Continual Trajectory Prediction with Uncertainty-Aware Generative Memory Replay

Xiushi Feng[1], Shuncheng Liu[1], Haitian Chen[1], Kai Zheng[1,*,†]
[1]University of Electronic Science and Technology of China, China
{xiushifeng,liushuncheng,haitianchen}@std.uestc.edu.cn, zhengkai@uestc.edu.cn

***Abstract*—A reliable autonomous driving system should take safe and efficient actions in constantly changing traffic. This requires the trajectory prediction model to continuously learn from incoming data and adapt to new scenarios. In the context of rapidly growing data volume, existing trajectory prediction models must retrain on all datasets to avoid forgetting previously learned knowledge when facing additional data from new environments. In contrast, the paradigm of continual learning solely necessitates training on new data, saving a significant amount of training overhead. Therefore, it is crucial to equip the trajectory prediction model with the ability of continual learning. In this paper, inspired by rehearsal and pseudo-rehearsal methods in continual learning, we propose a continual trajectory prediction framework with uncertainty-aware generative memory replay, CTP-UGR. Our framework effectively avoids excessive memory space requirements while generating trajectory data that is authentic, representative and discriminative for continual learning. Extensive experiments on two real-world datasets demonstrate our proposed CTP-UGR significantly outperforms other baselines in terms of both accuracy and catastrophic forgetting. Besides, our framework can be combined with other state-of-the-art trajectory prediction models to achieve better performance.**

***Index Terms*—Continual Trajectory Prediction, Uncertainty, Generative Memory Replay**

I. Introduction

Trajectory prediction(TP) is an essential technique in autonomous driving. For a reliable TP system, it should be able to operate safely and continuously in constantly changing new traffic scenarios. Therefore, the ability of continual learning for trajectory prediction systems becomes increasingly crucial in the future. However, currently, there are few works researching continual trajectory prediction (CTP) methods, and there is also no detailed research on the performance of CTP settings.

Current deep trajectory prediction models are trained offline on a complete dataset that includes all scenarios to achieve good prediction performance, referred to as *offline learning*. In this learning approach, if new data arrives, the model can only undergo complete retraining on all datasets. A more efficient strategy is to train TP models on a continuous data stream and only needs to update the existing model on the new dataset. This strategy is known as *continual learning*(CL). However, naive CL methods are likely to suffer from "catastrophic forgetting" [1]. This is manifested by a significant drop in performance on the previously learned dataset when training on a new dataset. We consider that in recent literature [2]–[4], replay-based methods have shown good performance in preserving old data information and mitigating catastrophic forgetting. One potential replay method is called *rehearsal* [5], which involves selecting appropriate samples for storage and periodically revisiting them during the training of new data streams. Another replay technique is called *pseudo-rehearsal* [6], which generates pseudo-data that simulates past experiences for replay. However, both these methods have inherent limitations. Rehearsal methods require explicit storage of old experiences, leading to a significant working memory requirement [7]. Pseudo-rehearsal methods heavily depend on the quality of the pseudo-data generated by the generative model, which leads to poor performance in TP tasks that involve intricate spatio-temporal data structures [8].

In our work, we integrate the advantages of rehearsal and pseudo-rehearsal, following the general framework introduced by [7], and propose an Uncertainty-Aware Generative Replay Model for continual learning trajectory prediction (CTP-UGR). The model comprises a base model and an uncertainty-aware dual-memory system. Initially, the model obtains representative and distinctive scene data through uncertainty-aware episodic memory. These data are subsequently utilized as the initial condition to generate pseudo-data using an attention-based Conditional Variational Autoencoder. The generated pseudo-data is replayed in new tasks to mitigate catastrophic forgetting. Our contributions can be summarized as follows:

1). To the best of our knowledge, our research is one of the few studies that utilize continual learning for vehicle trajectory prediction tasks, providing the potential to enhance the robustness and reliability of autonomous driving trajectory prediction systems.

2). We propose a novel framework called Continual Trajectory Prediction with Uncertainty-aware Generative memory Replay (CTP-UGR). This framework effectively addresses extensive working memory requirements in rehearsal methods and the challenge of pseudo-rehearsal generating unrealistic and non-representative data with low discriminative capability.

3). We validated our approach on two real-world datasets through effectiveness analysis, ablation study and robustness analysis. We demonstrated the effectiveness of CTP-UGR and its components for CL, and it can be integrated with other state-of-the-art TP models to achieve better performance.

*Corresponding author: Kai Zheng
†Kai Zheng is with Yangtze Delta Region Institute (Quzhou), School of Computer Science and Engineering, and Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China.

II. RELATED WORK

A. Trajectory Prediction

In recent years, deep learning-based TP methods have gained popularity due to their effectiveness. These methods largely utilize sequence and graph networks to extract scene and trajectory features [9], [10]. However, most of these methods are trained offline using a complete dataset, but real-world applications require models that can adapt to various traffic scenes. In this paper, we propose a novel CTP framework, which can be incorporated as a component of most TP models to achieve CL effects.

B. Continual learning

Currently, most of the research on continual learning is mainly focused on image classification tasks, and can be roughly divided into (1) Replay-based methods [2]–[4]. (2) Regularization-based methods [11]. (3) Architecture-based methods [12]. However, there are limited investigations on CTP problems. Recently, Wu et al. [13] proposed a continual pedestrian trajectory prediction method using scene-level generative replay. Ma et al. [8] proposed a continual learning framework for multi-agent trajectory prediction using a conditional generative replay model with road scenes as conditional inputs for generating trajectory data. However, their emphasis was only on the authenticity of the generated trajectory data, without considering the effectiveness of the generated trajectory data in overcoming catastrophic forgetting. Knoedler et al. [14] combined the rehearsal and regularization-based methods for continual pedestrian trajectory learning. However, it leads to significant working memory requirements. In contrast, our proposed method addresses the limitations of rehearsal and pseudo-rehearsal methods and leverages their benefits. Not only avoids the need for large memory requirements but also generates authentic, representative and discriminative trajectory data that effectively alleviate catastrophic forgetting.

III. PROBLEM FORMULATION

Our goal is to train a trajectory predictor through the application of continual learning. Initially, we introduce the domain problem(i.e., TP). Then we outline the relevant formula for continual learning problems in our application.

A. Trajectory Prediction

The trajectory prediction problem can be formally defined as predicting the future state of traffic participants using their past states in a given scenario. In this study, we assume that observation of N agents in each scenario is represented as $o = \{o^1, o^2, ..., o^N\}$, where $o^i_{past} = (p^t_i, c^i)$ represents the historical state observed by agent i. The coordinate of the historical trajectory observed by agent i at observation time T_{obs} is given by $p^i_t = (x^t_i, y^t_i)$, where $t = 1, ..., T_{obs}$. c^i represents the path point associated with the trajectory of agent i, which is extracted from the provided HD map. Our purpose is to leverage a data-driven prediction model P_θ to predict the future position $y^i_{fut} = (p^t_i), t = T_{obs+1}, ..., T_{pred}$, which is to identify the maximum posterior probability $\underset{\theta}{\operatorname{argmax}} \left(y^i \mid o^i\right)$.

B. Continual Learning

In the problem setting of continual learning, task data is trained sequentially, we assume that we cannot store all previous data. Each experiment is conducted on multiple datasets collected at different times and different locations. We denote D_i as the i-th dataset we received. The performance of the model is evaluated based on a given sequence of datasets $\{D_1, ..., D_M\}$, where M represents the number of datasets. The ultimate objective of the prediction model is to accurately forecast all M tasks after training on all task data.

IV. METHODOLOGY

Fig.1 shows an overview of CTP-UGR. It consists a base model(see Fig.1(d)) responsible for trajectory prediction and an uncertainty-aware dual memory module(see Fig.1(e)). Next, we provide a description of the training phase(see Fig.1(b)) and the replay memory phase(see Fig.1(c)). Our contribution mainly focuses on the overall design of the framework and the uncertainty-aware dual memory module.

A. Base Model

The base model adopts an encoder-decoder architecture for TP tasks(see Fig.1(d)). As our focus is on solving the issue of catastrophic forgetting in continual learning, and not on improving the TP model itself, we have selected a standard LSTM-based TP model as the base model. Furthermore, more advanced TP models can be employed to substitute this architecture. In the subsequent experimental section V-F, we tested deploying our approach on several state-of-the-art TP models to further evaluate the robustness of our method.

B. Uncertainty-Aware Dual-Memory Module

As mentioned in section I, rehearsal and pseudo-rehearsal have inherent limitations and perform poorly when dealing with complex spatio-temporal data structures. Therefore, we proposed an uncertainty-aware dual-memory module(see Fig.1(e)) including Uncertainty-Aware Episodic Memory and Conditional Generative Replay Memory.

1) ***Uncertainty-Aware Episodic Memory*** *:* In the context of CL tasks, we propose that the historical task data stored in the memory should be difficult to learn or more likely to be forgotten, and should also have distinctiveness from other task data. Currently, many studies [4], [15] have demonstrated that uncertainty sampling has superior performance over other updating memory methods. The uncertainty quantifies the lack of confidence the model has in its prediction, and estimates the relative position of each sample in the feature space [5]. Higher uncertainty indicates that samples are closer to the boundaries of the distribution, making them more distinctive. On the other hand, samples with lower uncertainty are closer to the center of the distribution, making them more representative. To guarantee that the memory stores representative samples from past task data as well as distinctive samples from other tasks, we store both uncertain and certain samples.

We use Bayesian Networks (BNN) to calculate sample uncertainty. The Monte Carlo Dropout method (MC Dropout)

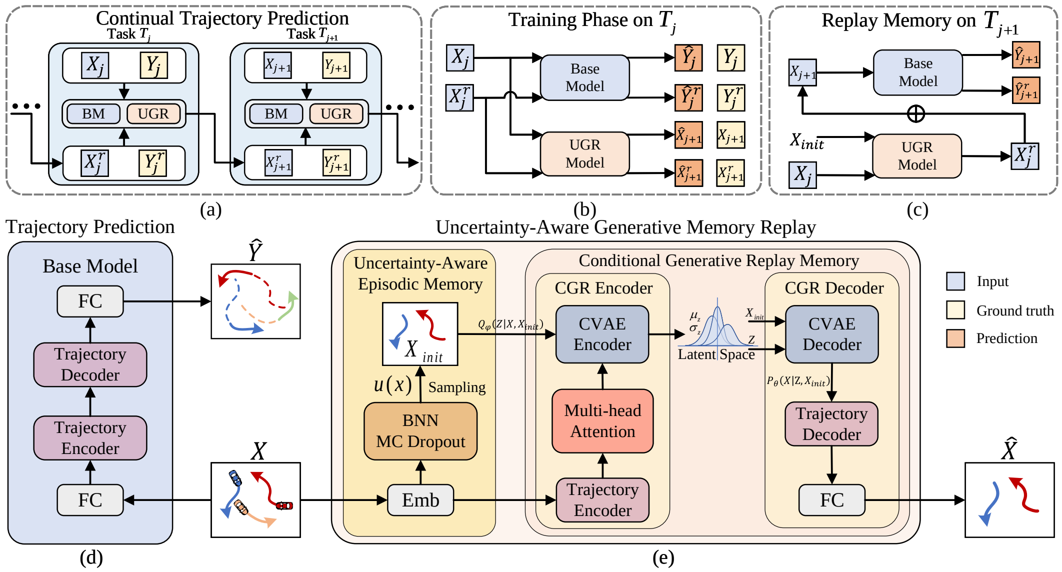


Fig. 1. The overview of our proposed CTP-UGR framework. (a) Continual Learning in CTP-UGR. The model undergoes continuous training according to the order of tasks. (b) The schematics of the training phase. For each task, the model is trained using both the current task data and the replay data. (c) The schematics of the replay memory phase. The pseudo-data generated from the previous task is incorporated into the data of the subsequent task to train together. (d) The structure of the Base Model(BM). (e) The structure of Uncertainty-aware Generative memory Replay model(UGR).

[16] is adopted to transform our base model into a Bayesian Network(see Fig.1(e)). Specifically, Dropout layers are added to the model and without turning off during forward propagation when testing. As a result, the weights $\widehat{w}_i$ of the model can be regarded as following a Bernoulli distribution$\widehat{w}_i \sim B(1,p)$, which enables variational inference. By applying Monte Carlo sampling, T forward propagations are performed for each sample x_i in the current task data, resulting in T predicted values $f^{\hat{w}_i}(x_i)$. The variance of these T predicted values serves as a measure of uncertainty, with a larger variance indicative of heightened uncertainty. The formula for calculation is as follows:

$$u(x_i) = \frac{1}{T}\sum_{i=1}^{T}\left(f^{\hat{w}_i}(x_i) - E(y)\right)^2 \tag{1}$$

In the equation, $u(x_i)$ represents the uncertainty of sample x, while $E(y)$ represents the mean of the outputs. We sort all these samples based on their uncertainty. Higher values of $u(x_i)$ indicate a region where the model exhibits lower confidence, i.e., fragile samples, while lower values of $u(x_i)$ indicate a region where the model is highly confident, i.e., robust samples. We set the data for memory slots as k_c, and select the first $k_c/2$ samples and the last $k_c/2$ samples from the sorted list to fill the memory. This brings diversity to the episodic memory based on uncertainty awareness. We define these samples in the episodic memory, denoted ad X_{init}, where $X_{init} = \left\{x_1, \ldots, x_{k_c/2}, x_{k_c/2+1}, \ldots, x_{k_c}\right\}$.

*2) **Conditional Generative Replay Memory**:* To generate both realistic, representative and discriminative pseudo data, we designed a conditional generative-replay model (CGR). The model is based on a conditional variational autoencoder (CVAE), and the objective of CGR is to solve $P(X \mid X_{init})$, where X represents a complete trajectory prediction scenario with trajectories of all agents in the scene. Through the utilization of conditional variational inference, it becomes feasible to generate authentic trajectories based on the provided initial episodic memory X_{init}. The model adopts an encoder-decoder architecture, and the overall framework is illustrated(see Fig.1(e)). The modules included in the CGR model are as follows:

CGR Encoder: To better capture the interaction between vehicles and generate more realistic trajectory data, we introduced a multi-head attention mechanism [17] in the encoder part. The CGR encoder is composed of three constituents: a trajectory encoder, a multi-head attention module, and a CVAE encoder.

The trajectory encoder's input comprises the trajectory data X from the current task and the trajectory data X_{init} from the episodic memory, based on LSTM that can capture the time dependence of each trajectory. The output is a set of hidden features for X and X_{init}, as shown in the formula.

$$\tilde{X}, \tilde{X}_{init} = LSTM\left(Emb\left(X, X_{init}\right); W_{enc}\right) \tag{2}$$

Emb is an embedding function with a ReLU non-linearity that embeds low-dimensional trajectory coordinates into a high-dimensional vector space. Next, LSTM encodes the hidden feature $\tilde{X}, \tilde{X}_{\text{init}} = \left\{h_t^1, h_t^2, \ldots, h_t^N\right\}$, where h_t^i denotes the kinematic feature of vehicle i at time t. All LSTM encoders share the same weight W_{enc}.

The feature $\tilde{X}$ extracted from the current task data is subsequently fed into a multi-head attention module, where the features of each vehicle i are embedded into groups of queries (Q), keys (K), and values (V).

$$Q_i, K_i, V_i = MLP\left(h_T^i; W_{Q_i}, W_{K_i}, W_{V_i}\right) \tag{3}$$

Next, each head h undergoes self-attention calculation to expose the interrelationships among input vehicle features. The calculation formula for head h is given below:

$$\mathrm{head}_h = \mathrm{At}\left(Q^h, K^h, V^h\right) = \mathrm{softmax}\left(\frac{Q^h\left(K^h\right)^T}{\sqrt{d_k}}\right)V^h \tag{4}$$

Multi-head attention performs the attention calculation in parallel for H times and uses the W^O linear transformation to the same dimension of Q. The output $\tilde{X}_{mh}$ are as follows, note that the dimension is the same as the input $\tilde{X}$

$$\tilde{X}_{mh} = Norm\left(Concat\left(head_1, \ldots, head_H\right)W^o\right) \tag{5}$$

Finally, the generated attention output $\tilde{X}_{mh}$ is mapped to the latent space distribution through CVAE, known as the posterior distribution $Q\left(Z \mid \tilde{X}_{mh}, \tilde{X}_{init}\right)$. Here, $Z = \left\{z^i\right\}_{i=1:N}$, where z^i is a Gaussian random variable for the i-th agent, allowing for a sampling of new latent scene points.

CGR Decoder: The CGR decoder $P\left(X \mid Z, X_{\text{init}}\right)$ reconstructs the trajectory scene X based on the sampled latent scene points Z and the initial scene memory X_{init}. Reconstruction is achieved through two decoders: the CVAE decoder decodes the latent scene points into appropriate feature points, and the trajectory decoder decodes each consistent feature point into a trajectory scene based on the given initial scene memory X_{init}. The structure of the decoder is similar to the CGR encoder, the encoder and decoder are jointly trained. We define the parameters of the CVAE encoder as φ, denoted as $Q_\varphi\left(Z \mid X, X_{init}\right)$, and the network parameters of the CVAE decoder as θ, denoted as $P_\theta\left(X \mid Z, X_{init}\right)$. The training loss of the conditional variational autoencoder is:

$$\begin{aligned} -\mathbb{E}_{Q_\varphi(Z \mid X, X_{init})}&\left[\log P_\theta\left(X \mid Z, X_{init}\right)\right] \\ &+ \beta KL\left[Q_\varphi\left(Z \mid X, X_{init}\right) \| P(Z)\right] \end{aligned} \tag{6}$$

Where β is a hyperparameter that adjusts the importance of regularization. The decoder ultimately outputs the replayed trajectory sample, denoted as X^r.

C. Training Phase

During the training phase(see Fig.1(b)), for each task j, the input of the base model consists of the observed trajectory scene X_j and the replayed observed trajectory scene X_j^r for the current task. The output is the predicted future trajectories $\hat{Y}_j$ and $\hat{Y}_j^r$ corresponding to the current task and the replayed observed trajectory scenes. Based on the given ground truth future trajectory scenes Y_j and Y_j^r, the model can be optimized using the $\mathcal{L}_2$ loss function [2].

$$L_B = \gamma\mathcal{L}_2\left(\hat{Y}_j, Y_j\right) + (1-\gamma)\mathcal{L}_2\left(\hat{Y}_j^r, Y_j^r\right) \tag{7}$$

The input of the dual-memory model also consists of the observed trajectory scene X_j and the replayed observed trajectory scene X_j^r for the current task. The output is the generated trajectory scenes $\hat{X}_j$ and $\hat{X}_j^r$ corresponding to the current task and the replayed observed trajectory scenes. By treating the input trajectory scenes X_j and X_j^r as ground truth values, the model can be optimized using a similar loss function [2].

$$L_{DM} = \gamma\mathcal{L}_s\left(\hat{X}_j, X_j\right) + (1-\gamma)\mathcal{L}_s\left(\hat{X}_j^r, X_j^r\right) \tag{8}$$

Where $\mathcal{L}_s$ represents the sum of the conditional variational autoencoder and the reconstruction loss $\mathcal{L}_2$.

D. Replay Memory Phase

After training on task j, it enters the replay memory phase to prepare memory data that needs to be replayed for the next task $j+1$(see Fig.1(c)). Generation of replay memory data occurs within the uncertainty-aware dual-memory module. Firstly, initial scenario memory data $X_{j_{init}}$ is obtained from task j through the uncertainty-aware scenario memory. Then, the conditional variational autoencoder model uses the initial scenario memory data $X_{j_{init}}$ and trajectory data X_j from task j to generate pseudo data X_j^r as replay samples for training with the data X_{j+1} from the upcoming task $j+1$. The future trajectory ground truth values Y_j^r of the pseudo data X_j^r are obtained from the base model trained on task j.

V. Experiment

A. Datasets and Setting

We evaluated our CTP-UGR in CL tasks on two real-world datasets, NGSIM [18] and INTERACTIONA [19]. We constructed multiple distinct tasks for CL, considering time and different scenarios. For NGSIM we divided six task-scene datasets from US-101 and I-80 based on time and scenarios. As for INTERACTION, we also organized six task-scene datasets based on the three types of road scenarios, with an equal number of samples in each scenario dataset.

For each task-scene dataset, we divided them into separate training sets (70%), validation sets (10%), and test sets (20%). In the dataset, a complete trajectory comprises an 8-second time span, which is further divided into a 3-second observation period and a 5-second future period.

B. Metrics

We choose two evaluation metrics: root mean square error (RMSE) and average displacement error (ADE), to assess the prediction results on the NGSIM and INTERACTION datasets respectively. Additionally, we define two evaluation metrics, average prediction error (APE) and average forgetting rate (AFR) following the concept mentioned in [20] to assess the model's efficacy of CL. The test error of the model trained on task scenario i and tested on task scenario j is defined as $R_{i,j}$. Here, M represents the total number of task scenarios.

The APE evaluates the average prediction error of the model after training on all tasks.

$$\mathrm{APE} = \frac{1}{M(M+1)/2}\sum_{j \leq i}^{M} R_{i,j} \tag{9}$$

The AFR evaluates the average performance decline of the model on old task scenarios after learning new task scenarios to assess the degree of catastrophic forgetting in CL.

$$\text{AFR} = \frac{1}{M(M-1)/2} \sum_{i=2}^{M} \sum_{j<i}^{M} R_{i,j} - R_{j,j} \tag{10}$$

C. Baselines

We refer to our proposed method as CTP-UGR. Due to the scarcity of research on CTP, we refer to the research method of [1] and design several models for comparison. To ensure fairness, we use the base model introduced in Section IV-A as the trajectory prediction model in all the methods.

1). Accumulate Learning(AL): This method is a form of replay, but all the previous task samples are replayed during each new task training. This method represents the optimal performance of the model considered as a lower bound (LB).

2). Continual Learning with No Replay(CL-NR): This is a continual learning trajectory prediction framework without any replay mechanism. This method represents the worst performance of the model considered as an upper bound (UB).

3). Elastic Weight Consolidation (EWC) [11]: This is a classic continual learning method based on regularization, which alleviates catastrophic forgetting by adding a regularization term to the objective function of the model to constrain the model parameters between new and old tasks.

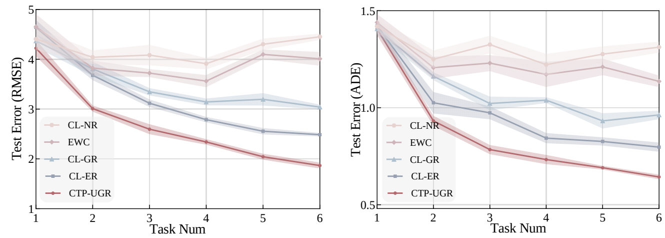
4). Continual Learning with Generative Replay(CL-GR) [6]: This is a generative replay model that exclusively employs a VAE without any conditional components.

5). Continual Learning with Experience Replay(CL-ER) [3]: This is a continual learning trajectory prediction framework based on experience replay. It explicitly stores sampled data in a memory buffer for replay.

D. Effectiveness of CTP-UGR Model

We conducted a comparative analysis between our proposed CTP-UGR and baseline methods. All models were trained on 6 task scenarios from two datasets. For training, we set the batch size to 64 and employed the Adam optimizer with a fixed learning rate of 0.001. For our model CTP-UGR, the number of multi-heads is set to 4, the hyperparameter β in the CVAE loss is set to 1.0, and the hyperparameter γ in the model loss is set to 0.5. The generated memory size is set at 10% of all previously learned datasets.

Initially, we assessed the average performance of each model on the testing data across all task scenarios. Fig. 2 illustrates this result, showing the performance of all baseline models on each training task and the trend of performance variations. The phenomenon of catastrophic forgetting is clearly evident, particularly in the CL-NR and EWC models. They also highlight a notable advantage of our CTP-UGR over CL-GR. Likewise, our model outperforms CL-ER, as CTP-UGR can generate representative and discriminative trajectory scene data without relying on an extensive memory space to store explicit memory data.



(a) NGSIM (b) INTERACTION

Fig. 2. Evaluation of overall performance (tested with a mixed dataset of all scenarios) in NGSIM and INTERACTION Dataset

TABLE I
EVALUATION RESULTS OF THE METHODS ON CONTINUAL TRAJECTORY PREDICTION BASELINE IN TERMS OF PREDICTION ERROR

	NGSIM(RMSE)		INTERACTION(ADE)	
Methods	APE	AFR	APE	AFR
AL(LB)	1.453	0.173	0.561	0.024
CL-NR(UB)	4.385	2.087	1.316	0.302
EWC	3.876	1.525	1.147	0.266
CL-GR	2.981	0.976	0.962	0.187
CL-ER	2.443	0.620	0.793	0.126
CTP-UGR(ours)	**1.885**	**0.268**	**0.643**	**0.056**

Secondly, we evaluated the CL performance metrics, APE and AFR, as depicted in Table I. Notably, our CTP-UGR surpasses several baseline models, exhibiting superior performance and effectively mitigating catastrophic forgetting. The performance of CTP-UGR is very close to the AL model.

E. Ablation Study on CTP-UGR Componets

We conducted an ablation study to examine the key components of CTP-UGR. We devised two additional models: 1) CTP-UGR with no Uncertainty-Aware (CTP-UGR-NUA). This simplified version of CTP-UGR excludes the Uncertainty-Aware Episodic Memory. Its replay data is randomly sampled from episodic memory. 2) CTP-UGR with Low-quality Generative Replay (CTP-UGR-LGR). This alternative version of CTP-UGR disables the attention-based trajectory encoder in the conditional generative replay model.

TABLE II
ABLATION STUDY ON THE KEP COMPONENTS OF THE CTP-UGR

	NGSIM(RMSE)		INTERACTION(ADE)	
Methods	APE	AFR	APE	AFR
CTP-UGR-NUA	2.339	0.481	0.769	0.106
CTP-UGR-LGR	2.014	0.327	0.692	0.071
CTP-UGR(ours)	**1.885**	**0.268**	**0.643**	**0.056**

The results are reported in Table II, revealing the substantial contributions of the individual components of CTP-UGR to the overall performance of the model. Specifically, CTP-UGR-LGR exhibits superior performance compared to

TABLE III
ROBUSTNESS ANALYSIS RESULTS OF THE MODEL

	NGSIM			INTERACTION	
Methods	APE	AFR	Methods	APE	AFR
AL-RGNN(LB)	1.216	0.147	AL-TNT(LB)	0.442	0.019
CL-NR-RGNN(UB)	3.745	1.862	CL-NR-TNT(UB)	1.132	0.247
CTP-UGR(ours)	1.885	0.268	CTP-UGR(ours)	0.643	0.056
CTP-UGR-RGNN	**1.449**	**0.193**	CTP-UGR-TNT	**0.529**	**0.041**

CTP-UGR-NUA, highlighting the effectiveness of our proposed uncertainty-aware conditional generative replay model. Furthermore, in comparison to Table I, the performance of CTP-UGR-NUA surpasses that of CL-GR, indicating that our generative model can generate trajectory data that closely resembles real-world scenarios.

F. Robustness Analysis of CTP-UGR

To further validate the robustness of CTP-UGR, we selected two state-of-the-art TP models, Graph Recurrent Neural Network (GRNN) [9] and Target-driven Trajectory prediction (TNT) [10], as replacements for our base model. We designated the combined models as CTP-UGR-GRNN and CTP-UGR-TNT, respectively. The results are presented in Table III. Compared to Table I, it is apparent that employing more advanced trajectory prediction models as baseline models leads to significant performance improvements, as evident from the APE and AFR metrics. Moreover, this approach effectively enhances overall performance and mitigates the extent of catastrophic forgetting. Intuitively, the utilization of more complex trajectory prediction models enables better capturing of complex interaction information between trajectories, enriches the feature extraction from trajectory data, and results in more refined prediction performance.

VI. CONCLUSION

This article addresses the problem of continual trajectory prediction and proposes a novel approach CTP-UGR, which combines the benefits of rehearsal and pseudo-rehearsal. Through the effectiveness and ablation analysis, we demonstrate that CTP-UGR is capable of generating realistic, representative, and discriminative data, and avoiding high memory requirements, thereby mitigating the catastrophic forgetting. Lastly, we demonstrate the robustness of our method.

ACKNOWLEDGMENT

This work is partially supported by NSFC (No. 61972069, 61836007, 61832017, 62272086), Shenzhen Municipal Science and Technology R&D Funding Basic Research Program (JCYJ20210324133607021), Municipal Government of Quzhou under Grant No. 2022D037, and Key Laboratory of Data Intelligence and Cognitive Computing, Longhua District, Shenzhen.

REFERENCES

[1] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.
[2] G. M. Van de Ven, H. T. Siegelmann, and A. S. Tolias, "Brain-inspired replay for continual learning with artificial neural networks," *Nature communications*, vol. 11, no. 1, p. 4069, 2020.
[3] H. Wang, W. Xiong, M. Yu, X. Guo, S. Chang, and W. Y. Wang, "Sentence embedding alignment for lifelong relation extraction," *arXiv preprint arXiv:1903.02588*, 2019.
[4] J. Bang, H. Kim, Y. Yoo, J.-W. Ha, and J. Choi, "Rainbow memory: Continual learning with a memory of diverse samples," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8218–8227.
[5] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
[6] J. Ramapuram, M. Gregorova, and A. Kalousis, "Lifelong generative modeling," *Neurocomputing*, vol. 404, pp. 381–400, 2020.
[7] E. Verwimp, M. De Lange, and T. Tuytelaars, "Rehearsal revealed: The limits and merits of revisiting samples in continual learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9385–9394.
[8] H. Ma, Y. Sun, J. Li, M. Tomizuka, and C. Choi, "Continual multi-agent interaction behavior prediction with conditional generative memory," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8410–8417, 2021.
[9] X. Mo, Y. Xing, and C. Lv, "Graph and recurrent neural network-based vehicle trajectory prediction for highway driving," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 1934–1939.
[10] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid *et al.*, "Tnt: Target-driven trajectory prediction," in *Conference on Robot Learning*. PMLR, 2021, pp. 895–904.
[11] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.
[12] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," *arXiv preprint arXiv:1708.01547*, 2017.
[13] Y. Wu, A. Bighashdel, G. Chen, G. Dubbelman, and P. Jancura, "Continual pedestrian trajectory learning with social generative replay," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 848–855, 2022.
[14] L. Knoedler, C. Salmi, H. Zhu, B. Brito, and J. Alonso-Mora, "Improving pedestrian prediction models with self-supervised continual learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4781–4788, 2022.
[15] S. Ebrahimi, M. Elhoseiny, T. Darrell, and M. Rohrbach, "Uncertainty-guided continual learning with bayesian neural networks," *arXiv preprint arXiv:1906.02425*, 2019.
[16] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.
[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
[18] V. Alexiadis, J. Colyar, J. Halkias, R. Hranac, and G. McHale, "The next generation simulation program," *Institute of Transportation Engineers. ITE Journal*, vol. 74, no. 8, p. 22, 2004.
[19] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle *et al.*, "Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps," *arXiv preprint arXiv:1910.03088*, 2019.
[20] N. Díaz-Rodríguez, V. Lomonaco, D. Filliat, and D. Maltoni, "Don't forget, there is more than forgetting: new metrics for continual learning," *arXiv preprint arXiv:1810.13166*, 2018.