss: $\mathcal{L}(Y, \hat{Y}_t) = \sum_{i=1}^{N}(\hat{Y}_t - Y_t)^2$. The architecture is shown in Figure 1. Note that the architecture is similar to our framework but without the memory module, and the certainty-based selector. This allows us to not only reconstruct the future trajectory from those from memory in later stage, but also to take into account of the predictor's prediction behaviors.

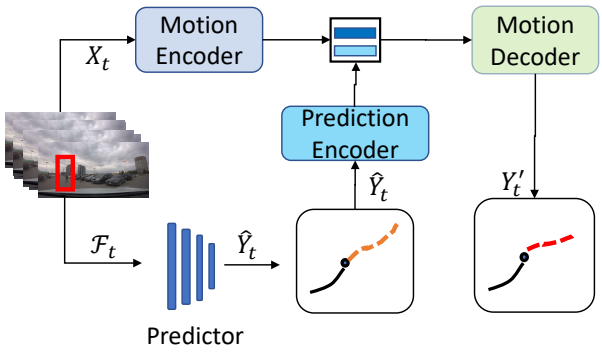

Figure 1. The motion encoder, prediction encoder, and motion decoder are trained for reconstruction task.

(3) Once the encoders/decoder are trained to enable the reconstructive ability, we plug the memory module into the framework, and train the entire framework with train data. This is to allow the encoder/decoders learn temporal motion from memory. The memory values are also initialized in this step and ready for testing.
(4) We train the certainty-based selector module on the train dataset. As mentioned in the main paper, the truth labels (0, 1) are needed to train the selector. Thus, we first generate the labels using the indicator function $\mathbb{1}(Y'_t, \hat{Y}_t) = \mathbb{1}(||Y'_t - Y_t||_2^2 < ||\hat{Y}_t - Y_t||_2^2)$ on train dataset, where the ground-truth trajectory Y_t is accessible. Then, the selector is trained separately using the binary cross entropy loss function [4].

Testing and adaptation. The testing and adaptation are performed in an online fashion, where the frames are continuous within a video sequence. For each coming video frame t, our framework will predict the future trajectory $\dot{Y}_t$ of all pedestrians present in that frame. For adaptation, we collect the pairs of $\{X_t, Y_t\}$ to be encoded into memory using the memory write operation. To further enhance the adaptive ability, the decoder's network weight is also online updated using the recent testing sample with the reconstruction loss.

3. Implementation Details

Predictors. We use the implementation of two predictors: BiTrap [5] and PIE [3], which can be downloaded from their official github pages: https://github.com/umautobots/bidireaction-trajectory-prediction, and https://github.com/aras62/PIEPredict. BiTrap supports multi-modal outputs with its stochastic model; however, we only focus of deterministic model, which outputs a single prediction, and set number of prediction of this predictor to 1. We use the same other configurations as mentioned in their github pages.

We implemented our framework using PyTorch [2]. The experiments were conducted on 4 GPU Tesla P100-SXM2 with 16GB memory each. In our encoders and decoder, we use Conv1D with channel input size is 4, channel output size is 16, stride and padding are set to 1. The GRU has input size of 16 and hidden state size of 48. In the certainty-based selector, we use one layer perception with hidden size of 24. Each training step mentioned in Section 2 is run with 100 epoches, the base learning rate is 0.0001 with Adam optimizer [6]. The batch size during training is 32, while during online testing we set it to 1.

4. Additional Qualitative results

In this section, we present our prediction results in comparison with the native predictor (BiTrap) in different scenarios which commonly occur in ego-centric views. Figure 2 show examples that the our framework produces more accurate predictions compared to predictor's in scenarios where the pedestrian remains similar speed and direction when crossing streets (Figure 2a), or when groups of crossing pedestrians share simmilar motions (Figure 2b). In these scenarios, we can see that the trajectories encoded in memories (left figures in each cases) are very simmilar to the ground-truth ones. Thus, our prediction is more accurate than those from the predictor.

Figure 3 shows another set of examples, where the memory consists of encoded trajectories that are dissimilar with the target pedestrians' motion. Some of these interesting scenarios are presented as follows. Figure 3a show an example where the ego-vehicle abruptly accelerate its speeds, which cause a large motion displacement in current target's trajectory. Figure 3b show an scenario where there are various motions in the intersections. Figure 3c shows an example of a ego-vehicle makes a right turn, which then causes a large motion in pedestrian's movements. Lastly,it is common that a new pedestrian could appear far from the camera and this pedestrian's motion is relatively smaller than others in memory. In these scenarios, we observe that the memory's information will not be helpful to predict future movements. However, our certainty-based selector is capable of mitigate this problems by deciding to use the predictor's prediction as the final prediction.

Lastly, we present some failure cases of our framework. These cases usually happen in scenarios that current movement of the target pedestrian is different from those in memory and our CS is not able to detect the differences. For example, Figure 4a) shows an example when pedestrians started to cross the streets, or there's accelerated speed by the pedestrian in Figure 4b. These failures indicate that despite of the success of certainty-based selector, other visual features could be used to enhance our certainty estimation module. We leave this for future research.

References

[1] Iuliia Kotseruba, Amir Rasouli, and John K Tsotsos. Joint attention in autonomous driving (jaad). *arXiv preprint arXiv:1609.04741*, 2016. 1

[2] 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. 2017. 2

[3] Amir Rasouli, Iuliia Kotseruba, Toni Kunic, and John K Tsotsos. Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6262–6271, 2019. 1, 2

[4] Usha Ruby and Vamsidhar Yendapalli. Binary cross entropy with deep learning technique for image classification. *Int. J. Adv. Trends Comput. Sci. Eng*, 9(10), 2020. 1

[5] Yu Yao, Ella Atkins, Matthew Johnson-Roberson, Ram Vasudevan, and Xiaoxiao Du. Bitrap: Bi-directional pedestrian trajectory prediction with multi-modal goal estimation. *IEEE Robotics and Automation Letters*, 6(2):1463–1470, 2021. 1, 2

[6] Zijun Zhang. Improved adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, pages 1–2. Ieee, 2018. 2

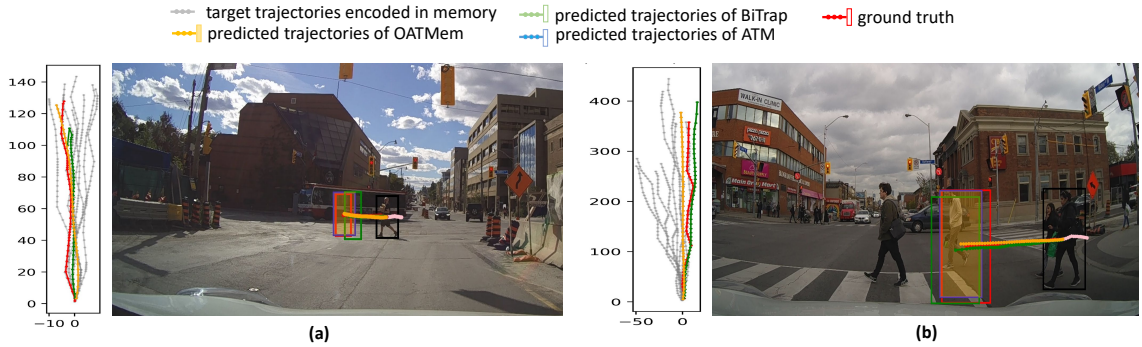

Figure 2. Results in scenarios where past motion encoded in trajectories are similar the target pedestrian's movement, and thus helpful for improving the predictor's trajectory. (a) pedestrians crossing the street. (b) a group of crossing pedestrians.

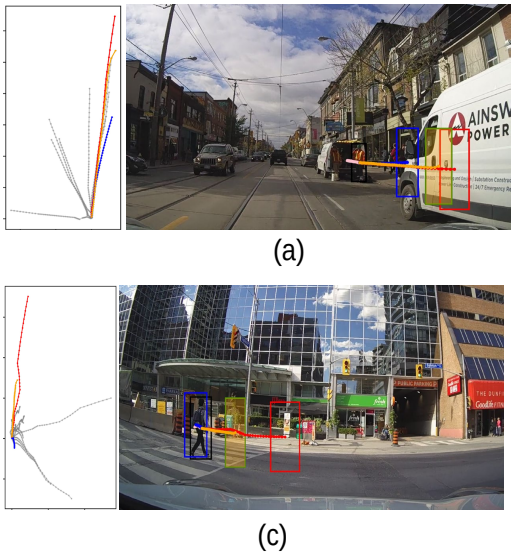

Figure 3. Results in scenarios where past motion encoded in trajectories are dissimilar with the target pedestrian's movement, and thus not helpful for improving the predictor's trajectory. (a) the ego-vehicle abruptly accelerates speed; (b) various pedestrian's motions in intersection, (a) the vehicle abruptly turns right; (d) new pedestrians appear far in distance. Our certainty-based selector has successfully selected the native predictor's predictions as finals.

(a) (b)

Figure 4. Failure cases where the selector failed to select the predictor as final prediction. Although the memory's prediction highly correlates with those in the past (stored in memory), the motion changes occur and thus, the memory's information is not helpful.