Online Adaptive Temporal Memory with Certainty Estimation for Human Trajectory Prediction

Manh Huynh, Gita Alaghband
Department of Computer Science and Engineering
University of Colorado Denver
{manh.huynh,gita.alghband}.ucdenver.edu

Abstract

Pedestrian trajectory prediction is an essential component of autonomous systems and robot navigation. Recent research has shown promising predictive performance by designing prediction networks to model a variety of motion-related features. Different from existing works, our focus is on designing a novel online adaptation framework (OATMem) to exploit the temporal similarities among trajectory samples encountered during testing to improve the prediction accuracy of any such models (i.e., predictors) without knowing the details of these predictors. Our framework consists of two novel modules: an augmented temporal observation-target memory network (ATM) and a certainty-based selector (CS). Inspired by the concept of key-value memory networks [16], ATM is proposed to learn the temporal information from short-term past frames by encoding the trajectory samples of past pedestrians in form of observation-target (i.e., key-value) during testing. In addition, we propose a certainty-based selector (CS) to enhance the predictive ability of our framework under scenarios where there are large temporal dissimilarities between current pedestrians' movements and those stored in memory. In dynamic scenes, these scenarios commonly occur due to abrupt changes in contexts, such as camera motions, scene contexts, and pedestrians' behaviors. We extensively evaluate our framework in commonly-used datasets: JAAD [12] and PIE [19] and show that our framework significantly improves the prediction accuracy of state-of-the-art models. Finally, in-depth studies are conducted to show the importance of each proposed component.

1. Introduction

Pedestrian trajectory prediction from ego-centric views (i.e., front views captured from dashboard cameras) is an essential component in a wide range of applications from robot navigation [5, 37] to autonomous intelligent systems [18, 6]. Despite of recent research efforts, predicting pedestrian trajectories from ego-centric views remains a challenging task because of the dynamic scene changes, where pedestrian's motions in testing sequences are different from those in training data.

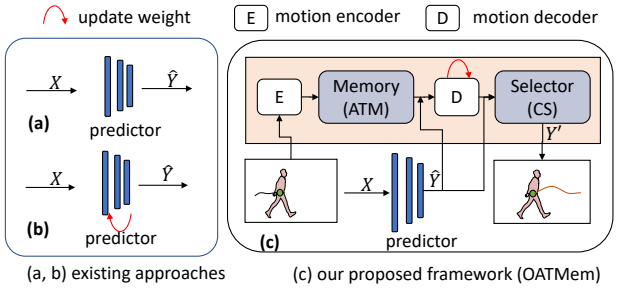

Figure 1. Comparisons of our framework (c) with the existing approaches (a, b) for trajectory prediction during test time. (a) predictor with fixed weights; (b) framework which updates predictor's weights using recent testing samples; (c) our framework with two novel components: ATM and CS for online adaptation.

Many recent models [19, 26, 31, 32, 33] have been proposed to predict pedestrians' trajectories in ego-centric views. A typical paradigm is that the predictor is first trained on large motion datasets and from that point on the model's weights will remain fixed during testing (Figure 1a). Due to dynamic changes of motions in video sequences, these predictors fail under scenarios where pedestrians' motions are unseen or rarely occur during the training phase. A possible solution is to adapt the predictor (i.e., update the network weights using recent samples) during test time (Figure 1b). With online adaptation, similar temporal dynamics between trajectory samples could be leveraged to improve the prediction accuracy [27, 10, 15, 13]. Although the temporal features were proven to effectively mitigate the issues of trajectory prediction in unseen scenarios, the limitations of the existing online adaptation methods for trajectory prediction remain. First, these methods are impractical for real-world applications as updating an

entire prediction network at every time step is extremely time and memory consuming. Second, if the motions of current testing samples are different from previous samples (used to update the model), then these frameworks worsen the prediction results of the predictor. Lastly, it requires a framework designer to know the details of each predictor to make an effective adaptation. This limits the flexibility to extend these frameworks to new predictors.

This paper introduces a novel online adaptation framework to handle the above limitations. The proposed framework (OATMem) consists of two main modules: an augmented temporal memory module (ATM), and a certainty-based selector (CS), shown in Figure 1. Motivated by the concepts of key-value memory networks [16, 15], we design the memory module to maintain the trajectories of pedestrians, extracted from short-term historical frames, in form of encoded presentations of observation-target as key-value. This structure allows us to select the most relevant target trajectory (i.e., value), whose key contains the most similar motion with the current trajectory sample's, to improve the prediction accuracy of this sample. To further enhance the adaptive ability to the temporal dynamics, we only update the weight of our motion decoder (i.e., a subpart of our framework to decode the future trajectory) during test time. This is contrast to previous works, where the weight of entire network model needs to be updated. In addition, we observe that relying only on the information (from memory) might not be effective in scenarios of newly encountered context changes (e.g., abrupt camera motions, scene changes, new pedestrians, or pedestrians with abruptly changing intentions). This is because the abrupt changes cause large motion dissimilarities between the current pedestrian's movement and those in memory. To mitigate this issue, we propose a certainty-based selector (CS) to estimate the certainty of our framework's prediction for different scenarios. Based on the certainty scores, the learned selector can select the final trajectory prediction i.e., either from our framework (when the certainty is high) or from the original prediction model (when the certainty is low). In summary, the key contributions of our work are:

- We propose a novel online adaptation framework consisting of a temporal memory network to improve the prediction accuracy of a trained predictor during testing. Our framework works with any prediction model without knowing the details of the prediction models
- We propose a new certainty-based selector to estimate the confidence of our framework and make a final prediction decision based on the certainty scores.
- We conduct an extensive evaluation of our framework with and state-of-the-art model on commonly used datasets for pedestrian trajectory prediction. We achieved state-of-the-art results without requiring any additional training data on these datasets.

2. Related Works

Human Trajectory Prediction in Dynamic Video Scenes. Most recent works [19, 26, 31, 32] utilize common network architectures such as recurrent neural networks (RNNs) [36], temporal convolution neural networks (TCNs) [39], graph neural networks [21], transformer networks [22] or other variants [9, 25] to predict human future locations. The main theme is how to efficiently incorporate different motion-related features such as camera motions, human poses, human intentions and behaviors, while some other learn human-scene interactions [11, 8, 34] and/or human-human interactions [7, 1, 17]. Forecasting human future locations in dynamic scenes remain a challenge due to abrupt camera motions, vast varieties of human motions, and dynamic scene changes. Since training data cannot be exhaustive, these prediction models cannot generalize well to all different scenarios during testing. This paper looks from a different perspective that given a trained predictor, could we improve its accuracy performance during testing without having access to details of this predictor?

Online Adaptation for Trajectory Prediction. In contrast to other research domains such as video classification [2], online adaptation to new scenes for trajectory prediction is still an under-explored topic. Much of recent research [30, 27, 10] focused on adapting a specific neural network (i.e., transfer learning) to new scenes. These works assume that the scene contexts of the testing data are known and clearly distinguished from those in training datasets. Such models are only applicable for prediction in bird's-eye views (BEV). By contrast, predicting trajectory from ego-centric views with constantly changing scenes is more challenging since the scene context changes are not easy to recognize. One of closely related research to our work is AOL [10] where it generates multiples weights of a prediction model, each representing a specific contexts for which it performs best. Although achieving promising accuracy, it results in high adaptation time and large memory consumption, which is impractical for real-time applications.

Memory networks for Trajectory Prediction. Memory networks [28, 23] can be used to explicitly store information and selectively access relevant values. Recent advances in key-value memory networks [16] have shown their effectiveness in applications such as visual question/answer [29], object tracking [38]. For trajectory prediction, MANTRA [15] relies on a key-value memory network [16] to remember the moments that it failed in the past. The information encoded in memory is then used to influence the current testing sample. Our work is clearly distinguished from these previous works [15] as the focus is on designing a framework instead of a prediction model. Be-

sides, the proposed memory structure allows to learn temporal dynamics of the scene by encoding recent trajectory samples in short-term past frames.

3. Problem Formulation

Our goal is to improve the prediction accuracy of a trajectory prediction model (i.e., a predictor) during test time. At current test time t, let $X_t = [X_t^1, \ldots, X_t^N]$ denote the observed trajectories of N pedestrians, where $X_t^i = [X_{t-T_o+1}^i, X_{t-T_o+2}^i \ldots, X_t^i]$ is the set of observed locations of pedestrian i in the last T_o frames. A predictor $p_\theta(\hat{Y}_t|X_t)$, with the network weight θ, can be used to estimate the future trajectories $\hat{Y}_t = [\hat{Y}_t^1, ..., \hat{Y}_t^N]$ of all N pedestrians in the next T_p frames, where $\hat{Y}_t^i = [\hat{Y}_{t+1}^i, \hat{Y}_{t+2}^i, ..., \hat{Y}_{t+T_p}^i]$. Similar to the existing setup [31, 19], we present each location $X_t^i = (x_t^i, y_t^i, w_t^i, h_t^i)$ as a bounding box with (x_t^i, y_t^i) is the center and w_t^i, h_t^i are the width and height of the bounding box, respectively. The predicted location is also presented as a predicted bounding box $\hat{Y}_{t+1}^i = (\hat{x}_{t+1}^i, \hat{y}_{t+1}^i, \hat{w}_{t+1}^i, \hat{h}_{t+1}^i)$. We aim to design a new framework $F_\theta(\dot{Y}_t|\hat{Y}_t, X_t)$ that produces a new prediction $\dot{Y}_t$ conditioned on the past trajectory X_t and the predicted trajectories $\hat{Y}_t$ of the predictor $p_\theta(\hat{Y}_t|X_t)$. As previous works, we assume that the past trajectories of pedestrians are fully observed, meaning that there are no noisy observations and thus the ground-truth locations can be used. Some works [14, 35] tackle the issues of noisy observations; however, this is not the focus of our work.

4. Methodology

4.1. Prediction Model

At current test time t, the predictor $p_\theta(\hat{Y}_t|X_t)$ estimates the future trajectories $\hat{Y}_t$ given the past trajectories X_t. In this work, we assume that the predictor was trained on a training dataset, and any predictor can be used. During testing, the predictor's network weight θ remains fixed, and our framework only relies on the predicted trajectory $\hat{Y}_t$ produced by the predictor. For later improvements, the predicted trajectory $\hat{Y}_t$ encoded using the prediction encoder as follows:

$$z_t^p = \text{GRU}(\text{Conv1D}(\hat{Y}_t)), \tag{1}$$

where $z_t^p \in \mathbb{R}^d$ is the encoded feature of $\hat{Y}_t$, and d is size of the hidden layer. For speed and efficiency, we use GRU (Gated Recurrent Unit [4]) followed by 1-dimensional convolutional layer to encode the temporal information of the predicted trajectory. The encoded feature z_t^p of the predictor is then used in combination with information from temporal memory for the framework's prediction.

4.2. Augmented Temporal Key-Value Memory Network (ATM)

To improve the accuracy of a predictor, we seek to exploit temporal similarities among trajectory samples. Our intuition is that there is a high probability that the future movement of a pedestrian is similar to their historical movements or other nearby pedestrians' movements as they walk in a group in the past frames. We show examples of these scenarios in Figure 3. We can see that the pedestrian, who is crossing streets, will likely remain at the same speed and direction in near future. Figure 3b shows another example of a group of pedestrians that share similar motions (e.g., passing streets); thus, future movements of the target pedestrian will likely be similar to the group's motion. Motivated by these observations, we propose an augmented temporal key-value memory network to capture these temporal similarities in short-term past frames. The augmented temporal memory network consist of M rows. We represent each row $m \in \{1, .., M\}$ as a pair of key and value $\{k_m, v_m\}$, where k_m and v_m are the encoded features of the observation $X_{t'}^j$ and the target $Y_{t'}^j$ of the past trajectory samples $\{X_{t'}^j, \hat{Y}_{t'}^j\}$, where $j \in [1, \ldots, N_{t'}]$ is the pedestrian id and $N_{t'}$ is the number of pedestrian at time t'. We collect M past samples, each corresponds to a memory row, from the short-term historical frames $t' \in [0, \ldots, t-\delta]$, where δ is the number of frames to ensure there is no overlap between the current testing samples $\{X_t^i, Y_t^i\}$ and those encoded in the memory. Note that we don't exclude the probability that past pedestrian's identity could be the same with the current one (i.e., $j = i$) as pedestrians could have long-term trajectories.

The encoding process is executed using the motion encoder, which maps the observed $X_{t'}^j$ and target trajectory $Y_{t'}^j$ into different latent feature spaces as follows:

$$k_m = \text{GRU}(\text{Conv1D}(X_{t'}^j)) \in \mathbb{R}^{d_v}, \tag{2}$$

$$v_m = \text{GRU}(\text{Conv1D}(Y_{t'}^j)) \in \mathbb{R}^{d_k}, \tag{3}$$

where d_v, and d_k are the size of hidden representation for key k_m and value v_m, respectively. Next, we discuss the details of read and write memory operations.

Read/Write operation. As mentioned earlier, the goal of our memory network is to find the past trajectory $\{X_{t'}, Y_{t'}\}$ that is most similar to the current testing sample $\{X_t, Y_t\}$ so that its encoded representation (i.e., retrieved from memory) could be used to improve the predictor's prediction $\hat{Y}_t$. This is achieved by calculating the similarity scores between the encoded observed trajectory X_t at the current time step t and all keys in the memory as:

$$s_m = \frac{e_t k_m}{||e_t|| ||k_m||}, \forall m \in \{0, ..., M-1\}, \forall s_m \in [0, 1], \tag{4}$$

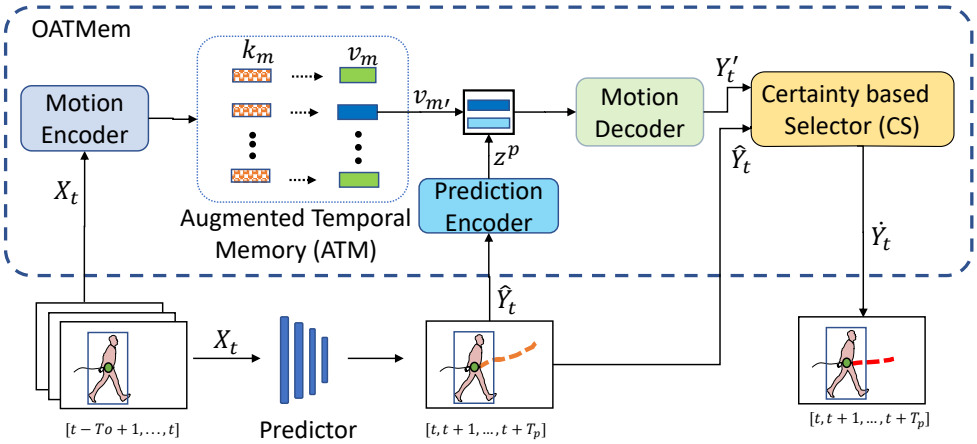

Figure 2. The overview of our proposed framework consisting of three modules: a predictor (Section 4.1), an augmented temporal memory (ATM) (Section 4.2), and a certainty-based selector (CS) (Section 4.4).

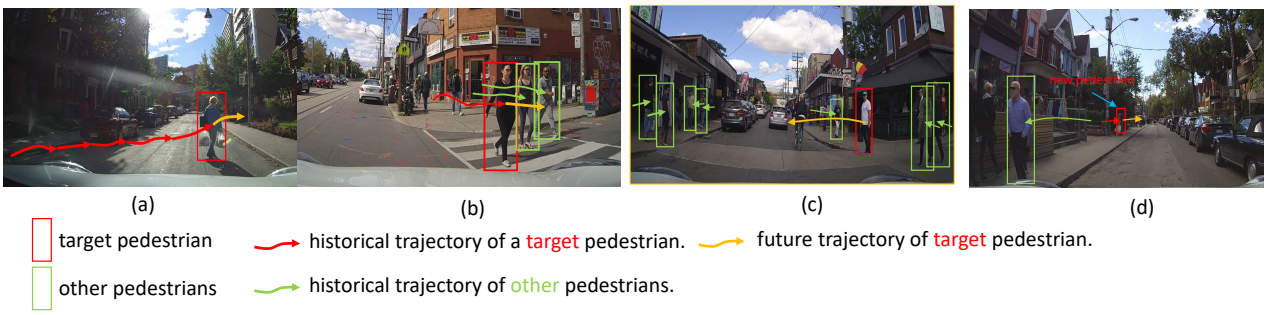

Figure 3. Scenarios where there are high motion similarities (a, b) and dissimilarities (c,d) between target pedestrian and other pedestrians.

where e_t is the encoded representation of X_t, obtained using Equation 2. The higher s_m score, the higher temporal similarity between the current testing sample $\{X_t, Y_t\}$ and past encoded trajectory sample $\{X_{t'}, Y_{t'}\}$. Based on these similarity scores, the selected memory value $v_{m'}$ can be retrieved as: $v_{m'} \leftarrow M(m'), m' = \text{argmax}_m(s_m)$, where m' is the selected row. The value $v_{m'}$ is then used to improve the predictor's prediction in the trajectory decoding stage, described in Section 4.3. At each time step during testing, the memory is augmented (i.e., written or updated) with new testing samples. This allows the memory to cope with the dynamic changes of video scenes. However, we maintain a fixed-size M-row memory as we do not only aim for accuracy but also for speed and memory efficacy. To achieve this, we use the first-in-first-out (FIFO) strategy to discard the trajectory samples from oldest frame and augment new samples from the most recent one. This is reasonable as these samples contain information that is not useful for improving the prediction with the current one, considering the fast temporal dynamics of video sequences. In scenarios that the number of newly augmented samples are large (i.e., in crowded scenes), writing all these samples would increase the memory size. In this situation, we randomly select these samples to keep M fixed. We analyze the effect of memory sizes in Section 5.2.

4.3. Motion Decoder

Given the representations of both predicted trajectories of the predictor z_t^p and memory $v_{m'}$, the motion decoder decodes the future trajectories as:

$$Y'_t = \text{fc}(\text{GRU}([z_t^p, v_{m'}])), \tag{5}$$

where fc is a fully connected layer. We concatenate the encoded representations of predicted trajectory of predictor and selected value from memory to leverage the advantages of both. The native predictor may rely on different features such as goals, pedestrian intentions, etc., to generate predictions while our memory provides useful temporal information for trajectory prediction. To further improve the predictive performance of our framework, the motion decoder's weights can be updated using the most recent trajectory sample. We show the impacts of these operations in our ablation study (Section 5).

4.4. Certainty-based Selector (CS)

Relying only on temporal information from memory is not sufficient to cope with different scenarios in dynamic scenes. Dynamic scenes might inherently contain many scenarios that the movements of current pedestrians do not correlate with those in the past, as shown in Figure 3c and Figure 3d. We can see that the scenario in Figure 3c consists of various motions from different pedestrians, while the scenario in Figure 3d shows an example of a new pedestrian appearing far in distance and this pedestrian's movement is very different from the one closer to the camera. In these scenarios, we observe that the prediction results from memory could worsen the predictors' results. To mitigate this issue, we propose a novel certainty-based selector, which learns to select the prediction from the predictor $\hat{Y}_t$ or from our framework Y'_t to become the final prediction $\dot{Y}_t$. The final prediction $\dot{Y}_t$ can be derived as:

$$\dot{Y}_t = (1 - \mathcal{S}_\phi(s_t|Y'_t, \hat{Y}_t))\hat{Y}_t + \mathcal{S}_\phi(s_t|Y'_t, \hat{Y}_t)Y'_t, \quad (6)$$

where $\mathcal{S}_\phi(s_t|Y'_t, \hat{Y}_t)$ is the proposed certainty-based selection function that estimates the certainty of our framework based on the prediction of our framework in comparison with the native predictor's. Specifically, the certainty score is estimated as:

$$c_t = \varsigma(\text{MLP}(\text{GRU}(\text{Conv1D}([\hat{Y}_t, Y'_t])))), c_t \in [0, 1], \quad (7)$$

$$s_t = \mathbb{1}(c_t > \delta_s), s_t \in \{0, 1\}, \quad (8)$$

where MLP is a multilayer-perceptron, $\varsigma(\cdot)$ is a sigmoid function. The high certainty score (i.e., $c_t \to 1$) indicates that the framework produces more accurate prediction than predictor's prediction. $\mathbb{1}(c_t > \delta_s)$ is an indicator function used to enforce 'hard' selection on either $\hat{Y}_t$ or Y'_t. δ_s is a pre-defined threshold, set to 0.5 in our experiments.

We train the certainty-based selector to imitate the behaviors of the indicator function $\mathbb{1}(Y'_t, \hat{Y}_t) = \mathbb{1}(||Y'_t - Y_t||_2^2 < ||\hat{Y}_t - Y_t||_2^2)$. Specifically, $\mathbb{1}(Y'_t, \hat{Y}_t) = 1$ indicates if the prediction from our framework Y'_t is more accurate (i.e., closer to ground truth trajectory Y_t measured by Frobenius norm $||\cdot||_2^2$) than the prediction $\hat{Y}_t$ from the predictor; otherwise, $\mathbb{1}(Y'_t, \hat{Y}_t) = 0$. Thus, the selector is trained separately using the binary cross entropy loss [20]. We present the details of training/testing procedures and loss functions in the supplementary materials.

5. Experiments

We evaluate our framework using JAAD [12] and PIE [19] datasets for pedestrian trajectory prediction from ego-centric views. We describe the details of these datasets and implementation details in the supplementary materials.

Evaluation metrics. We evaluate using the commonly used evaluation metrics [19, 31]: ADE, average displacement error of predicted bounding box and the ground truth; CADE, average displacement error of the center of bounding box; CFDE, final displacement error of bounding box's center at final location. All metrics are measured in pixels.

Comparison Models. We evaluate our framework in combination of three recent prediction models: BiTrap [31]: predict trajectory conditioned with goal conditions. PIEtraj [19]: an RNN based encoder-decoder model with temporal attention. PIEfull [19]: a variant of PIEtraj by incorporating human intention and vehicles speeds for trajectory prediction. We reported our results with other existing methods: Linear [19], LSTM [19], B-LSTM [3], FOL-X [33], PIEtraj [19], PIEfull [19], BiTrap [31].

5.1. Quantitative results

We present our quantitative results in Table 1. We can see that our framework in combination with other native predictors (PIEtraj [19], PIEfull [19], and BiTrap [31]) achieves better results (i.e., lower prediction errors) compared to the native predictors alone. This indicates that the temporal information from memory and the selector plays a vital role on improving prediction accuracy. In addition, our framework in combination with BiTrap gains the best prediction results in all metrics.

Ablation Study. We perform the ablation studies (Table 2) to investigate the impacts of each framework variant. These variants include: OATMem: our full framework; OATMem (w/o concat): our framework but without concatenating the representations of predictor's prediction and retrieved target trajectory from memory in Equation 5; OATMem (w/o selector): our framework without certainty-based selector; OATMem (w/o online update): our framework without updating the motion decoder (Section 4.3). We observe that dropping one of these components increases prediction errors. Among these components, it is notable that the selector plays the most important role as OATMem (w/o selector) results in the highest prediction error in most metrics. However, our framework without selector still achieves better results than the native predictor alone. This means the memory is capable of encoding temporal information that is useful for improving trajectory prediction.

5.2. Analysis

In this section, we present additional analyses to understand the performance of our framework.

Correlations between prediction errors and selector's accuracy. We analyze the correlation between the prediction errors and the performance of the selector, as shown in Figure 4a. We train the selector with 200 epochs on a random subset of training data splits and report the corresponding FDE of our framework + BiTrap at each epoch. We can see that the increase in selector's accuracy correlates with the reduction of prediction errors. This is reasonable because the more accurate selection of predictions leads to

Table 1. Quantitative results on PIE and JAAD datasets.

Methods	JAAD			PIE		
	ADE (0.5/1.0/1.5s)	CADE (1.5s)	CFDE (1.5s)	ADE (0.5/1.0/1.5s)	CADE (1.5s)	CFDE (1.5s)
Linear [19]	233/857/2303	1565	6111	123/477/1365	950	3983
LSTM [19]	289/569/1558	1473	5766	172/330/911	837	3352
B-LSTM [3]	159/539/1535	1447	5615	101/296/855	811	3159
FOL-X [33]	147/484/1374	1290	4924	47/183/584	546	2303
PIEtraj [3]	110/339/1248	1183	4780	58/200/636	596	2477
PIEfull [3]	-	-	-	42/154/559	520	2162
BiTrap [31]	93/378/1206	1105	4565	41/161/511	481	1949
OATMem (ours)						
+ PIEtraj [3]	105/306/1089	1107	4385	52/163/497	456	1944
+ PIEfull [3]	-	-	-	41/150/502	433	1819
+ BiTrap [31]	**83/294/926**	**876**	**3690**	**40/157/457**	**369**	**1726**

Table 2. Ablation Study. We investigate the impact of each proposed component of our framework.

Method	Framework variants	JAAD			PIE		
		ADE (0.5/1.0/1.5s)	CADE (1.5s)	CFDE (1.5s)	ADE (0.5/1.0/1.5s)	CADE (1.5s)	CFDE (1.5s)
BiTrap [31]		93/378/1206	1105	4565	41/161/511	481	1949
	OATMem	**83/294/926**	**876**	**3690**	**40/157/457**	**369**	**1726**
	OATMem (w/o concat)	92/329/1037	891	4132	57/171/495	405	1810
	OATMem (w/o selector)	87/325/1018	969	4017	54/168/506	417	1918
	OATMem (w/o online update)	87/309/976	923	3889	77/195/453	425	1801

more accurate trajectory prediction as we discussed earlier.

Correlations between prediction errors and motion variances. We seek to understand the performance of predictor and our framework on different trajectories categorized by high, medium, and low motion variances (Figure 5). The motion variance of a trajectory sample is measured using $L2$-norm distance between the final location (i.e., center of bounding box) and current location of each trajectory sample. Then, we classify them into three main categories: high variance: top 20%, low variance: lowest 20%, and medium: in between, for further analysis. Figure 5a shows the t-SNE visualization [24] of trajectory samples categorized by their prediction errors (CFDE) and their motion variance (Figure 5b). It is reasonable that for those trajectories that result in large motions (i.e., moving with high speeds or abrupt motions), the predictors will likely suffer (i.e., high prediction errors). Interestingly, we can see that our framework could improve the prediction errors on these samples significantly (Figure 5c). This explains that our memory module provides useful temporal information that could be used for improving the predictors' accuracy. Lastly, Figure 4d shows that we can reduce the prediction error on those samples of high errors (i.e., corresponds to large motions) up to 27%.

Impact of memory size. The impact of memory size (i.e., number of rows) on trajectory prediction errors of two framework variants (with and without certainty-based selector) is illustrated in Figure 4b. We can see that with enough samples (i.e., 8) both variants achieve the best results. However, it is interesting to observe that the larger memory size does not lead to better prediction performance. This indicates that with a memory size of 8 we capture most of the temporal variations. Thus, encoding more samples increases the dissimilarities between samples from farther past frames, leading to the increased prediction errors.

Impact of the number of iterations for updating the decoder. As discussed in Section 4.3, adapting the decoder increases the adaptive ability of our framework. To gain in-depth understanding, we analyze the performance of two variants with a different number of iterations for updating the decoder using the most recent testing sample, shown in Figure 4c. We observe that our variant without CS converges fast when the number of iterations increases and saturates at 3 iterations. This indicates that adapting decoders significantly helps improve the adaptive ability. Additionally, we find that the variant with CS can tolerate the prediction errors when the number of iterations is small, even though its predictive performance also converges with increasing number of iterations.

Time and Memory Complexities. Processing using extracted data from images is computationally expensive. One of our goals is to reduce the time and memory complexity by designing a framework with minimal memory and time consumption. Our comparisons with recent framework

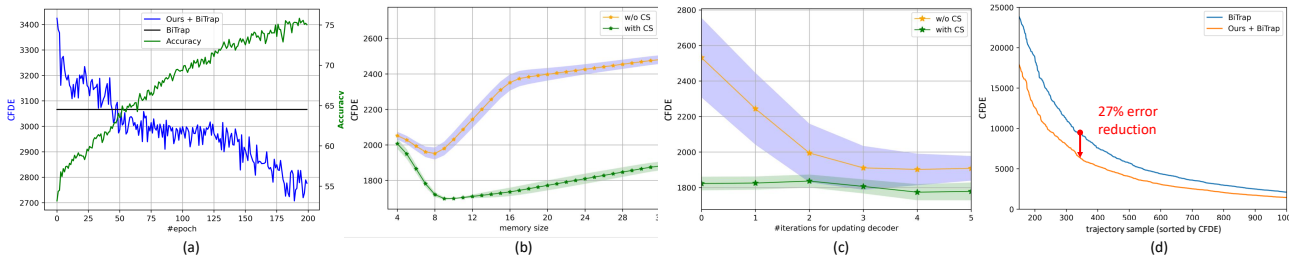

Figure 4. (a) Correlation between selector's accuracy and trajectory prediction errors; the impact of memory size (b) and (c) the impact of number of iterations for updating decoder on trajectory prediction; (d) the error reduction on large-motion samples. Shaded areas represent variances.

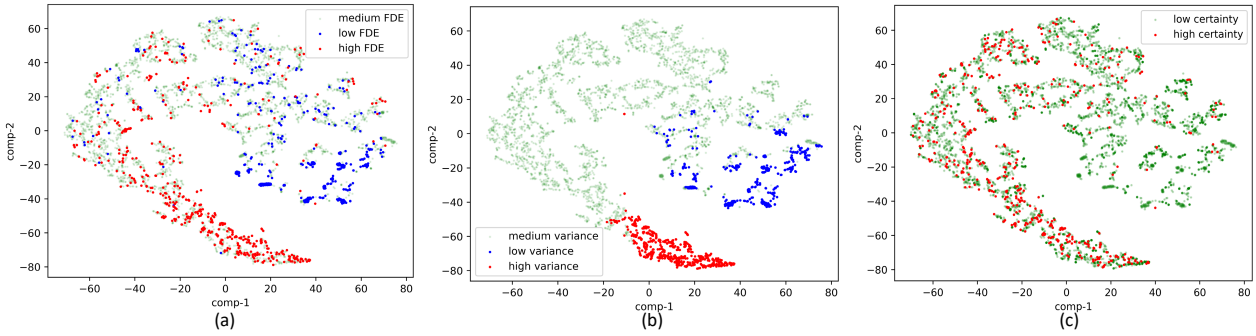

Figure 5. 2D t-SNE visualization [24] with two components (comp-1, comp-2) (i.e., dimensions) of samples from PIE dataset.

BAOL [10] is shown in Table 3. Because our framework does not require any access to the predictor, the number of trainable parameters of our framework is much less than BAOL. Additionally, since we only adapt the decoder during test time, our adaptation time is significantly less.

Correlations between prediction behaviors and ego-vehicle's movements. The movements of ego-vehicles such as speed, and turn directions, strongly impact the dynamics of the scenes; and thus, affect the predictive performance, as shown in Figure 6a. We can see that when the vehicle moves in stable states such as going forward with the same or gradually changing speed, the memory can carry useful information to improve the predictor's results. This can be observed in scenarios 1 and 3, where our variant OATmem+BiTrap (without CS) outperforms BiTrap. However, this is not the case when ego-vehicles abruptly make a turn (scenario 2) or accelerate their speeds (scenario 4). By learning the comparative prediction behaviors of both memory and predictor, our framework with a certainty-based selector is able to recognize these scenarios. Thus, our framework with a certainty-based selector exceeds the prediction performance of BiTrap in these scenarios.

Adaptation to new scenes. We investigate a common scenario of testing a predictor in new scenes, where new pedestrians with different movement patterns appear. In Figure 6b, OATmem+BiTrap (without CS) performs worse than the native predictor at the beginning of the new video sequence as the movements of pedestrians in this new scene are dissimilar from those initialized from training dataset. Interestingly, this variant started to improve the prediction accuracy (i.e., lowering prediction error) and outperformed the predictor at frame 60, where sufficient similar motions are encoded in memory. Finally, our final framework OATmem+BiTrap (with CS) takes advantages of both predictor and memory; thus, it performs best in most frames.

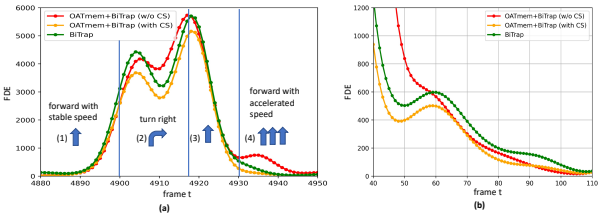

Figure 6. (a) Prediction results (averaged FDE of all samples in each frame) of continuous video sequence from PIE dataset, which consists of different ego-vehicle's movements.(b) Prediction results when adapting to new video sequence.

6. Qualitative results

We present our qualitative results in Figure 7. For each scenario, we visualize the prediction results (right figure) and the trajectory encoded in the memory (left figure). In

Table 3. Comparisons with base adaptive online learning (BAOL) [10] on JAAD [12] dataset.

Methods	CADE/CFDE (pixels)	Inference time (milliseconds)	Adaptation time (milliseconds)	Trainable Parameters (millions)
BAOL [10] + BiTrap [31]	1014/3824	20.92	415.51	1.53
OATMem + BiTrap	876/3690	6.2	145.41	0.12

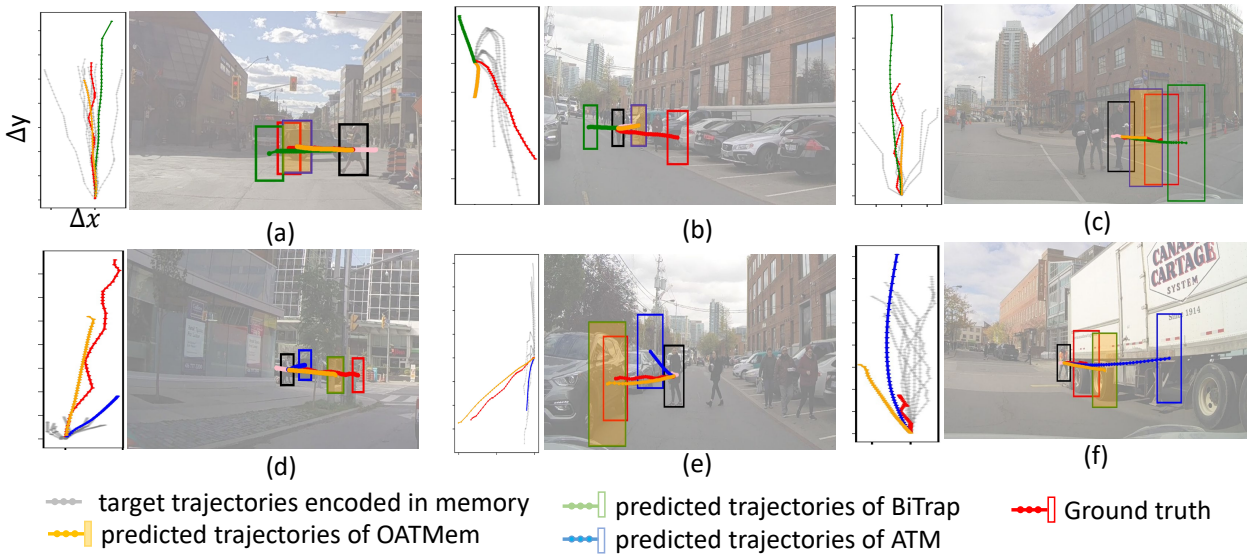

Figure 7. Qualitative results of our framework. In each scenario, trajectories encoded in memory are visualized on left, the predictions on image are shown on right.

the first row, the selector selects the prediction from our framework in scenarios where the movement of the target pedestrian is highly similar to those stored in memory. These scenarios include a pedestrian crossing the street (Figure 7a) or a pedestrian walking along the street (Figure 7b), and a pedestrian walks in group (Figure 7c). On the other hand, the second row shows scenarios where there are abrupt or various motions that are not helpful for improving trajectory prediction. For example, Figure 7d shows an example of a pedestrian who changes action from standing still to crossing street. As we can see, the memory stores all short-term trajectories, which represents the slow motions of this pedestrian. However, this information from memory is not relevant to the future motion of the target pedestrian. Another example of changing intention is shown in Figure 7f. In this scenario, the pedestrian changes motion from fast to slow. Lastly, Figure 7e shows examples of multiple motions of different pedestrians, which are not helpful for predicting future trajectory. However, in these scenarios, the selector is still capable of selecting better predictions, which are produced by the predictor.

7. Conclusions

We presented a novel framework for online adapting and improving a given prediction model during test time. The key components of our framework include an augmented temporal key-value memory (ATM) module that encodes temporal information from past trajectories. We also propose a certainty-based selector (CS) that infers certainty score based on the predictor's performance. In future works, the framework could be extended to improve prediction models in other applications such as multi-agent trajectory prediction and can be incorporated with other predictors for improving the prediction accuracy of these predictors.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016.

[2] Jawadul H Bappy, Sujoy Paul, and Amit K Roy-Chowdhury. Online adaptation for joint scene and object classification. In

European Conference on Computer Vision, pages 227–243. Springer, 2016.

[3] Apratim Bhattacharyya, Mario Fritz, and Bernt Schiele. Long-term on-board prediction of people in traffic scenes under uncertainty. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4194–4202, 2018.

[4] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[5] Guilherme N DeSouza and Avinash C Kak. Vision for mobile robot navigation: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 24(2):237–267, 2002.

[6] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

[7] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2255–2264, 2018.

[8] Sirin Haddad and Siew-Kei Lam. Self-growing spatial graph network for context-aware pedestrian trajectory prediction. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 1029–1033. IEEE, 2021.

[9] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[10] Manh Huynh and Gita Alaghband. Aol: Adaptive online learning for human trajectory prediction in dynamic video scenes. *arXiv preprint arXiv:2002.06666*, 2020.

[11] Arash Kalatian and Bilal Farooq. A context-aware pedestrian trajectory prediction framework for automated vehicles. *Transportation research part C: emerging technologies*, 134:103453, 2022.

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

[13] Maosen Li, Siheng Chen, Yanning Shen, Genjia Liu, Ivor W Tsang, and Ya Zhang. Online multi-agent forecasting with interpretable collaborative graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[14] Karttikeya Mangalam, Ehsan Adeli, Kuan-Hui Lee, Adrien Gaidon, and Juan Carlos Niebles. Disentangling human dynamics for pedestrian locomotion forecasting with noisy supervision. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2784–2793, 2020.

[15] Francesco Marchetti, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. Mantra: Memory augmented networks for multiple trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7143–7152, 2020.

[16] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126*, 2016.

[17] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14424–14432, 2020.

[18] Ryosuke Okuda, Yuki Kajiwara, and Kazuaki Terashima. A survey of technical trend of adas and autonomous driving. In *Technical Papers of 2014 International Symposium on VLSI Design, Automation and Test*, pages 1–4. IEEE, 2014.

[19] 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.

[20] 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.

[21] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[22] Ze Sui, Yue Zhou, Xu Zhao, Ao Chen, and Yiyang Ni. Joint intention and trajectory prediction based on transformer. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7082–7088. IEEE, 2021.

[23] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. *Advances in neural information processing systems*, 28, 2015.

[24] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[25] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[26] Chuhua Wang, Yuchen Wang, Mingze Xu, and David J Crandall. Stepwise goal-driven networks for trajectory prediction. *IEEE Robotics and Automation Letters*, 7(2):2716–2723, 2022.

[27] Letian Wang, Yeping Hu, Liting Sun, Wei Zhan, Masayoshi Tomizuka, and Changliu Liu. Transferable and adaptable driving behavior prediction. *arXiv preprint arXiv:2202.05140*, 2022.

[28] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

[29] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In *International conference on machine learning*, pages 2397–2406. PMLR, 2016.

[30] Yi Xu, Lichen Wang, Yizhou Wang, and Yun Fu. Adaptive trajectory prediction via transferable gnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6520–6531, 2022.

[31] 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.

[32] Yu Yao, Ella Atkins, Matthew Johnson Roberson, Ram Vasudevan, and Xiaoxiao Du. Coupling intent and action for pedestrian crossing behavior prediction. *arXiv preprint arXiv:2105.04133*, 2021.

[33] Yu Yao, Mingze Xu, Chiho Choi, David J Crandall, Ella M Atkins, and Behzad Dariush. Egocentric vision-based future vehicle localization for intelligent driving assistance systems. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9711–9717. IEEE, 2019.

[34] Jian Yu, Meng Zhou, Xin Wang, Guoliang Pu, Chengqi Cheng, and Bo Chen. A dynamic and static context-aware attention network for trajectory prediction. *ISPRS International Journal of Geo-Information*, 10(5):336, 2021.

[35] Rui Yu and Zihan Zhou. Towards robust human trajectory prediction in raw videos. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8059–8066. IEEE, 2021.

[36] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[37] Alexander Zelinsky. A mobile robot navigation exploration algorithm. *IEEE Transactions of Robotics and Automation*, 8(6):707–717, 1992.

[38] Zikun Zhou, Xin Li, Tianzhu Zhang, Hongpeng Wang, and Zhenyu He. Object tracking via spatial-temporal memory network. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(5):2976–2989, 2021.

[39] Ali Ziat, Edouard Delasalles, Ludovic Denoyer, and Patrick Gallinari. Spatio-temporal neural networks for space-time series forecasting and relations discovery. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 705–714. IEEE, 2017.