Continual Multi-Agent Interaction Behavior Prediction With Conditional Generative Memory

Hengbo Ma, Yaofeng Sun, Jiachen Li, *Member, IEEE*, Masayoshi Tomizuka, *Life Fellow, IEEE*, and Chiho Choi

***Abstract*—Multi-agent trajectory prediction plays a crucial role in robotics and autonomous driving. The current mainstream research focuses on how to achieve accurate prediction on one large dataset. However, whether the multi-agent trajectory prediction model can be trained with a sequence of datasets, i.e., continual learning settings, remains a question. Can the current prediction methods avoid catastrophic forgetting? Can we utilize the continual learning strategy in the multi-agent trajectory prediction application? Motivated by the generative replay methods in continual learning literature, we propose a multi-agent interaction behavior prediction framework with a graph-neural-network-based conditional generative memory system to mitigate catastrophic forgetting. To the best of our knowledge, this work is the first attempt to study the continual learning problem in multi-agent interaction behavior prediction problems. We empirically show that several approaches in literature indeed suffer from catastrophic forgetting, and our approach succeeds in maintaining a low prediction error when datasets come in a sequential way. We also conduct an ablation analysis to show the effectiveness of our proposed approach.**

***Index Terms*—Continual learning, intelligent transportation systems, multi-agent systems, interactive behaviors.**

I. Introduction

PREDICTING the possible future trajectories of surrounding traffic participants in different scenarios is essential to achieve the efficiency and safety of an autonomous driving system. The complicated interaction behaviors are attributed to many aspects, such as various complex road geometries [1]–[3] and multiple traffic agents [4], [5]. There are many existing works devoted to providing practical approaches by considering as many factors as possible given an enormous dataset [6]. However, there is little work investigating whether there is a multi-agent prediction approach working on the datasets continuously coming in a sequential way, and to our best knowledge, there is also no detailed investigation about the performance of existing multi-agent behavior prediction models in the continual learning settings.

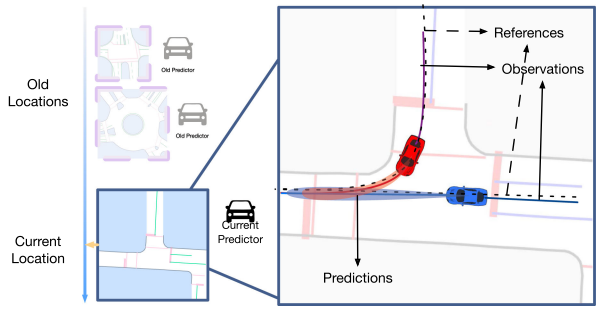


Fig. 1. Multi-agent interaction behavior prediction in the continual learning setting. The faded scenarios demonstrate the datasets with which the prediction system has been trained before. The datasets of the faded scenarios are no longer fully accessible. The prediction system takes the references information and observations as input to predict the distribution of future trajectories.

With the fast development of hardware and autonomous driving infrastructure, the amount of data increases rapidly every day. It becomes not efficient to retrain a prediction model on the increasing datasets. The ideal way is to update the trained model on the new datasets without access to the old datasets. Also, those new datasets can be collected in new scenarios which are unseen before [7]–[9]. The current literature studies how to achieve a good prediction performance based on the datasets with all the scenarios. However, current methods may not work when the datasets are collected at different scenarios which are not available simultaneously. Intuitively, since the interaction behaviors at the new location may differ from the old ones remarkably due to the very different road layout, the model may prefer to "fit" more on the current one rather than the old locations if we continually train our model on the new location without access to the old ones. Then the model will "forget" what it has learned before and perform worse on the previous locations. This phenomenon is well known as the "catastrophic forgetting issue" in the continual learning area [10]. Fig. 1 is a demonstration of continual learning settings in the multi-agent interaction prediction. Although several works have investigated the adaptation in behavior prediction [11], it has significant differences with continual learning. Adaptive or transfer learning merely cares about whether the model can adapt to new datasets. In contrast, continual learning concerns the performance on both the new and old datasets [10].

Continual learning has been studied in many different areas, such as computer vision, machine learning, cognitive science, and neuroscience. Most of the current works focus on image

Manuscript received April 7, 2021; accepted July 26, 2021. Date of publication August 12, 2021; date of current version September 9, 2021. This letter was recommended for publication by Associate Editor H. Myung and Editor Y. Choi upon evaluation of the reviewers' comments. *(Hengbo Ma and Yaofeng Sun contributed equally to this work.) (Corresponding author: Hengbo Ma.)*

Hengbo Ma and Jiachen Li are with the Honda Research Institute, San Jose, California 95134 USA, and also with the Department of Mechanical Engineering, University of California, Berkeley, CA 94720 USA (e-mail: hengbo_ma@berkeley.edu; jiachen_li@berkeley.edu).

Yaofeng Sun is with the Peking University, Beijing 100871, China (e-mail: sunyaofeng8@pku.edu.cn).

Masayoshi Tomizuka is with the Department of Mechanical Engineering, University of California, Berkeley, CA 94720 USA (e-mail: tomizuka@me.berkeley.edu).

Chiho Choi is with the Honda Research Institute, San Jose, California 95134 USA (e-mail: cchoi@honda-ri.com).

Digital Object Identifier 10.1109/LRA.2021.3104334

classification and reinforcement learning tasks [12]–[14]. Several approaches have been developed from different perspectives. For instance, regularization methods [15] were proposed from the optimization view by constraining the difference between old parameters and new learned parameters; pseudo-rehearsal methods like generative replay [16] inspired by neuroscience were proposed to generate the old data from random noises, etc. Though such works have achieved remarkable results on the image classification task, it does not imply that those methods could work in high-dimensional regression problems such as multi-agent interaction behavior prediction. To our knowledge, there is little related research studying the continual learning problems for trajectory prediction in the autonomous driving domain, especially for multi-agent trajectory prediction.

In our work, we adopt the concepts of pseudo-rehearsal approaches [16] in the continual learning literature. We propose a graph-neural-network-based double memory system that can generate similar interaction behaviors compared with the ground truth data. We update our multi-agent interaction behavior predictor with the generated dataset and the new dataset together to reduce the catastrophic forgetting issue. To summarize, our contributions are three folds:

- To the best of our knowledge, our work is the first to investigate the continual learning problems in multi-agent interaction behavior prediction. We show that several current prediction approaches suffer from forgetting problems when data is coming in sequence.
- We propose a graph-neural-network-based continual multi-agent trajectory prediction framework. In this framework, we propose a conditional generative memory model to mitigate catastrophic forgetting. We also design an episodic memory to store the initial graphs of multi-agent trajectories, which are provided to the conditional generative memory model.
- We validate our approach on two datasets. We show that our method effectively mitigates the catastrophic forgetting problems. An ablation analysis is provided to show the necessity and efficiency of the proposed conditional generative memory and episodic memory buffer, especially compared with the method directly using generative replay.

II. Related Work

A. Social Interaction Modeling and Prediction

Behavior and trajectory prediction is crucial for autonomous driving, especially when considering the multi-agent interaction in different scenarios. Several methods have been developed to predict interactive behaviors among multiple pedestrians [17]–[19] and vehicles [2], [20], [21]. Some of these methods use learning-based approaches such as generative models [22]–[25], probabilistic graphical models [26] and dynamic Bayesian network [27]. In recent years, more sophisticated models such as graph neural network (GNN) and its variations are proposed [28]–[30] and applied to trajectory prediction [4], [31], [32]. In some literature, maps have been widely used to provide context information [1]–[3]. [2] proposes to use convolutional neural networks as an encoder. [6] focuses on jointly detecting vehicles and predicting future trajectories from high-definition maps and LiDAR information. However, most of these methods are trained and tested on the datasets in which many vehicles are driving on straight roads or roads with similar layouts [33]. In our work, we propose the graph-neural-network-based predictor and memory systems to capture the interaction behaviors. We also leverage the Frenét coordinate system to represent the map information.

B. Continual Learning

Continual learning has been receiving more attention in recent years. There are three major categories of methods: rehearsal methods [13], [34], regularization methods and architecture methods [14], [35]. Our proposed approach is inspired by the rehearsal-based method. The core idea of this kind of approaches is using a memory system to "remember" the seen data points, and using this memory to reduce catastrophic forgetting problems when we continually receive new datasets. Gradient episodic memory (GEM) [34] and its variant dubbed-average GEM (A-GEM) [36] use episodic memory as the constraints in the optimization. Instead of storing the data into the episodic memory buffer, [16] used a generative model to generate the seen data, and mixed it with the new dataset to optimize the current model. Although several works [37] further develop the idea of generative memory, they are all tested in the image classification or reinforcement learning tasks. To our knowledge, there are few works to validate whether such ideas can also work in regression or more complex tasks. The most related work is generative replay (GR) [16] that uses generative adversarial network (GAN) to generate previous data as memory. This method can work for image classification. However, we show that it is not enough to generate realistic trajectories in our application due to the complex spatial-temporal data structures. From this intuition, we propose a conditional generative model and combine it with a small episodic memory buffer together to generate better trajectories as memory.

III. Problem Formulation

Our goal is to train the multi-agent future trajectories predictor given a stream of datasets collected at different scenarios. We firstly formulate the domain problem (i.e., multi-agent trajectory prediction), and then introduce the formulation of the continual learning problem in our application.

A. Multi-Agent Trajectory Prediction

We assume that there are N agents in each case, and the number of agents for each case may vary. We denote the observations of N agents as $\mathbf{o} = \{\mathbf{o}^1, \mathbf{o}^2, \mathbf{o}^3, \ldots, \mathbf{o}^N\}$, where $\mathbf{o}^i = [\mathbf{p}^i_{-t_h+1:0}, \mathbf{c}^i]$. $\mathbf{p}^i_{-t_h+1:0} \in \mathbf{R}^{t_h \times d}$ is the i-th agent's history trajectories with t_h time steps. d is the dimension of state $\mathbf{p}$. $\mathbf{c}^i$ represents the waypoints of the i-th agent's reference in the Cartesian coordinate system. We define I^i as the bird-view rasterized image of $\mathbf{c}^i$. $\mathbf{c}^i$ and I^i come from a provided high-definition (HD) map. Our purpose is to predict the future trajectories of multiple agents $\mathbf{y} = \{\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3, \ldots, \mathbf{y}^N\}$ given the observations $\mathbf{o}$, i.e., the distribution $P(\mathbf{y}|\mathbf{o})$, where $\mathbf{y}^i$ is defined as $\mathbf{p}^i_{1:t_f}$ and t_f is the time horizon of the future trajectory of each agent.

B. Continual Learning Problem

Under the problem setting of continual learning, we assume that we cannot store *all* previous data. Each experiment is conducted on several datasets collected at different time and different locations. We denote $\mathcal{D}^i$ as the i-th dataset we received. We evaluate the performance of prediction systems given a sequence of datasets $\{\mathcal{D}^1, \mathcal{D}^2, \ldots, \mathcal{D}^M\}$, M is the number of

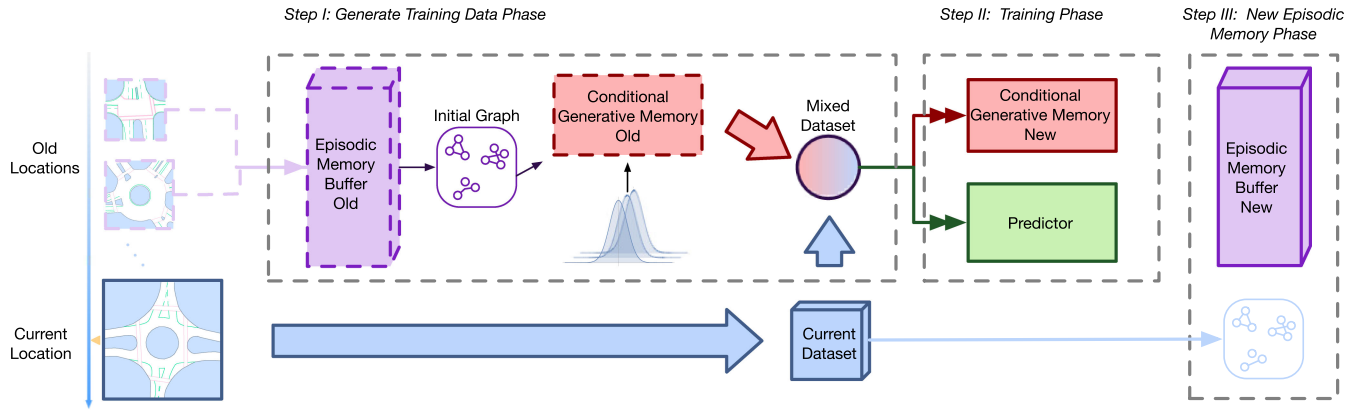


Fig. 2. The overall framework. Different datasets are collected from several locations. We assume that we cannot store the whole dataset after we optimize our model with it. **Step I**: The purple cube represents the episodic memory, which only stores the initial graph information of the previous scenarios. We sample a batch of initial graphs and use conditional generative memory to generate the trajectories. Then we mix the generated data and the current dataset together. **Step II**: We use the mixed data to train both the conditional generative memory and predictor. **Step III**: We uniformly sample a small portion of the current dataset to store in the episodic memory buffer.

datasets. Notice that if the current dataset is the k-th dataset, we do not store the past datasets $\{\mathcal{D}^1, \ldots, \mathcal{D}^{k-1}\}$.

IV. Methodology

Our proposed framework is illustrated in Fig. 2. We firstly introduce the three modules used in our proposed framework: a graph-neural-network-based predictor, an episodic memory buffer and a conditional-variational-autoencoder-based generative memory module. Then we introduce the continual learning strategy, which includes three steps. Our contribution mainly focuses on the framework design and the conditional generative memory module, and this whole approach has the potential to be incorporated with other suitable predictors.

A. Predictor

Graph Representation of Multi-agent Trajectories Our predictor computes a multi-modal probabilistic multi-agent trajectory distribution using the observation $\mathbf{o}$ of all agents. Given the reference $\mathbf{c}^i$ of each agent from $\mathbf{o}^i$, we transform the trajectory $\mathbf{p}^i$ of the i-th agent into the Frenét coordinate system and denote it as $X^i = \{d^i_{\text{lon}}, d^i_{\text{lat}}\}$, where d^i_{lon} and d^i_{lat} represent the longitudinal position and lateral position in the Frenét coordinate associated with the reference $\mathbf{c}^i$.

To get a feature representation that is invariant to the origin and the direction of the coordinate system, we use only velocity information $\delta^i = \{\dot{d}^i_{\text{lon}}, \dot{d}^i_{\text{lat}}\}$ for each agent i, where $\dot{d}^i_{\text{lon}}$ and $\dot{d}^i_{\text{lat}}$ represent the longitudinal velocity and lateral velocity w.r.t. the reference $\mathbf{c}^i$. For each edge e_{ij}, we define the edge feature as the relative position Δ^{ij} of agent j in the view of agent i. We use the rasterized image I^i to provide the future lane geometry information. Here we recenter the image at the i-th agent's current position and set the y-axis direction of the image as the velocity direction of the i-th agent. We denote the transformed image as $\tilde{I}^i$. We use a feature embedding function to process the aforementioned information to form initial node attributes v^0_i and edge attributes e^0_{ij}. Given a set of trajectory observations, we have:

$$\begin{aligned} v^0_i &= \text{MLP}((\text{CNN}(\tilde{I}^i)||\text{RNN}(\delta^i))), \\ e^0_{ij} &= \text{RNN}(\Delta^{ij}). \end{aligned} \tag{1}$$

Message Passing Graph Neural Network Following the extracted features $\{v^0_i, e^0_{ij}\}$, a fully-connected graph is constructed to represent the interaction mechanism between different agents. We denote the graph as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_i\}$ denotes the node attributes, and $\mathcal{E} = \{e_{ij}\}$ denotes the edge attributes. At the m-th message passing:

$$\begin{aligned} e^m_{ij} &= f^m_e([v^{m-1}_i, v^{m-1}_j]), \\ v^m_i &= f^m_v(\Phi[j \in \mathbf{N}(v_i)](e^m_{ij})), \ m = 1, \ldots, n. \end{aligned} \tag{2}$$

where f_e and f_v are the embedding functions for edges and nodes, respectively. The superscripts of v^m_i, e^m_{ij}, f^m_v, f^m_e denote the m-th message passing. $\Phi[j \in \mathbf{N}(v_i)](\cdot)$ aggregates the information of all the edges e_{ij} between v_i and its neighbors $\mathbf{N}(v_i)$. We use the attention mechanism similar to Graph Attention Networks (GAT) [30]:

$$\alpha^m_{ij} = \text{softmax}(e^m_{ij}), v^m_i = \sigma\left(\sum_{j \in \mathbf{N}(v_i)} \alpha^m_{ij} \mathbf{W} v^{m-1}_j\right). \tag{3}$$

Multi-modal Prediction Layer In order to capture the multi-modal interactive behaviors, we use a Gaussian mixture model (GMM) to represent the future predicted actions at different time steps:

$$\begin{aligned} w_j &= \text{softmax}(f^j_w(v^n_i)), \\ \mu_j &= f^j_\mu(v^n_i), \Sigma_j = f^j_\Sigma(v^n_i), \\ \{\dot{d}_{\text{lon},0:t_f-1}, \dot{d}_{\text{lat},0:t_f-1}\} &\sim \sum_j w_j \mathcal{N}(\mu_j, \Sigma_j), \end{aligned} \tag{4}$$

where w_j, μ_j, and Σ_j denote the weight, mean, and covariance of the j-th Gaussian function, respectively. Each Gaussian function represents the distribution of the future actions.

After obtaining the action information for each agent i, we use a first-order integrator to get the position X^i in the Frenét coordinate system. Then the predicted trajectories are transformed back to the Cartesian coordinate system according to $\mathbf{c}^i$. This procedure incorporates the road routing information directly. The loss $\mathcal{L}_{\text{P}}(\psi; \mathcal{D})$ is the log-likelihood:

$$\mathbb{E}_{(X,I)\sim\mathcal{D}}[\log P_\psi(X_{1:t_f} | X_{-t_h+1:0}, I)], \tag{5}$$

where ψ is the parameter of the predictor.

B. Double Memory System

The traditional pseudo-rehearsal methods such as the generative replay method [16] generates samples from noises. In our case, it is not enough to generate high-quality full-time-length multi-agent trajectories purely from high-dimensional noises due to the complicated spatial-temporal data structure of multi-agent trajectories. Hence, we propose a graph-neural-network-based conditional generative replay module, which is conditioned on some initial information given by the proposed episodic memory buffer.

Episodic Memory Buffer Episodic memory is defined as a kind of memory of everyday events which could be conjured or explicitly stated. It is the collection of past experiences that occurred at particular times and places [38]. In our application, one "experience" should be one case of multi-agent interaction trajectories. Some works in image classification and reinforcement learning have been proposed to construct the episodic memory by storing a fixed percentage of original data. We intend to reduce the storage of full-time-length trajectories data by only storing some initial information of the trajectories. This reduction is significant, especially when the length of the trajectory is large. We define such initial information as an initial graph $\mathcal{G}_{init} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}^i = \{X_0^i, X_{-t_h+1}^i, X_{t_f}^i, I^i, \mathbf{c}^i\}$ and the edge attribute $E^{ij} \in \mathcal{E}$ denotes whether there is an edge between node i and node j. There is no edge between two agents if their references do not have any intersection or the vehicle on one reference cannot shift to the other reference due to traffic rules. Here X_0^i is the current state defined in previous section, $X_{-t_h+1}^i$ is the state at H time steps before the current state, and $X_{t_f}^i$ is the goal position. We intend to generate the interaction behaviors conditioned on this initial information. It is reasonable that the interaction behaviors could be "conjured" if we know the trajectory tendency (given X_{-t_h+1}) and goal (given X_{t_f}) at the current state X_0. We will show in the experiments that the performance of our methods with this episodic memory buffer is significantly improved compared with directly using the vanilla generative replay methods.

Conditional Generative Memory The objective of the conditional generative memory is to solve $P(X|\mathcal{G}_{init})$, where $X = X_{-t_h+1:t_f}$ means the whole trajectories of all agents. We design a graph-neural-network based conditional variational autoencoder (CVAE) to estimate this distribution. The graph neural networks are used to capture the interaction mechanism between agents. By leveraging the conditional variational inference, we can generate realistic trajectories given the initial graph information. The overall conditional generative memory model structure is illustrated in Fig. 3.

Encoder The encoder is used to map the trajectories X to the latent variables $\mathbf{z}$, namely the posterior distribution $Q(\mathbf{z}|X, \mathcal{G}_{init})$, where $\mathbf{z} = \{\mathbf{z}^i\}_{i=1:N}$ and $\mathbf{z}^i$ is the Gaussian random variable for the i-th agent. For each agent i, the reference image I^i and the initial state information $\{X_0^i, X_{-t_h+1}^i, X_{t_f}^i\}$ are encoded by a convolutional neural network (CNN) and a multiple-layer perceptron (MLP) respectively. They are served as the conditional inputs. The trajectories $X_{-t_h+1:t_f}^i$ are encoded by a recurrent neural network (RNN). We use an MLP to integrate these three features and construct a graph neural network to process the interaction behaviors. Each node i of the GNN module outputs the mean and the covariance of $\mathbf{z}^i$.

Decoder The decoder is used to map the latent variables $\mathbf{z}$ to the trajectories X, namely $P(X|\mathbf{z}, \mathcal{G}_{init})$. Similar to the encoder, we use a CNN and an MLP to process the reference image I^i and the initial state information for each node. Then we use a GNN to capture the interaction pattern and use an RNN to output the trajectories X. The input $\mathbf{z}$ of the decoder is sampled from the output of the encoder $Q(\mathbf{z}|X, \mathcal{G}_{init})$ during training and sampled from the prior distribution $P(\mathbf{z})$ during testing. $P(\mathbf{z})$ is the standard Gaussian distribution. The encoder and decoder are trained jointly. The training loss of the CVAE $\mathcal{L}_{\mathrm{G}}(\theta, \phi; \mathcal{D})$ is:

$$\begin{aligned} &\mathbb{E}_{(X,\mathcal{G}_{init})\sim\mathcal{D}}\mathbb{E}_{Q_\phi(\mathbf{z}|X,\mathcal{G}_{init})}[\log P_\theta(X|\mathbf{z}, \mathcal{G}_{init})] \\ &- \beta\mathbb{E}_{(X,\mathcal{G}_{init})\sim\mathcal{D}}\mathcal{KL}[Q_\phi(\mathbf{z}|X, \mathcal{G}_{init})||P(\mathbf{z})], \end{aligned} \tag{6}$$

where ϕ and θ are the parameters of the encoder $Q_\phi(\mathbf{z}|X, \mathcal{G}_{init})$ and the decoder $P_\theta(X|\mathbf{z}, \mathcal{G}_{init})$. β is a parameter to adjust the importance of the second regularization.

Once we get the initial graph $\mathcal{G}_{init}$, we can sample r trajectories from the decoder $P_\theta(X|\mathbf{z}, \mathcal{G}_{init})$ by sampling r times of different $\mathbf{z}$ from the standard Gaussian distribution. Similar to the predictor, we can transform X^i to $\mathbf{p}^i$ by using the reference $\mathbf{c}^i$.

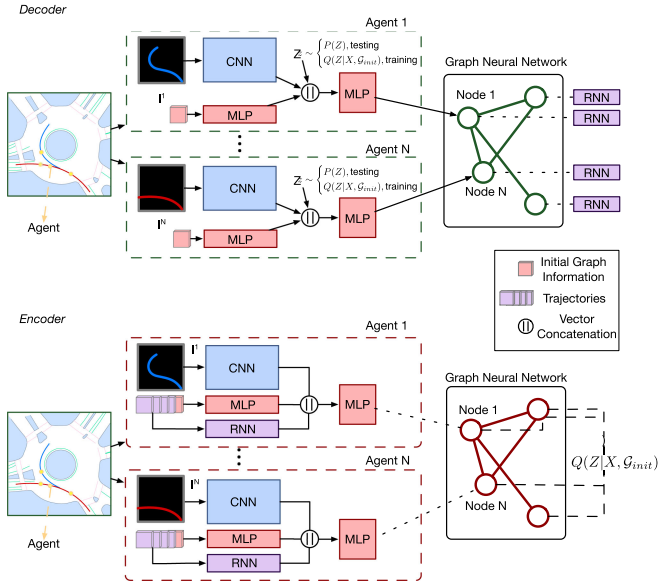


Fig. 3. The conditional generative memory model. The blue boxes, red boxes and the purple boxes represent CNN, MLP and RNN respectively. I^i demonstrates the rasterized image of the i-th agent's reference.

C. Training Strategy

There are three procedures in total: generate the training batch, train predictor and generator, and form new episodic memory. The framework is illustrated in Fig. 2.

Step I: First, we need to construct the mixed training dataset at the k-th scenario. We uniformly sample $|\mathcal{D}^k|/r$ initial graphs $\mathcal{G}_{init}$ from the episodic memory buffer $\mathcal{B}^{k-1}$. For each initial graph $\mathcal{G}_{init}$, we randomly sample r times of different $\mathbf{z}$ and use the decoder to generate r multi-agent trajectories. We denote the generated data as $\hat{\mathcal{D}}^k$. Hence, $\hat{\mathcal{D}}^k$ includes the generated real-like data of the past scenarios $\mathcal{D}^1, \mathcal{D}^2, \ldots, \mathcal{D}^{k-1}$, which serves as the replay data for the predictor and generator.

Step II: After we get the current new dataset $\mathcal{D}^k$ and the generated dataset $\hat{\mathcal{D}}^k$ from Step I, we optimize our predictor

and generative memory separately.

$$\psi^k = \min_{\psi} \gamma \mathcal{L}_{\mathrm{P}}(\psi; \mathcal{D}^k) + (1-\gamma)\mathcal{L}_{\mathrm{P}}(\psi; \hat{\mathcal{D}}^k),$$

$$\theta^k, \phi^k = \min_{\theta,\phi} \gamma \mathcal{L}_{\mathrm{G}}(\theta, \phi; \mathcal{D}^k) + (1-\gamma)\mathcal{L}_{\mathrm{G}}(\theta, \phi; \hat{\mathcal{D}}^k), \quad (7)$$

where γ is the hyperparameter to determine the importance of the current scenario. The current scenario is more important if γ is larger.

Step III: The final step is to construct the new episode memory buffer $\mathcal{B}^k$. We randomly sample a small portion of the whole number of cases in the new dataset and store their initial graphs into the episodic memory buffers. In our application, we only allow storing the initial graphs of ten percent of the cases for each $\mathcal{D}^k$. The three steps above will be repeatedly executed as long as the new dataset is available.

V. Experiments

We conduct our experiments with INTERACTION dataset [7] collected in the USA, InD dataset [8] and RounD dataset [9] collected in Germany. First, we show that baselines indeed suffer from catastrophic forgetting and our method can reduce it. Then we conduct an ablative analysis to show that our method is better than applying the generative replay method directly and our approach can achieve similar performance with significantly less memory compared with real data replay.

A. Dataset and Preprocessing

To evaluate our approach's effectiveness, we use INTERACTION dataset, InD and RounD dataset. The aforementioned datasets include many complicated interaction scenarios like roundabouts and intersections, which is suitable to test the multi-agent interaction behaviors. Also, the cases of these datasets are divided by their scenarios originally. Since the driving behaviors are significantly different between different scenarios, these datasets are suitable to test the continual learning performance. We selected four scenarios for each dataset, and each scenario has a similar number of cases. Both datasets provide the HD map from which we can extract the references information. For both datasets, we use four time steps as observation and predict eight time steps of future trajectories. The interval between time steps is 0.5 s for INTERACTION datasets and 0.4 s for RounD/InD datasets.

B. Metrics

Prediction Metrics To evaluate the probabilistic prediction, we use the similar metrics in [39], the minimum average displacement error (ADE) and minimum final displacement error (FDE) over k samples:

$$\mathrm{ADE} = \frac{1}{Nt_f} \sum_{i=1}^{N} \min_{j \in \{1,\dots,k\}} \left\{ \sum_{t=1}^{t_f} ||\mathbf{p}_t^i - \hat{\mathbf{p}}_t^i(j)||_2 \right\},$$

$$\mathrm{FDE} = \frac{1}{N} \sum_{i=1}^{N} \min_{j \in \{1,\dots,k\}} \{||\mathbf{p}_{t_f}^i - \hat{\mathbf{p}}_{t_f}^i(j)||_2\}. \quad (8)$$

Continual Learning Metrics For evaluating the continual learning performance, we adopt the concepts mentioned in [40] and adapt them to our application. Here we define two metrics: average error (AER) and average forgetting (FGT). We denote the testing error (ADE / FDE) at the j-th scenario after training on the i-th scenario as $R_{i,j}$. Let M be the number of scenarios, AER and FGT are defined as:

$$\mathrm{AER} = \frac{1}{M(M+1)/2} \sum_{j \le i}^{M} R_{i,j},$$

$$\mathrm{FGT} = \frac{1}{M(M-1)/2} \sum_{i=2}^{M} \sum_{j<i}^{M} R_{i,j} - R_{j,j}. \quad (9)$$

AER measures the average performance of all scenarios across all training datasets. FGT measures the average decrease of performance on the old scenarios after seeing the new scenarios, which can represent the degree of catastrophic forgetting. The greater the value of FGT is, the more forgetting occurs. Please refer to [40] for more details.

C. Overall Performance Evaluation

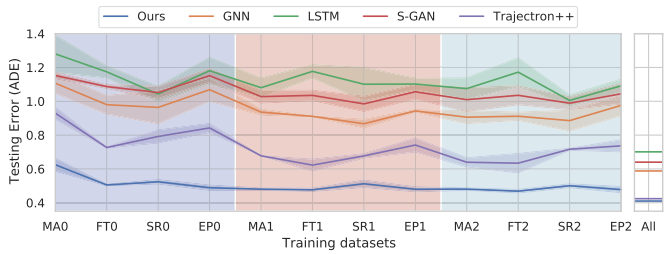
Experiment Settings For this experiment, we use four different locations from both the INTERACTION dataset and the RounD/InD dataset. The cases of each location are divided into three parts equally by time stamps since we intend to simulate that we repeatedly collect data from different locations. Different parts are regarded as different datasets. The four scenarios from the INTERACTION dataset are MA, FT, SR, and EP (Including EP-R and EP-T). The sequence of tasks is $\mathcal{D}^{\mathrm{MA0}}, \mathcal{D}^{\mathrm{FT0}}, \mathcal{D}^{\mathrm{SR0}}, \mathcal{D}^{\mathrm{EP0}}, \mathcal{D}^{\mathrm{MA1}}, \dots, \mathcal{D}^{\mathrm{EP2}}$, where under the same scenario, e.g., $\mathcal{D}^{\mathrm{MA0}}, \dots, \mathcal{D}^{\mathrm{MA2}}$ means three different datasets collected at different time at MA. Our framework will be trained in this order. The four scenarios from the RounD/InD dataset are R, IA, IB, and IC. R indicates Neuweiler of the RounD dataset. IA, IB, and IC indicate Bendplatz, Frankenburg, and Heckstrasse in the InD dataset. We evaluate the prediction performance for each scenario after we train the model with the current dataset. We compare our method with several baselines and recent works including long-short term memory (LSTM), GNN, S-GAN [17] and Trajectron++ [5]. We provide the same input information for all the methods. LSTM is equipped with a GMM model in order to generate the probabilistic prediction. GNN is a simple message passing graph neural network without any special design. We trained five models randomly to get the means and variances of performance. **Qualitative Analysis** We illustrate the prediction performance of several representative cases with different numbers of agents at different scenarios of the INTERACTION dataset in Fig. 4. We demonstrate that we can get accurate prediction at all the scenarios after training on a sequence of datasets. Furthermore, we find that if the references are relatively straight, the variance of predicted trajectories is small.

Quantitative Analysis We calculate the average performance of the testing data across all scenarios. This value represents the overall performance of the method after training on a sequence of datasets. We illustrate the ADE error for both INTERACTION and RounD/InD datasets in Fig. 5.

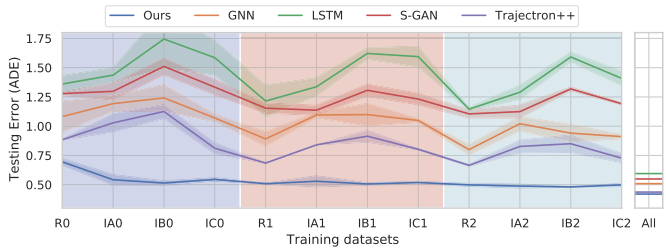
In Fig. 5, the other baseline models have bad performance even they could be trained on the same scenario again. We observed that baselines have a large error during the whole testing phase. It implies that it may only have good performance when the training scenario and testing scenario are the same. This conjecture is validated in Fig. 6.

(a) MA (b) FT (c) SR (d) EP-R (e) EP-T

Fig. 4. The visualization of multi-agent trajectory prediction results. We represent the ground truth trajectories as box markers. The star markers represent the starting points of the historical trajectory. We use kernel density estimation (KDE) to fit the sampled trajectories, shown as density maps in the pictures. The grey dash lines are the references of each vehicle. The predicted trajectory with the minimum ADE is illustrated in a solid line.

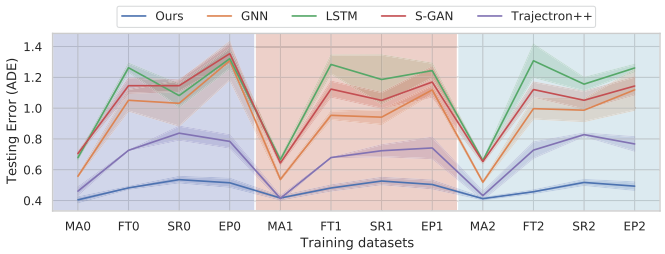


(a) INTERACTION dataset

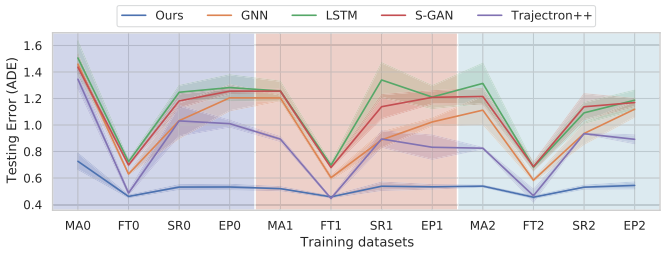


(b) RounD/InD dataset

Fig. 5. Evaluation of overall performance (tested with a mixed dataset of all scenarios) in the continual learning setting. The column "All" means that the methods are trained with the mixed datasets of different scenarios, i.e., the non-continual learning setting.



(a) Evaluation on MA



(b) Evaluation on FT

Fig. 6. ADE for selected scenarios, MA and FT in INTERACTION dataset. Models are tested on a particular scenario to show the catastrophic forgetting issue clearly.

Notice that the average error is for the data of all scenarios. The first point at MA0 means that we train models on MA0 dataset and evaluate on a mixed testing data of all scenarios. We observe that our model has better performance at MA0 than the other baselines. This advantageous performance of our predictor could be attributed to the reason that we directly use the reference information $\mathbf{c}$ (by using Frenét coordinate transformation). It enables our predictor to have a relatively good performance on the other unseen scenarios. While the other baseline models incorporate the information of references from rasterized images, they have to extract the useful information from the image directly. It also can be validated in Fig. 6(b). For instance, in Fig. 6(b), we find that our approach can have a better performance on FT than the others even though we only used the first dataset MA0 for training.

To find out why baseline models have bad performance when the training datasets come in a sequential way, we investigate the prediction performance on each scenario. We illustrate the prediction error of two representative scenarios of the INTERACTION dataset in Fig. 6. We notice that the testing performances of all the models are good when the current training and the testing scenarios are the same. For instance, the state-of-the-art model, Trajcetron++, can achieve a very low prediction error when testing on the same scenario as the current training one. It implies that the bad overall performances in Fig. 6 are not only caused by the capability of the model itself but also attributed to the catastrophic forgetting issues. For instance, in Fig. 6(a), when we test our prediction performance on MA, the errors of baseline models rapidly increase when they are trained on the following scenarios, which is precisely the catastrophic forgetting phenomenon. Our model significantly reduces the catastrophic forgetting problems compared with all the methods without continual learning, i.e., the fluctuation in Fig. 6 is significantly smaller than the others.

We also compare the performance of different approaches between the continual learning setting and non-continual learning setting, i.e., training on the entire dataset of all scenarios. The non-continual learning setting (column "All" in Fig. 5) can serve as the lower bound of the error of the models in the continual learning settings. In Fig. 5 column "All," we see that Trajectron++ and Ours can achieve the similar best performance when training on the entire datasets. It demonstrates that our predictor module can achieve state-of-the-art performance. In

TABLE I
THE PERFORMANCE OF ADE / FDE (METERS) IN ABLATION ANALYSIS FOR INTERACTION DATASET

	Order 1		Order 2		Order 3		Order 4	
	AER	FGT	AER	FGT	AER	FGT	AER	FGT
Ours	**0.441/0.900**	**0.058/0.189**	**0.464/0.952**	**0.045/0.122**	**0.466/0.891**	**0.060/0.188**	**0.449/0.956**	**0.058/0.166**
Ours w/o CL	0.558/1.247	0.249/0.814	0.603/1.431	0.322/0.952	0.577/1.265	0.245/0.755	0.556/1.237	0.265/0.806
GR	0.543/1.278	0.270/0.836	0.567/1.263	0.243/0.718	0.534/1.198	0.179/0.486	0.537/1.240	0.245/0.731
EWC	0.502/1.109	0.198/0.609	0.543/1.235	0.217/0.661	0.512/1.115	0.129/0.308	0.489/1.084	0.175/0.506

TABLE II
THE PERFORMANCE OF ADE / FDE (METERS) IN ABLATION ANALYSIS FOR ROUND/IND DATASET

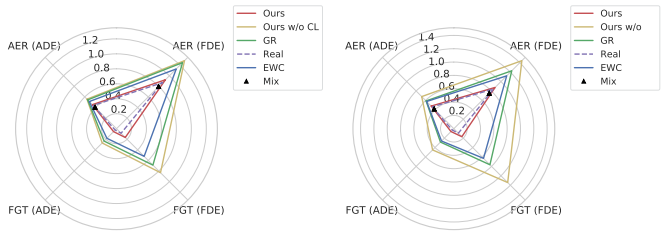
	Order 1		Order 2		Order 3		Order 4	
	AER	FGT	AER	FGT	AER	FGT	AER	FGT
Ours	**0.509/0.938**	**0.112/0.284**	**0.466/0.832**	**0.036/0.113**	**0.463/0.846**	**0.029/0.080**	**0.484/0.889**	**0.067/0.203**
Ours w/o CL	0.669/1.376	0.424/1.083	0.674/1.453	0.442/1.135	0.740/1.609	0.559/1.407	0.644/1.337	0.379/0.947
GR	0.625/1.310	0.320/0.841	0.568/1.177	0.243/0.674	0.576/1.203	0.288/0.786	0.590/1.226	0.286/0.762
EWC	0.617/1.236	0.325/0.791	0.548/1.028	0.205/0.464	0.558/1.085	0.227/0.579	0.576/1.116	0.278/0.665

addition, comparing the testing performance at column EP2 in which the models have seen all the datasets with the testing performance at column "All" in Fig. 5 (a), our method can also achieve similar performance in the continual learning setting to the non-continual learning setting. It shows that our method reduces catastrophic forgetting remarkably. Meanwhile, we observe large gaps between the continual learning and the non-continual learning setting for each baseline.

D. Ablation Analysis

In this section, we demonstrate the effectiveness of our approach by comparing the following ablation models: i) our framework (Ours); ii) our predictor without double memory system (Ours w/o CL); iii) our predictor with vanilla generative replay (GR), i.e., the full-length trajectories are generated purely from a Gaussian random variable; iv) our predictor with elastic weighted consolidation (EWC). In order to show that the comparisons are invariant to the order of scenarios, we randomly sample four different orders. Without loss of generality, we use the whole dataset for each scenario. We show the results in Table I and Table II for both datasets.

We demonstrate that the results are consistent under different orders. From Table I and II, we show that Ours improves at least 19% in ADE for AER and 73% in ADE for FGT compared with Ours w/o CL. It implies that our conditional generative memory system is effective, and only using the Frenét-based predictor is not enough to reduce the catastrophic forgetting. Comparing Ours w/o CL with GR, we find that GR only slightly improves the performance. It shows that directly using generative replay does not work well in our application. This comparison illustrates the necessity of the proposed episodic memory buffer to store the initial graph information. Although GR and EWC do not use replay buffer, they cannot mitigate catastrophic forgetting effectively in our application. In Fig. 7, we compute the average performance across different orders and demonstrate the comparison. We also include the model (Real), which uses full-length ground truth data replay to serve as the ideal performance bound of our methods. The difference between Real and Ours is that Ours only stores initial graph information and generates the whole trajectories by conditional generative memory, while Real stores the full-length ground truth multi-agent trajectories.



(a) INTERACTION dataset (b) RounD/InD dataset

Fig. 7. Average performances across different orders for both datasets. "Mix" shows the prediction error of models trained on the whole datasets including all scenarios, i.e. the common settings without continual learning. Details and memory comparison analysis are in Section V-D.

This difference allows Ours reduces about 87% memory usage compared with Real. Meanwhile, Ours can achieve almost the same performance as Real (shown in Fig. 7). It demonstrates our method's effectiveness and shows that utilizing generative models is possible to reduce catastrophic forgetting in multi-agent trajectory prediction.

VI. IMPLEMENTATION DETAILS

Our predictor and conditional generative memory are trained using Adam optimizer with the batch size 64. We use 0.0005 as the learning rate for each task. The MLP modules have three fully connected layers with the activation function ReLU. The RNN modules use the GRU blocks. The CNN modules have five convolution layers with channels 16, 32, 64, 128, 256. The GNN modules have two rounds of message passing. The hidden dimension of our models is 256. For the conditional generative memory, the dimension of the Gaussian random variable z^i for each node i is 4 and the hyperparameter β in the VAE loss is 1.0. For the predictor, the Gaussian mixture model has 4 kernels, and all of the methods sample 20 trajectories to calculate the metrics. We use $\gamma = 0.5$ for training.

VII. CONCLUSION

How to continually learn a good trajectory predictor is a potential problem in the near future. In this work, we attempt

to reduce the catastrophic forgetting in the complicated multi-agent spatial-temporal prediction. We give an approach based on graph representation and the conditional generative memory model to show the possibility of realizing continual learning in the multi-agent trajectory prediction tasks. Such a problem is under-investigated since most of the existing continual learning works focus on image classification and reinforcement learning while relational reasoning and feature representation appeal more attention in the trajectory prediction literature. Hence, this work paves the way for both the continual learning and the trajectory prediction communities. For the continual learning community, we demonstrate that continual learning can also work in high-dimensional regression tasks. For the trajectory prediction community, we show that there is an elegant solution to deal with sequentially increasing data. This work also raises many interesting questions and future directions including but not limited to i) Is there any way to completely abandon the episodic memory buffer? ii) Is it possible to design a predictor which has good zero-shot prediction in new scenarios? We believe that solving such questions can improve performance in the future.

REFERENCES

[1] J. Hong, B. Sapp, and J. Philbin, "Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2019, pp. 8454–8462.

[2] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," *Proc. Conf. Robot Learn.*, 2019, pp. 86–99.

[3] J. Gao *et al.*, "VectorNet: Encoding HD maps and agent dynamics from vectorized representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2020, pp. 11525–11533.

[4] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "Trafficpredict: Trajectory prediction for heterogeneous traffic-agents," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, pp. 6120–6127.

[5] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Comput. VisionECCV 2020: 16th Eur. Conf., Proc., Part XVIII 16*. Springer, 2020, pp. 683–700

[6] S. Casas, W. Luo, and R. Urtasun, "IntentNet: Learning to predict intention from raw sensor data," in *Proc. Conf. Robot Learn*., 2018, pp. 947–956.

[7] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle, and M. Tomizuka, "INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps," 2019, *arXiv:1910.03088*.

[8] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The inD dataset: A drone dataset of naturalistic road user trajectories at German intersections," in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 1929–1934.

[9] R. Krajewski, T. Moers, J. Bock, L. Vater, and L. Eckstein, "The round dataset: A drone dataset of road user trajectories at roundabouts in Germany," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2020, pp. 1–6.

[10] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, "Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges," *Inf. Fusion*, vol. 58, pp. 52–68, 2020.

[11] W. Si, T. Wei, and C. Liu, "AGen: Adaptable generative prediction networks for autonomous driving," in *Proc. IEEE Intell. Veh. Symp.*, 2019, pp. 281–286.

[12] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, pp. 3302–3309, 2018.

[13] J. von Oswald, C. Henning, J. Sacramento, and B. F. Grewe, "Continual learning with hypernetworks," in *Proc. Int. Conf. Learn. Representations*, 2020.

[14] T. Adel, H. Zhao, and R. E. Turner, "Continual learning with adaptive weights (CLAW)," in *Proc. Int. Conf. Learn. Representations*, 2020.

[15] J. Pomponi, S. Scardapane, V. Lomonaco, and A. Uncini, "Efficient continual learning in neural networks with embedding regularization," *Neurocomputing*, 397, Feb. 2020, pp. 139–148.

[16] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Proc. Adv. Neural Inf. Process. Syst.*, Curran Assoc., 2017, pp. 2990–2999.

[17] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2018, pp. 2255–2264.

[18] J. Li, H. Ma, and M. Tomizuka, "Conditional generative neural system for probabilistic trajectory prediction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 6150–6156.

[19] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2016, pp. 961–971.

[20] C. Choi, J. H. Choi, J. Li, and S. Malla, "Shared cross-modal trajectory prediction for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*., 2021, pp. 244–253.

[21] J. Li, F. Yang, H. Ma, S. Malla, M. Tomizuka, and C. Choi, "RAIN: Reinforced hybrid attention inference network for motion forecasting," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021.

[22] H. Ma, J. Li, W. Zhan, and M. Tomizuka, "Wasserstein generative learning with kinematic constraints for probabilistic interactive driving behavior prediction," in *Proc. IEEE Intell. Veh. Symp.*, 2019, pp. 2477–2483.

[23] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "PRECOG: Prediction conditioned on goals in visual multi-agent settings," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2019, pp. 2821–2830.

[24] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "DESIRE: Distant future prediction in dynamic scenes with interacting agents," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2017, pp. 336–345.

[25] J. Li, H. Ma, and M. Tomizuka, "Interaction-aware multi-agent tracking and probabilistic behavior prediction via adversarial learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 6658–6664.

[26] C. Dong, J. M. Dolan, and B. Litkouhi, "Intention estimation for ramp merging control in autonomous driving," in *Proc. IEEE Intell. Veh. Symp.*, 2017, pp. 1584–1589.

[27] L. Sun, W. Zhan, D. Wang, and M. Tomizuka, "Interactive prediction for multiple, heterogeneous traffic participants with multi-agent hybrid dynamic Bayesian network," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 1025–1031.

[28] Y. Hoshen, "VAIN: Attentional multi-agent predictive modeling," in *Proc. Adv. Neural Inf. Process. Syst*., 2017, pp. 2701–2711.

[29] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, "Neural relational inference for interacting systems," in *Int. Conf. Machine Learn. PMLR*, 2018, pp. 2688–2697.

[30] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[31] X. Li, X. Ying, and M. C. Chuah, "GRIP: Graph-based interaction-aware trajectory prediction," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 3960–3966.

[32] H. Ma, Y. Sun, J. Li, and M. Tomizuka, "Multi-agent driving behavior prediction across different scenarios with self-supervised domain knowledge," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2021.

[33] M.-F. Chang *et al.*, "Argoverse: 3D tracking and forecasting with rich maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2019, pp. 8748–8757.

[34] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6467–6476.

[35] S. C. Y. Hung, C.-H. Tu, C.-E. Wu, C.-H. Chen, Y.-M. Chan, and C.-S. Chen, "Compacting, picking and growing for unforgetting continual learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 13669–13679.

[36] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with a-gem," *Proc. Int. Conf. Learn. Representations*, 2019.

[37] M. Rostami, S. Kolouri, and P. K. Pilly, "Complementary learning for overcoming catastrophic forgetting using experience replay," 2019. [Online]. Available: http://arxiv.org/abs/1903.04566

[38] W. D. Schacter DL and Gilbert DT"Semantic and episodic memory," *Psychology, Macmillan International Higher Education*, 2009, pp 185–6.

[39] J. Li, F. Yang, M. Tomizuka, and C. Choi, "Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 19783–19794.

[40] N. Díaz-Rodríguez, V. Lomonaco, D. Filliat, and D. Maltoni, "Don't forget, there is more than forgetting: New metrics for continual learning," 2018, *arXiv:1810.13166*.