Continual Interactive Behavior Learning With Traffic Divergence Measurement: A Dynamic Gradient Scenario Memory Approach

Yunlong Lin, Zirui Li, *Student Member, IEEE*, Cheng Gong, *Graduate Student Member, IEEE*, Chao Lu, *Member, IEEE*, Xinwei Wang, *Member, IEEE*, and Jianwei Gong, *Member, IEEE*

***Abstract*— Developing autonomous vehicles (AVs) helps improve the road safety and traffic efficiency of intelligent transportation systems (ITS). Accurately predicting the trajectories of traffic participants is essential to the decision-making and motion planning of AVs in interactive scenarios. Recently, learning-based trajectory predictors have shown state-of-the-art performance in highway or urban areas. However, most existing learning-based models trained with fixed datasets may perform poorly in continuously changing scenarios. Specifically, they may not perform well in learned scenarios after learning the new one. This phenomenon is called "catastrophic forgetting". Few studies investigate trajectory predictions in continuous scenarios, where catastrophic forgetting may happen. To handle this problem, first, a novel continual learning (CL) approach for vehicle trajectory prediction is proposed in this paper. Then, inspired by brain science, a dynamic memory mechanism is developed by utilizing the measurement of traffic divergence between scenarios, which balances the performance and training efficiency of the proposed CL approach. Finally, datasets collected from different locations are used to design continual training and testing methods in experiments. Experimental results show that the proposed approach achieves consistently high prediction accuracy in continuous scenarios without re-training, which mitigates catastrophic forgetting compared to non-CL approaches. The implementation of the proposed approach is publicly available at https://github.com/BIT-Jack/D-GSM.**

***Index Terms*— Continual learning, interactive behavior modeling, intelligent transportation systems, autonomous vehicles, trajectory prediction.**

NOMENCLATURE

Symbol	Definition
t_h	Observed time horizon.
t_f	Predicted time horizon.
$\mathbf{tr}$	Positions of vehicles.
$\hat{\mathbf{tr}}$	Predicted position of vehicles.
N_{ts}	Number of testing samples.
$\mathbf{X}$	Historical trajectories observed by predictor.
$\mathbf{Y}$	Future trajectories to be predicted.
Γ	Bivariate Gaussian distribution over the predicted time horizon.
$\mathbf{X}^{\mathbf{N}}_{\mathbf{cond}}$	Condition considered N surrounding vehicles.
$\mathrm{CKLD}^{\mathrm{wt}}_{i,j}$	Weighted-CKLD between distribution p_i and p_j.
m^{D}_r	Amounts of memory data allocated to the r^{th} task.
$f_{\boldsymbol{\theta}}$	Model parameterized by $\boldsymbol{\theta}$.
$f'_{\boldsymbol{\theta}}$	Model state at the end of learning of the last task.
$\mathbf{g_r}$	Loss gradient of the r^{th} previous task.
S	Continuous scenarios.
d_i	The i^{th} scenario in continuous scenarios.
c	Index of "the current scenario".

I. INTRODUCTION

AUTONOMOUS vehicles (AVs) play an essential role in improving traffic safety and efficiency in intelligent transportation systems (ITS) [1]. Since understanding interactive behavior of road users and predicting their future trajectories support the efficient decision-making and enable the risk assessment of AVs, they are fundamental to develop AVs [2], [3], [4].

In early studies, trajectories of vehicles are predicted based on kinetic and dynamic models [5], [6]. These approaches are classified into physics-based methods in [7]. Physics-based methods perform well in short-term (less than 1s) prediction. However, driving behaviors among vehicles influence each other, but these methods do not consider these interactive behaviors. Thus, physics-based methods cannot handle changes caused by the execution of a particular behavior (e.g., a lane-changing action, acceleration, or brake).

Manuscript received 20 June 2022; revised 19 December 2022 and 9 July 2023; accepted 28 September 2023. Date of publication 27 October 2023; date of current version 18 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 52372405 and Grant 61703041 and in part by the China Scholarship Council (CSC). The Associate Editor for this article was J. E. Naranjo. *(Yunlong Lin and Zirui Li contributed equally to this work.) (Corresponding authors: Chao Lu; Jianwei Gong.)*

Yunlong Lin, Cheng Gong, Chao Lu, and Jianwei Gong are with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China (e-mail: yunlonglin@bit.edu.cn; chenggong@bit.edu.cn; chaolu@bit.edu.cn; gongjianwei@bit.edu.cn).

Zirui Li is with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China, and also with the Chair of Traffic Process Automation, "Friedrich List" Faculty of Transport and Traffic Sciences, TU Dresden, 01069 Dresden, Germany (e-mail: z.li@bit.edu.cn).

Xinwei Wang is with the School of Engineering and Materials Science, Queen Mary University of London, E1 4NS London, U.K. (e-mail: xinwei.wang@qmul.ac.uk).

Digital Object Identifier 10.1109/TITS.2023.3324983

It limits their long-term prediction performance in interactive scenarios [7].

More advanced interaction-aware methods are developed to overcome these limitations. Most interaction-aware methods are learning-based, which predict future trajectories based on the modeling of interactions between agents [8]. By utilizing Long Short-Term Memory (LSTM), Social LSTM is the first work that modeled interactions between pedestrians as social behaviors [9]. In Social LSTM, a grid-based pooling mechanism was designed to represent the spatial information of pedestrians in a scene. Then social-pooling layers were used to capture interactions among pedestrians based on the spatial information. Reference [10] improved the performance of Social LSTM by applying convolutional layers to replace the fully connected layers.

Moreover, Social LSTM was applied in the generator of Generative Adversarial Networks (GAN) to predict socially plausible trajectories in Social GAN [11]. Due to the more explicit modeling of interactions, graph-based methods with the better interpretability are developed. Graph-based methods represent interactions between agents by the nodes and edges in graphs. In [12], interactions and trajectories were modeled as spatial-temporal graphs using a Graph Neural Network (GNN). Then trajectories of vehicles and pedestrians were predicted by a GNN-based multitask learning framework. Similarly, [13] extracts interactions into social behavior graphs. Then a graph convolutional neural network was applied to propagate social interactions in such graphs, outperforming prior works.

However, the studies above mainly focus on designing networks or architectures to improve the accuracy of predictions. Most existing models are trained and tested specifically for each dataset. Models may fail under this learning paradigm if testing data distribution differs from the training data distribution [14], [15]. Meanwhile, AVs are expected to drive through various scenarios continually in applications. Since factors including traffic rules, the density of traffic flow, and road geometry are divergent, interactive behaviors vary in different scenarios. Most models need to be re-trained on increasing datasets to guarantee the good performance in continuous scenarios [16]. With a heavy computing burden and data storage requirement, this scheme is not efficient and practical. This paper is inspired by Continual Learning (CL) [17], which trains models on new data streams without directly accessing old data to address these problems.

Given a potentially unlimited data stream, learning from a sequence of partial experiences where all data is not available at once is called Continual Learning [16]. In continual learning, the model is updated with a dataset of the current task without directly accessing old ones. As shown in Fig. 1, models in this work observe different scenarios sequentially. Only the data of currently observed scenario are fully available.The data of past scenarios can be stored with limited amounts by CL strategies. Under this assumption, trainable parameters of non-CL models are optimized to minimize the loss of the current scenario. Conversely, the training loss of past scenarios is ignored. Thus, non-CL models may have low prediction accuracy in past scenarios after learning the new one, which is called "catastrophic forgetting". In other words, catastrophic forgetting happens since models try to "fit" the current task without considering the performance on past ones. Most learning based approaches are data-driven. From the aspect of data, one of the reasons for catastrophic forgetting is that the distributions of training data and the testing data are different, where the datasets corresponding to different tasks are not independent and identical distributed (non-i.i.d.) [18]. One of the primary purposes of continual learning strategies is to alleviate catastrophic forgetting [17], [19], [20]. To overcome the catastrophic forgetting, this paper aims to enable learning-based trajectory predictors to have consistent good performance in sequential tasks without re-training. The task similarity is also exploited to balance the learning cost and the performance.

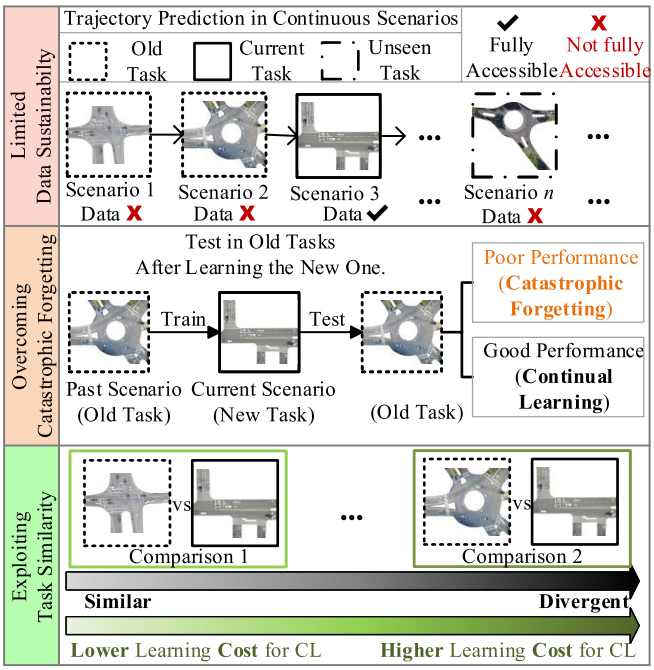


Fig. 1. Continuous scenarios are coming in a sequence. Due to the limited data sustainability, the data of past and unseen scenarios are not fully accessible. In continuous scenarios, trajectory predictor trained with the current scenario data may have poor performance in learned scenarios (old tasks), which is called catastrophic forgetting. The proposed CL approach aims to address this problem. Measurements of the traffic divergence (task similarity) are also exploited to reduce learning cost in this study.

More detailedly, this work is inspired by rehearsal methods, which are advanced strategies in CL [21], [22], [23]. Rehearsal methods achieve the goal of CL by using a memory system to keep the learned knowledge. The specific strategy used in this work is Gradient Episodic Memory (GEM) [23]. GEM firstly uses the data stored in the episodic memory to calculate losses on previous tasks. Then, these losses are utilized to define an inequality constraint, interfering with the training process. Under the constraint, models are required to avoid the increment of losses on previous tasks when updating the trainable parameters to mitigate the catastrophic forgetting.

In this research, a novel CL approach termed Dynamic Gradient Scenario Memory (D-GSM) is proposed for the prediction of vehicle trajectory in continuous scenarios. Compared with non-CL models, D-GSM improves the predicting performance in continuous scenarios by constraining the loss increment in observed scenarios. Moreover, inspired by brain

science [24], a dynamic memory is designed in D-GSM by exploiting the similarity of tasks. The similarity of tasks are modeled by traffic divergence between scenarios. Instead of treating all previous tasks equally as in [25] and [26], the dynamic memory allocates different memory resource to divergent tasks, balancing the performance and training efficiency of the CL strategy in D-GSM. Main contributions of this paper are:

1) A novel continual learning approach named Dynamic Gradient Scenario Memory (D-GSM) is proposed for interactive behavior learning in continuous scenarios. With the help of GEM, the proposed D-GSM enables learning-based prediction models to consistently perform well in various scenarios coming in a sequence without re-training by avoiding the increment of losses on previous tasks.
2) A new metric based on Kullback-Leibler divergence (KLD) is proposed to measure traffic divergence between different scenarios. The distance between distributions of different scenario data is measured by the KLD-based metric, representing the traffic divergence. Furthermore, a dynamic memory exploiting the measurement of traffic divergence is developed to improve the training efficiency of the proposed CL approach.
3) Evaluation methods for vehicle trajectory prediction in continuous scenarios are designed. Then, three experiments are conducted based on the divergent scenario data collected from different locations. It should be noted that the proposed CL approach is a plug-and-play approach. The base model adopted in experiments is demonstrated as an example.

II. Problem Formulation and Preliminaries

In this work, interaction-aware predictors are trained using a sequence of datasets collected in different scenarios. To formulate the trajectory prediction in continuous scenarios, the fundamental problem, i.e., the prediction of vehicle trajectory in the interactive scenario, will be firstly formulated in Section II-A. Then, based on the fundamental problem, the CL task for trajectory prediction in continuous scenarios will be described in Section II-B.

A. Trajectory Prediction in Interactive Scenarios

As most interactive behavior learning research [12], [27], [28], this work considers scenarios with highly interactive behaviors among vehicles, including scenarios in urban and highways. The scenarios in this work are represented by datasets collected from divergent locations at different times. The scenario is defined as a traffic region with specific road geometry and interactive traffic participants. It is assumed that datasets representing different scenarios can be non-i.i.d. Both intuitive comparisons and rigorous statistics will be provided in Section IV-A to show this property. This paper also proposes a method in Section III-B to quantitatively measure the divergence between distributions of motion data from different scenarios. The following of this subsection will generally formulate the trajectory prediction task in the interactive scenario.

In an interactive scenario, driving behaviors and motions of vehicles influence each other. The interaction-aware trajectory predictor observes historical trajectories of vehicles in the scene over t_h seconds. Then, it predicts future trajectories of target vehicles over t_f seconds. In detail, the input to the predictor is:

$$\mathbf{X} = \left[\mathbf{tr}^{(t-t_h)}, \ldots, \mathbf{tr}^{(t-1)}, \mathbf{tr}^{(t)}\right]. \tag{1}$$

In (1), $\mathbf{tr}^{(t)} = \left[x_0^{(t)}, y_0^{(t)}, \ldots, x_i^{(t)}, y_i^{(t)}, \ldots, x_n^{(t)}, y_n^{(t)}\right]$ represents x and y co-ordinates of vehicles at time t. $\mathbf{tr}^{(t)}$ includes $x_0^{(t)}$ and $y_0^{(t)}$, representing co-ordinates of the target vehicle (predicted vehicle). $x_i^{(t)}$ and $y_i^{(t)}$, $(i = 1, \ldots, n)$ are co-ordinates of surrounding vehicles. Since decision-making expects the prediction to provide all possible future motions [4], the output is not a single predicted trajectory of target vehicle. Instead, the output is considered as the estimated bi-variate distribution over:

$$\mathbf{Y} = \left[\mathbf{tr}_0^{(t+1)}, \ldots, \mathbf{tr}_0^{(t+t_f)}\right], \tag{2}$$

with $\mathbf{tr}_0^{(t)} = \left[x_0^{(t)}, y_0^{(t)}\right]$ are future co-ordinates of the target vehicle. Thus, the output can be formulated as:

$$\mathrm{P}(\mathbf{Y}|\mathbf{X}) \sim \Gamma. \tag{3}$$

where $\Gamma = \left[\Gamma^{(t+1)}, \ldots, \Gamma^{(t+t_f)}\right]$ are parameters of a bivariate Gaussian distribution at each time step over the prediction horizon. After introducing the fundamental task, the CL task for trajectory prediction in continuous scenarios is described in Section II-B.

B. Continual Learning Task for Trajectory Prediction

CL problem consists of a sequence of tasks coming in a stream [16], [19]. In this work, tasks refer to trajectory predictions in interactive scenarios, which is formulated in Section II-A. Due to the limited data and computing resource, it is assumed that all data are not available at once. Moreover, the observed data cannot be entirely stored. Under these assumptions, the trajectory predictor is expected to perform well consistently in sequential tasks, i.e., predictions in continuous scenarios. Important concepts of the CL tasks in this work are defined as follow:

1) Continuous Scenarios: Continuous scenarios consist of a sequence of datasets which can be characterized as $S = \{d_1, \ldots, d_c, \ldots d_n\}$. Samples in each dataset d_i are collected in the same location, which are assumed corresponding to an unknown distribution Ψ_i, while distinct datasets are collected in different location at different time. The scenario is represented by d_i. The AV is assumed to pass these scenarios orderly, and $d_i \in S, (i = 1, \ldots, n)$ is the i^{th} scenario to be observed by the AV. It should be noted that scenarios are allowed to appear in S more than one times, corresponding to the situation that the AV passes the same scenario again.

2) Current Scenario: The current scenario refers to the newly coming scenario d_c in S. The data of the current scenario are fully available to train the trajectory predictor. The trajectory prediction in the current scenario corresponds to the current task in CL.

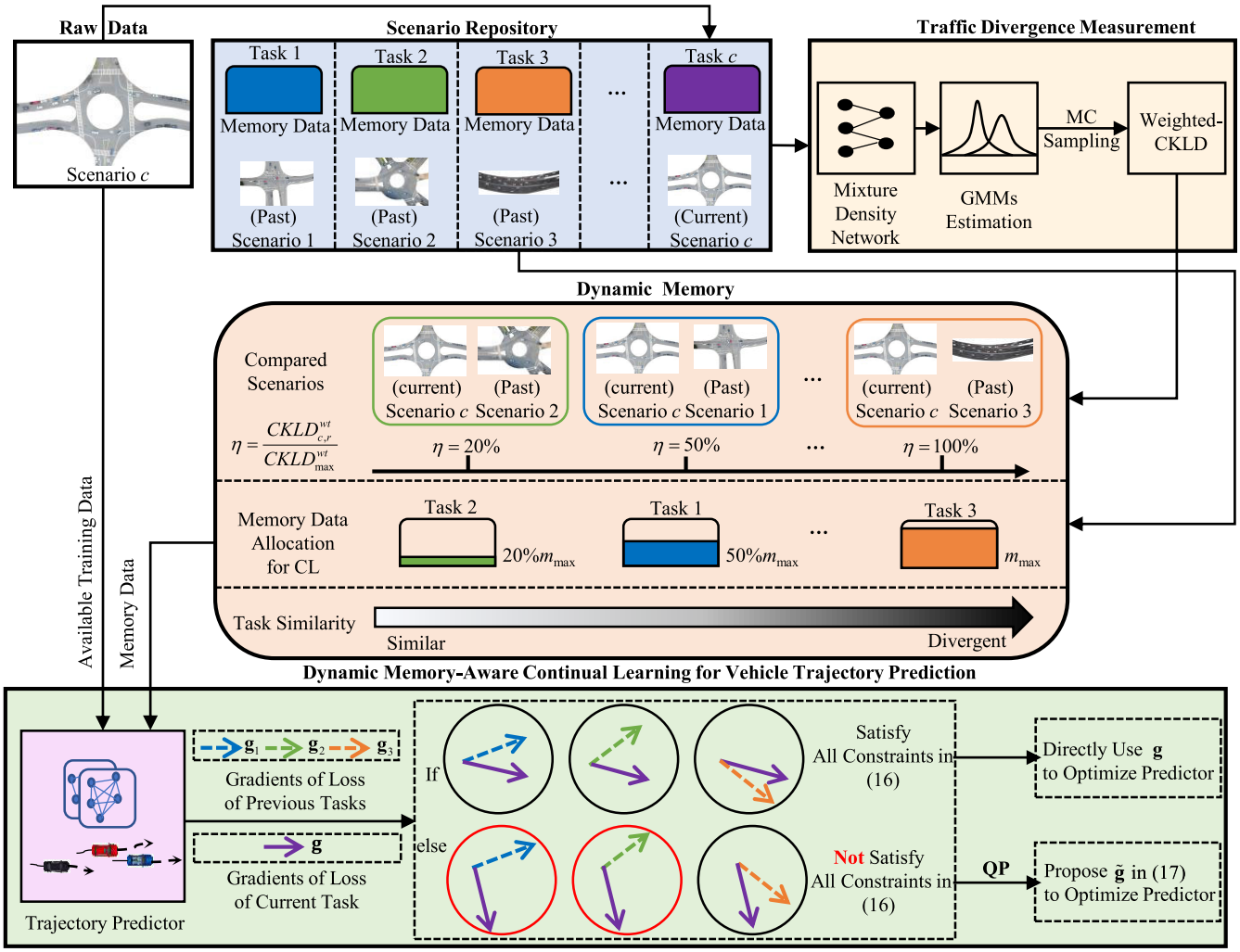


Fig. 2. Dynamic Gradient Scenario Memory (D-GSM): When a new traffic scenario arrives, scenario repository will firstly store new data as memory data. Then, the traffic divergence measurement module utilizes memory data to measure the divergence between the current scenario and each past scenario. Based on the measuring of divergence, the dynamic memory module allocates memory data with different amounts for different previous tasks to the CL module. Finally, with the help of GEM strategy, the trajectory predictor is trained in the CL module. Since the more memory data bring the higher computing cost, the dynamic memory balances the training efficiency and performance by allocating memory data in a reasonable way.

3) Past Scenarios: In continuous scenarios S, the scenarios $d_i \in S, (i = 1, .., c-1)$ that have been observed are termed past scenarios. The data of past scenarios cannot be directly used for training, while the CL strategy can store these data with limited amounts. The trajectory predictions in past scenarios refer to previous tasks in CL.

4) Future Scenarios: The scenarios $d_j \in S, (j = c+1, .., n)$ to be arrived are termed future scenarios. Data of future scenario are not available. The next "current scenario" will come from future scenarios. The number of scenarios n can be unknown.

5) Catastrophic Forgetting: Catastrophic forgetting in this work refers to the phenomenon that the prediction accuracy in past scenarios declines after the predictor learns the data of current scenario.

After learning the data of current scenario, a trajectory predictor is expected to perform well among all scenarios that have been learned. For example, in continuous scenarios $S = \{d_1, \ldots, d_c, \ldots, d_n\}$, the data of the current scenario $d_c \in S$ are fully accessible for training. Conversely, the data of past scenarios $d_i \in S, (i = 1, .., c-1)$ are not fully accessible. The performance of prediction is evaluated on all testing sets from $d_i \in S, (i = 1, .., c)$. Denoting the prediction error in the i^{th} scenario as R_i, the aim of CL task for trajectory prediction in continuous scenarios is to minimize the average errors R of all scenarios that have been learned in S:

$$\text{minimize } R = \frac{1}{c}\sum_{i=1}^{c} R_i. \tag{4}$$

III. Dynamic Gradient Scenario Memory for Vehicle Trajectory Prediction

This section introduces the proposed CL approach for vehicle trajectory prediction in continuous scenarios, termed Dynamic Gradient Scenario Memory (D-GSM). As shown in Fig. 2, D-GSM consists of four modules: 1) a scenario repository, 2) a traffic divergence measuring module,

3) a dynamic memory module, and 4) a dynamic memory-aware continual learning module.

Since the old data is not directly accessible, the scenario repository is used to store the observed data of past scenarios as memory data. When a new scenario comes, the scenario repository updates by storing new data with abandoning a part of old memory data due to the limited storage. Then, the traffic divergence between the current and past scenarios is measured using the memory data from the scenario repository. Next, according to measuring results, the dynamic memory module dynamically allocates memory data into the CL module. Finally, with the help of allocated memory data, the CL module trains the trajectory predictor using GEM strategy. The details of four modules in D-GSM are presented in Section III-A, Section III-B, Section III-C, and Section III-D, respectively.

A. Scenario Repository

The scenario repository is used to store observed data with a limited amount as memory data. In $S = \{d_1, \ldots, d_c, \ldots, d_n\}$, when a new scenario d_c arrives, the scenario repository updates by storing the data from the current scenario d_c. Since the storage space of the repository is not infinite, it also abandons a part of memory data stored from $d_i (i = 1, \ldots, c-1)$. Supposing the upper limit of data storage is M, amount of memory data corresponding to the i^{th} scenario will be $m_i = M/c$. Thus, when a new scenario arrives, m_i declines while c increases.

In D-GSM, the memory data stored in the scenario repository are used to measure the traffic divergence between scenarios. Then, according to the measurement, the dynamic memory module allocates memory data into continual learning modules for model training. Different tasks are allocated with different amounts of memory data to improve the training efficiency.

B. Traffic Divergence Measuring

The divergence between different scenarios is the main reason causing catastrophic forgetting of trajectory prediction in continuous scenarios. On the contrary, if the new scenario is quite similar with the observed one, the catastrophic forgetting may not happen. Moreover, exploiting similarity of tasks is also a key factor in CL [24]. Thus, the divergence (or described as similarity) measurement between traffic scenarios is considered in this work. In detail, traffic divergence measuring module in D-GSM uses a KLD-based metric to quantify the traffic divergence between scenarios. The measurement estimates the similarity between CL tasks, enabling the strategy of data allocation in dynamic memory module.

According to previous studies [26], [29], the divergence between interactive scenarios can be represented by the difference of spatiotemporal dependency among vehicles. From the aspect of scenario data, the spatiotemporal dependency can be formulated as the conditional probability density function (CPDF) $p(\mathbf{Y}|\mathbf{X})$, where $\mathbf{Y}$ represents the future trajectories of the predicted vehicle, and $\mathbf{X}$ represents the observed historical trajectories of all vehicles.

Supposing p_1 and p_2 are the CPDFs of two scenarios. The distance between CPDFs are calculated by conditional Kullback-Leibler divergence (CKLD):

$$\begin{aligned}&\mathrm{CKLD}\left(p_1(\mathbf{Y}|\mathbf{X})\,||\,p_2(\mathbf{Y}|\mathbf{X})\right)\\&\quad=\int p_1(\mathbf{X})\int\log\left(\frac{p_1(\mathbf{Y}|\mathbf{X})}{p_2(\mathbf{Y}|\mathbf{X})}\right)p_1(\mathbf{Y}|\mathbf{X})\,d\mathbf{Y}d\mathbf{X},\end{aligned}\tag{5}$$

In (5), p_1 and p_2 are approximated by estimations of Gaussian Mixture Models (GMMs) from a Mixture Density Networks (MDN) [30]. It should be noted that, in the implementation, to facilitate the learning process, the condition $\mathbf{X}$ with fixed dimensions is used, which only considers the closest N surrounding vehicles. Besides, to represent the interactions of vehicles, instead of directly using historical trajectories, k eigenvectors of a 2D Laplacian matrix concatenated with historical trajectories of the predicted vehicle $\mathbf{X}_0 = \left[\mathbf{tr}_0^{(t-t_h)}, \ldots, \mathbf{tr}_0^{(t-1)}\right]$ are used to generate the condition $\mathbf{X}$. To distinguish this processed condition $\mathbf{X}$ from the historical trajectories described in (1), we denote this condition one as $\mathbf{X}_{cond}^N$:

$$\mathbf{X}_{cond}^N = [\mathbf{X}_0, \mathbf{v}_1, \ldots, \mathbf{v}_k],\tag{6}$$

where $\mathbf{v}_i (i = 1, \ldots, k)$ are eigenvectors of a 2D Laplacian matrix.

Supposing that $e\left(\mathbf{tr}_i^k, \mathbf{tr}_j^k\right)$ is Euclidean distance between the i^{th} vehicle and j^{th} vehicle at time k, and λ is a decay parameter, the Laplacian matrix $\mathbf{L_p}$ is computed by:

$$\mathbf{L_p} = \mathbf{D} - \mathbf{A}\tag{7}$$

where $\mathbf{A} = \left(a_{i,j}\right)_{N\times N}$ and $\mathbf{D} = \left(d_{i,j}\right)_{N\times N}$. The elements in matrix $\mathbf{A}$ and $\mathbf{D}$ are calculated by:

$$\begin{aligned}a_{i,j} &= \exp\left(-\sum_{k=t-t_h}^{t-1}\omega_k e\left(\mathbf{tr}_i^k, \mathbf{tr}_j^k\right) \Big/ \sum_{k=t-t_h}^{t-1}\omega_k\right),\\ \omega_k &= \lambda^{(t-1)-k}, k = t-t_h, \ldots, t-1,\\ d_{i,j} &= \begin{cases}\sum_{j=1}^{N} a_{i,j}, & i = j\\ 0, & i \neq j,\end{cases}\end{aligned}\tag{8}$$

After estimating the GMM for each condition $\mathbf{X}_{cond}^N$, Monte-Carlo sampling is used to compute the Kullback-Leibler divergence (KLD) since the KLD between two GMMs is not analytically attractable. Then CKLD is obtained based on KLD. Specifically, assuming that distribution $p_1(\mathbf{X}_{cond}^N)$ has n_1 samples denoted as $\mathbf{X}_{cond,i}^N (i = 1, \ldots, n_1)$, n_{mc} samples denoted as $\mathbf{Y}_j (j = 1, \ldots, n_{mc})$ are sampled from $p_1(\mathbf{Y}|\mathbf{X}_{cond,i}^N)$, KLD between two distributions is obtained by Monte-Carlo sampling:

$$\begin{aligned}&\mathrm{KLD}\left(p_1\left(\mathbf{Y}|\mathbf{X}_{cond,i}^N\right)||\,p_2\left(\mathbf{Y}|\mathbf{X}_{cond,i}^N\right)\right)\\&\quad=\frac{1}{n_{mc}}\sum_{j=1}^{n_{mc}}\left(\log p_1\left(\mathbf{Y}_j|\mathbf{X}_{cond,i}^N\right)-\log p_2\left(\mathbf{Y}_j|\mathbf{X}_{cond,i}^N\right)\right).\end{aligned}\tag{9}$$

Finally, CKLD is calculated as:

$$\begin{aligned}&\mathrm{CKLD}\left(p_1\left(\mathbf{Y}|\mathbf{X}_{cond}^{N}\right)||p_2\left(\mathbf{Y}|\mathbf{X}_{cond}^{N}\right)\right)\\&=\frac{1}{n_1}\sum_{i=1}^{n_1}\mathrm{KLD}\left(p_1\left(\mathbf{Y}|\mathbf{X}_{cond,i}^{N}\right)||p_2\left(\mathbf{Y}|\mathbf{X}_{cond,i}^{N}\right)\right).\end{aligned}\quad(10)$$

Moreover, since the original KLD is calculated using memory data in the scenario repository with a storage limit, the amount of memory data for each scenario decreases when a new scenario arrives, as stated in Section III-A. The influence of data usage on CKLD is also considered in this paper. More investigations and the basic requirement of this calculation are represented in Appendix D.

CKLD can preliminarily reveal the divergence between scenarios since large CKLD indicates a significant distance between GMMs. If two scenarios are the same, CKLD will equal zero. However, CKLD is asymmetrical, which means that if the p_1 and p_2 are exchanged in (5), CKLD will be different. The asymmetry may lead to contradictory measuring results when comparing divergence among traffic scenarios. Therefore, this work proposes a novel metric termed weighted-CKLD to address the problem mentioned above. The weighted-CKLD is calculated as:

$$\mathrm{CKLD}_{1,2}^{wt}=w_1\mathrm{CKLD}\left(p_1||p_2\right)+w_2\mathrm{CKLD}\left(p_2||p_1\right)\quad(11)$$

In (11), w_1 and w_2 are weights ($w_1+w_2=1$). And p_1 and p_2 in (11) represent CPDFs of scenarios to compare. Which weight needs to be assigned a larger value depends on which distribution we pay more attention to. For example, there are three CPDFs $\{p_1, p_2, p_3\}$ corresponding to three scenarios. We want to compare the divergence between $\{p_1, p_2\}$ and $\{p_1, p_3\}$. The weighted-CKLD $\mathrm{CKLD}_{1,2}^{\mathrm{wt}}$ and $\mathrm{CKLD}_{1,3}^{\mathrm{wt}}$ need to be calculated. In this case, the highlighted one is p_1. Then we choose larger values for w_1 in $\mathrm{CKLD}_{1,2}^{\mathrm{wt}}$ and $\mathrm{CKLD}_{1,3}^{\mathrm{wt}}$.

In our work, D-GSM explores the relationship between catastrophic forgetting and the spatiotemporal dependency divergence between current traffic scenarios and past scenarios. Thus the highlighted scenario is the current one. Weighted-CKLD measuring the traffic divergence is utilized to improve the training efficiency of CL module by applying the dynamic memory, which is introduced in Section III-C.

C. Dynamic Memory Module

The dynamic memory module allocating different memory data for divergent tasks to improve the efficiency of CL is inspired by the synaptic theory of memory in brain science [31]. The synaptic theory indicates that the forgetting process can become rapid when memory resources are restricted [24], [32]. The idea inspired by brain science and the allocating principle in the dynamic memory module are shown in Fig. 3. In this work, the memory resources refer to the memory data. Using more memory data may enable the model to have better retention. However, more memory data also burden the training cost in CL.

Based on the inspiration, an allocation strategy for memory data is developed in the dynamic memory module to make a trade-off between the performance and training cost: Intuitively, compared with tasks in pretty different scenarios, tasks in past scenarios that are similar to the current one are "easier" for models to remember since the learning-based model updates parameters depending on the training data. Therefore, to balance the effectiveness and efficiency of CL, the dynamic memory in D-GSM allocates more memory data for observed tasks that are "hard" to remember.

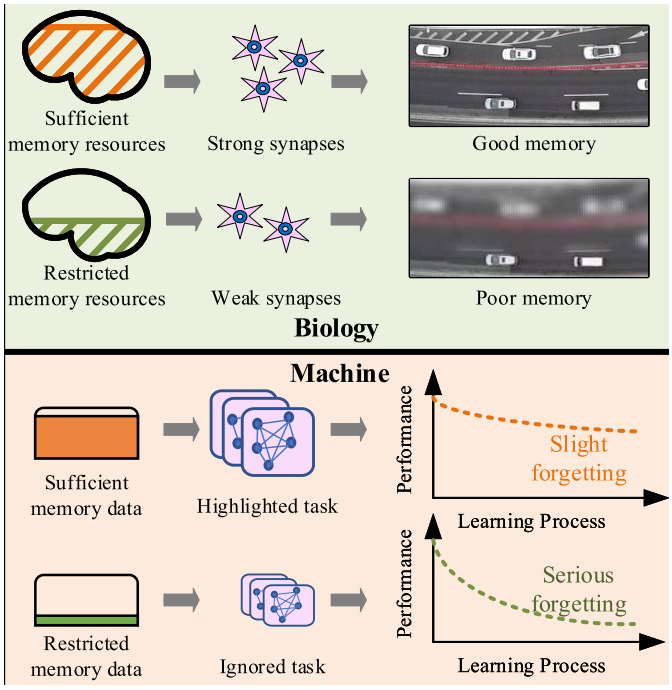


Fig. 3. Dynamic memory in D-GSM is inspired by synaptic theory in brain science. Sufficient memory resources help agents to remember specific issues/tasks. Restricted memory resources bring rapid forgetting.

In detail, results of traffic divergence measuring is used to enable the dynamic memory module to dynamically allocate memory data from the scenario repository to the CL module. Instead of allocating memory data to all previous tasks equally as GEM [23], the dynamic memory of D-GSM allocates a different number of memory data to different previous tasks. The specific amounts of allocated data to a past scenario depend on its traffic divergence to the current scenario. Firstly, the traffic divergence measuring module uses memory data stored in the scenario repository to calculate weighted-CKLD. Then the maximum weighted-CKLD is selected:

$$\mathrm{CKLD}_{\max}^{\mathrm{wt}}=\max\left\{\mathrm{CKLD}_{\mathrm{c,r}}^{\mathrm{wt}}|\mathrm{r}\in\{1,\ldots,\mathrm{c}-1\}\right\}\quad(12)$$

where $\mathrm{CKLD}_{\mathrm{c,r}}^{\mathrm{wt}}$ is the weighted-CKLD between the current scenario and the r^{th} past scenario. The traffic scenario corresponding to the maximum weighted-CKLD indicates the largest divergence with the current scenario. Denote the maximum amounts of memory data for CL as $M_{cl}(M_{cl}\leq M)$. Then, memory data with the most amounts $m_{\max}=\frac{M_{cl}}{c-1}$ is allocated to this traffic scenario. Finally, memory data amounts allocated to the r^{th} scenario is calculated by:

$$m_r^{\mathrm{D}}=m_{\max}\frac{\mathrm{CKLD}_{\mathrm{c,r}}^{\mathrm{wt}}}{\mathrm{CKLD}_{\max}^{\mathrm{wt}}}.\quad(13)$$

Equation (13) presents the specific data allocation of dynamic memory, which will be used to formulate the dynamic memory-aware CL module of D-GSM in Section III-D.

D. Dynamic Memory-Aware Continual Learning

The allocated memory data from the dynamic memory module are used to apply the CL strategy in model training. Inspired by GEM [23], the CL module in D-GSM firstly defines loss functions for previous tasks, i.e., trajectory predictions in past scenarios. Then, inequality constraints are set to interface the training process, where the model observes the training data of the current scenario. Finally, with the help of the Quadratic Program (QP) algorithm, proposed gradients satisfying the inequality constraints are applied to update parameters, which avoids the increment of losses on previous tasks.

The loss functions for previous tasks are calculated using memory data. Original GEM treats all previous tasks equally without considering the similarity between tasks, which may bring an unnecessary computational burden. As introduced in Section III-C, instead of treating all tasks equally, D-GSM uses dynamic memory data allocation to construct loss functions for previous tasks:

$$l\left(f_{\boldsymbol{\theta}}, m_r^{\mathrm{D}}\right) = \frac{1}{m_r^{\mathrm{D}}} \sum_{i=1}^{m_r^{\mathrm{D}}} l\left(f_{\boldsymbol{\theta}}\left(\mathbf{X}_i, r\right), \mathbf{Y}_i\right) \tag{14}$$

where $m_r^{\mathrm{D}} (r = 1, .., c-1)$ is the allocated amounts of samples described in (13). $f_{\boldsymbol{\theta}}$ is the vehicle trajectory prediction model parameterized by $\boldsymbol{\theta}$, and $(\mathbf{X}_i, r, \mathbf{Y}_i)$ is the i^{th} sample in the allocated memory data corresponding to the r^{th} $(r = 1, .., c-1)$ past scenario. Then, in the training process of the current task, losses in (14) are used to define inequality constraints to avoid the increment of losses on previous tasks.

Supposing that $(\mathbf{X}, c, \mathbf{Y})$ are samples of the current task, and $f'_{\boldsymbol{\theta}}$ represents the prediction model state at the end of learning of the last traffic scenario, the inequality constraints are formulated as:

$$\begin{aligned} &\text{minimize}_{\boldsymbol{\theta}}\ l\left(f_{\boldsymbol{\theta}}\left(\mathbf{X}, c\right), \mathbf{Y}\right) \\ &\text{s.t.}\quad l\left(f_{\boldsymbol{\theta}}, m_r^{\mathrm{D}}\right) \leq l\left(f'_{\boldsymbol{\theta}}, m_r^{\mathrm{D}}\right), \text{for all } r < c. \end{aligned} \tag{15}$$

For an efficient implementation, this paper assumes that the loss function is local linear (when learning rate is small), and denotes loss gradients of the current and previous tasks as $\mathbf{g}$ and $\mathbf{g}_r$, respectively. (15) can be rephrased into:

$$\begin{aligned} \langle \mathbf{g}, \mathbf{g}_r \rangle := \left\langle \frac{\partial l\left(f_{\boldsymbol{\theta}}\left(\mathbf{X}, c\right), \mathbf{Y}\right)}{\partial \boldsymbol{\theta}}, \frac{\partial l\left(f_{\boldsymbol{\theta}}, m_r^{\mathrm{D}}\right)}{\partial \boldsymbol{\theta}} \right\rangle \geq 0, \\ \text{for all } r < c. \end{aligned} \tag{16}$$

If constraints (16) are satisfied, the proposed gradient $\mathbf{g}$ to update parameters will not increase the loss of previous tasks. Otherwise, the gradient $\mathbf{g}$ will be projected to the closest gradient $\tilde{\mathbf{g}}$ (in squared L2 norm) satisfying all constraints in (16):

$$\begin{aligned} &\text{minimize}_{\tilde{\mathbf{g}}}\quad \frac{1}{2}\left\|\mathbf{g} - \tilde{\mathbf{g}}\right\|_2^2 \\ &s.t.\quad \left\langle \tilde{\mathbf{g}}, \mathbf{g}_r \right\rangle \geq 0, \text{for all } r < c. \end{aligned} \tag{17}$$

To solve (17) efficiently, a QP algorithm is used. The primal of a QP with inequality constraints is described as:

$$\begin{aligned} &\text{minimize}_z\quad \frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{C}\mathbf{z} + \mathbf{p}^{\mathrm{T}}\mathbf{z} \\ &\text{s.t.}\quad \mathbf{Q}\mathbf{z} \geq \mathbf{b} \end{aligned} \tag{18}$$

where $\mathbf{C} \in \mathbb{R}^{p \times p}$, $\mathbf{p} \in \mathbb{R}^p$, $\mathbf{Q} \in \mathbb{R}^{(c-1) \times p}$ and $\mathbf{b} \in \mathbb{R}^{c-1}$. The dual problem of (18) is:

$$\begin{aligned} &\text{minimize}_{u,v}\quad \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{C}\mathbf{u} - \mathbf{b}^{\mathrm{T}}\mathbf{v} \\ &\text{s.t.}\quad \mathbf{Q}^{\mathrm{T}}\mathbf{v} - \mathbf{C}\mathbf{u} = \mathbf{p}, \\ &\qquad \mathbf{v} \geq 0. \end{aligned} \tag{19}$$

If $(\mathbf{u}^*, \mathbf{v}^*)$ is a solution to (19), then there will be a solution $\mathbf{z}^*$ satisfying $\mathbf{C}\mathbf{z}^* = \mathbf{C}\mathbf{u}^*$. Thus, the primal GEM QP to (16) can be described as:

$$\begin{aligned} &\text{minimize}_{\mathbf{z}}\quad \frac{1}{2}\mathbf{z}^T\mathbf{z} - \mathbf{g}^T\mathbf{z} + \frac{1}{2}\mathbf{g}^T\mathbf{g} \\ &\text{s.t.}\quad \mathbf{G}\mathbf{z} \geq 0, \end{aligned} \tag{20}$$

where $\mathbf{G} = -\left[\mathbf{g}_1, \ldots, \mathbf{g}_{c-1}\right]$ and the constant term $\mathbf{g}^T\mathbf{g}$ is discarded. (20) is a QP on p variables (the number of trainable parameters of trajectory predictor). The dual of it is formulated as:

$$\begin{aligned} &\text{minimize}_{\mathbf{v}}\quad \frac{1}{2}\mathbf{v}^{\mathrm{T}}\mathbf{G}\mathbf{G}^{\mathrm{T}}\mathbf{v} + \mathbf{g}^{\mathrm{T}}\mathbf{G}^{\mathrm{T}}\mathbf{v} \\ &\text{s.t.}\quad \mathbf{G}\mathbf{z} \geq 0. \end{aligned} \tag{21}$$

Once the dual problem (21) is solved for $\mathbf{v}^*$, the projected gradient update can be recovered as $\tilde{\mathbf{g}} = \mathbf{G}^{\mathrm{T}}\mathbf{v}^* + \mathbf{g}$. According to [23], adding a small constant $\gamma \geq 0$ to $\mathbf{v}^*$ biased the gradient projection to updates that favored beneficial backwards transfer in practice. Fig. 2 shows the complete process of D-GSM. Fig. 4 also shows the mechanism of D-GSM to realize continual learning. Compared with not-CL methods, D-GSM applies the memory-aware continual learning strategy to avoid the loss on previous tasks during the current updating.

The entire algorithm of D-GSM is summarized in Algorithm 1. To clearly explain the input and output in each step, symbols in including f_{repo} for repository updating, f_{div} for divergence measuring, f_{alloc} for dynamic memory module, f_{cl} for CL training strategy and l_{prev}, l_{cur} for previous and current losses are newly used in Algorithm 1.

IV. Experiments

To investigate CL problems formulated in Section II and evaluate the proposed approach introduced in Section III, three experiments using datasets representing different scenarios are conducted. As shown in Fig. 5, the first experiment explores the relationship of catastrophic forgetting and task similarity of trajectory prediction in continuous scenarios. The second experiment investigates the influence of memory resource on the CL performance. The comparison between non-CL models and models applied with the proposed CL approach is demonstrated in the third experiment, evaluating the performance of D-GSM in continuous scenarios. This section firstly introduces used datasets and experimental settings. Then, experimental results with analysis are presented.

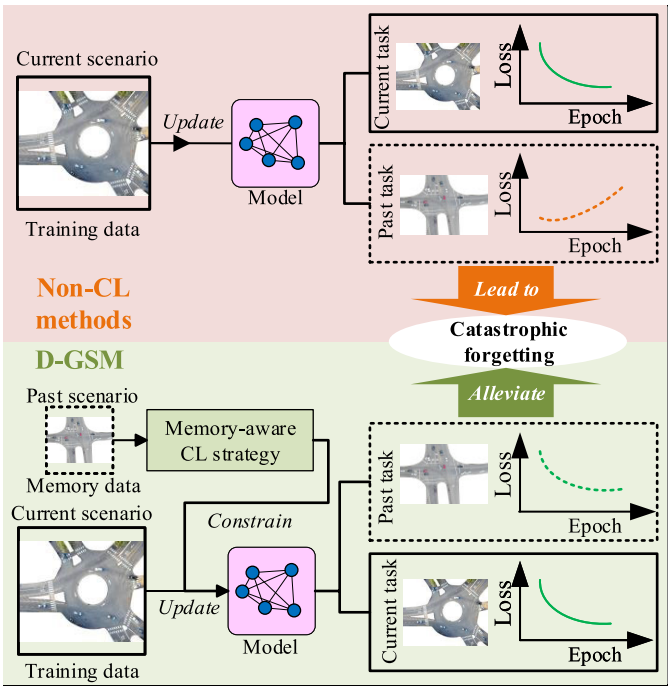


Fig. 4. The mechanism of D-GSM to realize continual learning in continuous scenarios. Compared with non-CL methods, dynamic memory-aware continual learning strategy of D-GSM constrains the increment of loss on previous tasks, which alleviates catastrophic forgetting.

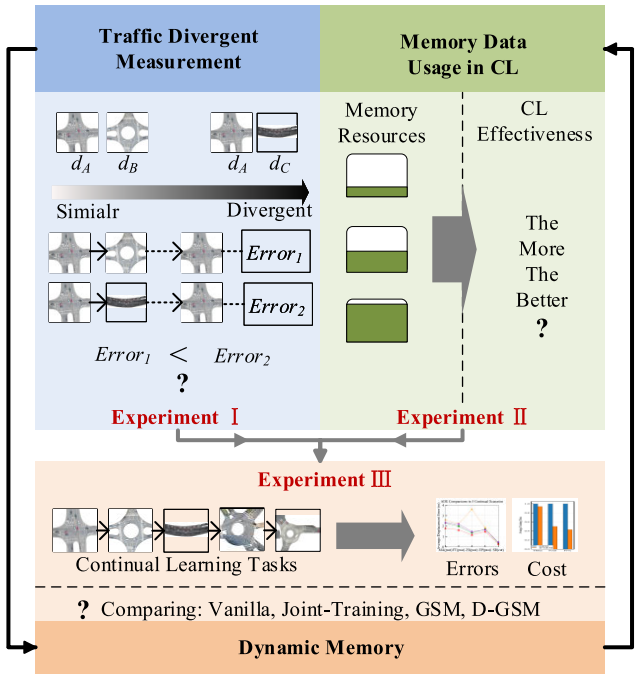


Fig. 5. The logic of three experiments in this paper. The experiment I explores the relationship between catastrophic forgetting and traffic divergence. After exploring the reasons for forgetting, the experiment II investigates the memory capability of the proposed CL approach using different amounts of memory data. The proposed D-GSM utilizes measurements of traffic divergence to dynamically allocate memory data. Experiment III compares the performance of D-GSM with several baselines.

A. Datasets and Implementations

Since this study focuses on CL tasks for interactive behavior learning, typical interactive scenarios, including merging scenarios in highways, intersections, and roundabout scenarios in urban areas, are selected as examples to evaluate the proposed approach. Specifically, experiments are conducted based on INTERACTION dataset [33]. INTERACTION is a widely used natural driving dataset that collects real traffic data from different locations to represent divergent scenarios [33], [34]. For convenience, these scenarios are denoted with different notations, as shown in Fig. 6.

These scenarios are divergent from many aspects such as road geometry, traffic participants, traffic rules, driving behaviors of vehicles, etc. Fig.7 also demonstrates the comparison of the distribution of the minimum time-to-conflict-point($\triangle TTCP_{min}$). $\triangle TTCP_{min}$ represents the density of interactive behaviors among these scenarios [33]. As shown in Fig. 7, it can be found that selected five scenarios have different density of interactive behaviors. More details on the definition of $\triangle TTCP_{min}$ are provided in Appendix C. Besides, Fig.8 shows the comparison of joint distributions of vehicle velocity and acceleration between different scenarios. These qualitative and statistical descriptions of the data used in experiment demonstrate the divergent between scenarios. According to these evidences, it is assumed that the datasets can be non-i.i.d. Furthermore, to rigorously quantify the divergence of scenarios, as described in Section III-B, this paper mainly focuses on the different spatiotemporal dependencies among vehicles between scenarios. And the weighted-CKLD is proposed to measure the divergence of distribution in the aspect of spatiotemporal dependencies between these non-i.i.d datasets which represent divergent scenarios.

In data processing, extracted features include IDs and coordinates of vehicles with timestamps. Then, trajectories samples

Algorithm 1 Dynamic Gradient Scenario Memory (D-GSM) for Vehicle Trajectory Prediction in Continuous Scenarios

Inputs: Data of continuous scenarios $S = \{d_1, \ldots, d_c, \ldots, d_n\}$; memory limit M, M_{cl}; functions f_{repo}, f_{div}, f_{alloc}, f_{cl}; loss l_{prev}, l_{cur}; model f_{θ}.

Outputs: Prediction of the r^{th} scenario $\mathrm{P}_r(\mathbf{Y}|\mathbf{X}) \sim \Gamma$ in (3), for r=1:c.

for current scenario $c = 1$ to n **do**
  for $i = 1$ to c **do**
    Memory data $d_i' \leftarrow f_{repo}(d_i, m_i)$. ▷ Scenario repository updating. Amount $m_i = M/c$.
    if $c > 1$ **then**
      for $r = 1$ to $c - 1$ **do**
        $\mathrm{CKLD}^{\mathrm{wt}}_{\mathrm{c,r}} \leftarrow f_{div}(d_c', d_r')$ ▷ Traffic divergence measuring in (6)-(11).
        Calculate $\mathrm{CKLD}^{\mathrm{wt}}_{\mathrm{max}}$ as (12).
        $m_r^{\mathrm{D}} \leftarrow f_{alloc}(\mathrm{CKLD}^{\mathrm{wt}}_{\mathrm{max}}, \mathrm{M_{cl}}, \mathrm{CKLD}^{\mathrm{wt}}_{\mathrm{c,r}})$. ▷ Dynamic memory allocating as (13).
        $l_{prev} \leftarrow l(f_{\theta}, m_r^{\mathrm{D}})$ ▷ Previous loss in (14).
        $\boldsymbol{\theta} \leftarrow f_{cl}(l_{prev}, l_{cur})$ ▷ CL strategy in (15)-(21).
      end for
    else
      Model training in a single scenario (c==1).
    end if
    Evaluate f_{θ} with the testing set of d_i.
  end for
end for

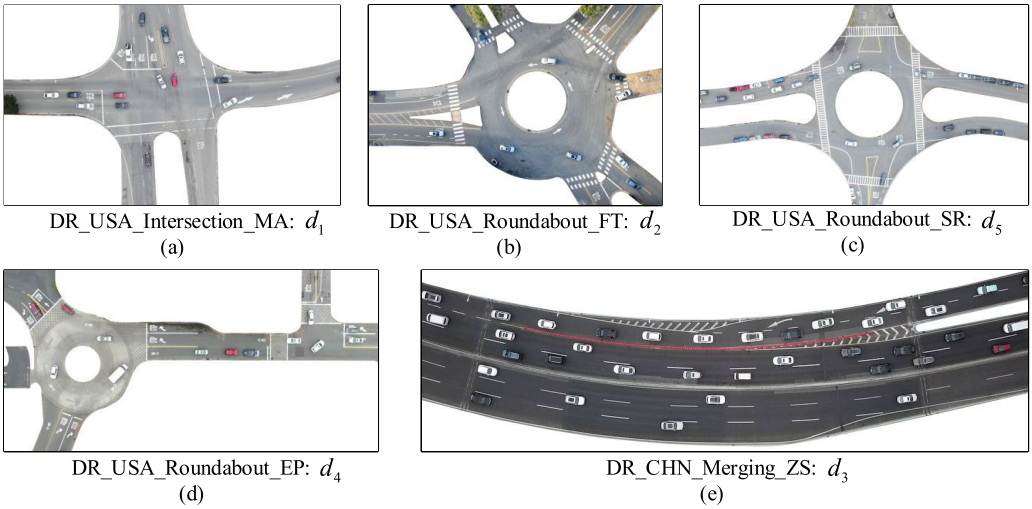


Fig. 6. Intuitive visual aids for scenarios: Various interactive driving scenarios are used in experiments based on INTERACTION datasets. The first two letters ("DR") of names represent that the data are collected by drones. The following three letters represent the corresponding country ("USA" is for the U.S.A., and "CHN" is for China). The last two letters are scenario codes to distinguish different locations [33]. d_1 is a busy all-way-stop intersection with lanes controlled by stop signs. d_2 is demonstrated by Fig.6 (b), which is a busy 7-way roundabout with one "yield" branch and six "stop" branches. As Fig.6 (e) shows, d_3 is collected on a highway that contains several sub-scenarios. The upper two lanes are zipper merging, and it is a ramp for the middle two lanes. Moreover, it is a forced merging for the lower three lanes where vehicles must change lanes. Fig.6 (c) and (d) show the d_5 and d_4, which are roundabouts controlled by stop signs and yield signs, respectively.

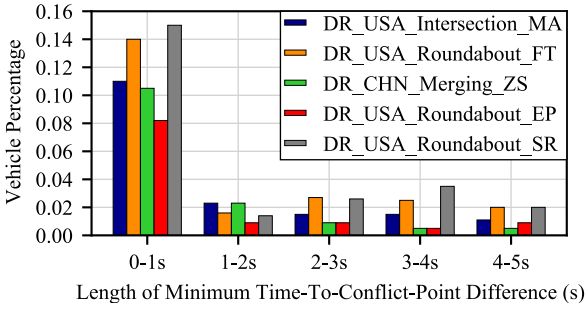


Fig. 7. Comparisons for distributions of the minimum time-to-conflict-point among selected scenarios. The x-axis represents the length of minimum time-to-conflict-point ($\triangle TTCP_{min}$) in seconds. And the y-axis are the percentage of vehicles that with particular $\triangle TTCP_{min}$ relative to total number of vehicles in the dataset. The higher values of y-axis represent larger density of interactive behaviors [33]. Detailed definition of minimum time-to-conflict is described in Appendix C.

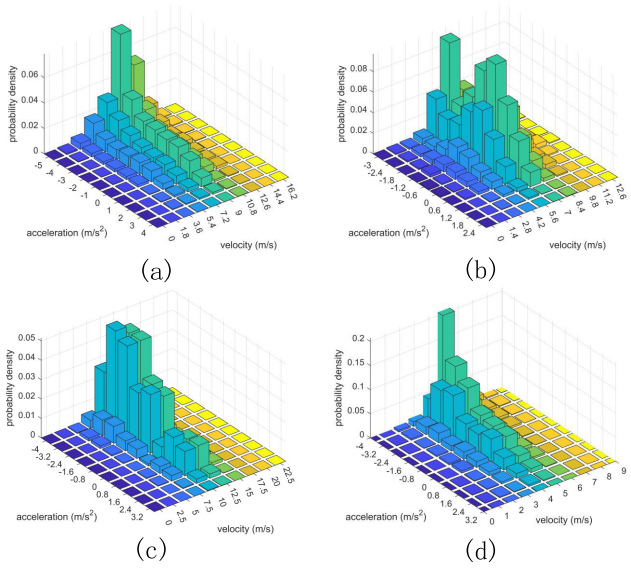


Fig. 8. Joint distributions of vehicle velocity and acceleration of different scenarios: (a) scenario d_1. (b) scenario d_2. (c) scenario d_3. (d) scenario d_4.

are processed for vehicle trajectory prediction and traffic divergence measuring, respectively, as described in Section II-A and Section III-B. In detail, the closest 5 surrounding vehicles are considered ($N = 5$) in the conditional $\mathbf{X}^N_{cond}$ for measurements of the traffic divergence. Furthermore, the number of eigenvector k in (6) is set as 3. It should be noted that only vehicles are considered in the experiments since this work focuses on modeling the interactive behaviors of vehicles, and other types of traffic participants, including pedestrians and cyclists, take a small proportion of the total dataset, as shown in Table I. Besides, according to [33], the data are smoothed by a Rauch-Tung-Striebel (RTS) smoother [35] to obtain smooth motions of the vehicles. To investigate the impacts of using the smoothed data, an experiment is conducted using data with noise, which is demonstrated in Appendix A. The training strategy of continual trajectory prediction is that the learned model is loaded to be trained on new datasets without directly accessing old data from past scenarios when a new scenario comes. Approximate 10,000 data samples are used from each scenario. The selected dataset corresponding to each scenario is split into training, validation, and testing datasets by 7:1:2.

TABLE I
INFORMATION OF SCENARIOS USED IN EXPERIMENTS

Scenario	Traffic Environmet	Vehicle Ratio	Vehicle Number
d_1	Urban Intersection	96.97%	2982
d_2	Urban Roundabout	95.31%	7496
d_3	Ramp Merging	100%	10359
d_4	Urban Roundabout	84.90%	708
d_5	Urban Roundabout	80.65%	965

TABLE II
IMPLEMENTED KEY PARAMETERS

Parameters	Implemented Values
Observation horizon	2s
Prediction horizon	4s
Input Size	2
Output Size	5
Number of ST-GCNN Layers	1
Number of TXP-CNN Layers	5
Number of Training Epochs	250
Learning Rate	0.001
Kernel Size	3

The proposed approach is appropriate for interaction-aware models using gradient descent-based methods for updating parameters. In this work, the Social-STGCNN model [36], a graph-based interaction-aware trajectory predictor, is adopted as the base model. In Social-STGCNN, interactions between vehicles are represented by a graph where nodes represent the positions of vehicles, and edges with a weighted matrix encode the spatial interdependency between vehicles. Then, graph representations are inputted into the spatiotemporal graph convolutional neural networks (ST-GCNN). Based on the operation of ST-GCNN, a time-extrapolator convolutional neural network (TXP-CNN) outputs the predicted distribution of trajectories as (3).

The model observes 2s historical trajectories and predicts trajectories for 4s. They are trained for 250 epochs using Stochastic Gradient Descent (SGD), and the learning rate is set as 0.001. Detailed implementation of models are shown in Table II. The training loss function is the negative log-likelihood:

$$l(\boldsymbol{\theta}) = -\sum \log\left(\mathrm{P}\left(\mathbf{Y}|\hat{\Gamma}\right)\right) \tag{22}$$

where $\hat{\Gamma}$ is the estimated Gaussian distribution over future trajectories $\mathbf{Y}$. Models are implemented using PyTorch.[1]

The metrics for prediction accuracy are Average Displacement Errors (ADE) and Final Displacement Errors (FDE), commonly used in trajectory predictions [9], [12], [36]. Suppose each test set has N_{ts} samples, the i^{th} sample of predicted co-ordinates of target vehicle and the ground truth at time t are denoted as $\hat{\mathbf{tr}}_{0,i}^{(t)}$ and $\mathbf{tr}_{0,i}^{(t)}$. ADE represents the average Euclidean distance for the whole predicted trajectory:

$$ADE = \frac{\sum_{i=1}^{N_{ts}} \sum_{t'=t+1}^{t+t_f} \left\| \hat{\mathbf{tr}}_{0,i}^{(t')} - \mathbf{tr}_{0,i}^{(t')} \right\|_2}{N_{ts} \times t_f}. \tag{23}$$

FDE represents the Euclidean distance for the final predicted position:

$$FDE = \frac{\sum_{i=1}^{N_{ts}} \left\| \hat{\mathbf{tr}}_{0,i}^{(t+t_f)} - \mathbf{tr}_{0,i}^{(t+t_f)} \right\|_2}{N_{ts}}. \tag{24}$$

B. Experimental Settings

This work mainly investigates the catastrophic forgetting of vehicle trajectory prediction in continuous scenarios. Instead of treating all previous tasks equally as GEM, the proposed D-GSM uses dynamic memory for CL. Dynamic memory allocates more memory resources, i.e., the memory data for tasks that are easier to be forgotten to balance the effectiveness and efficiency. D-GSM judges different tasks by measuring traffic divergence between the current task and the previous ones. To validate the rationality of dynamic memory and evaluate the performance of D-GSM, three experiments are conducted based on INTERACTION dataset. Experiments I and II investigate key factors of predictions in continuous scenarios and also validate the rationality of our design of dynamic memory. Experiment III evaluates the proposed approach. Fig. 5 demonstrates the logic of experiments in this work. It should also be noted that the constructed continuous scenarios are devised to evaluate the trajectory prediction model in the changing driving environments as AVs drive in the real world. The shift between different scenarios is simplified as switching different datasets for model training in experiments. This simplification enriches testing situations and enhances the effectiveness of the evaluation.

In detail, experiment I explores the relationship between catastrophic forgetting and traffic divergence. Base models are firstly trained with continuous scenarios consisting of two scenarios. After learning the second scenario, the model is tested with the firstly learned one. Compared to the test after learning the first scenario, the increment of prediction errors reflects the catastrophic forgetting. Then, the traffic divergence between different scenarios is presented by the weighted-CKLD. Finally, the relationship between catastrophic forgetting and traffic divergence is analyzed.

After investigating influencing factors of forgetting in experiment I, experiment II explores the retention of the proposed approach from the aspect of memory resources. The performance of the proposed approach with different memory data is compared. Models with the proposed CL approach are trained and tested in continuous scenarios. To make a clear comparison between different memory usage, models allocate 100, 500, and 1,000 samples of memory data for each previous task in the CL strategy without using the dynamic memory. The average performance over continuous scenarios is compared.

Experiment III aims at evaluating the performance of the proposed approach. Since the dynamic memory allocating is expected to be compared to the model with equal memory allocating, three groups of continuous scenarios are set, including $S_{three} = \{d_1, d_2, d_3\}$, $S_{four} = \{d_1, d_2, d_3, d_4\}$, and $S_{five} = \{d_1, d_2, d_3, d_4, d_5\}$. Each group of continuous scenarios has more than two scenarios (In continuous scenarios with only two scenarios, the memory allocation of D-GSM is the same as GSM.) The model settings are:

- **Vanilla**: The base model (Social-STGCNN) without applying continual learning approach.
- **GSM (ours)**: The base model applied with the proposed approach but without dynamic memory. The memory usage for all past scenarios are equally set as $m_{\max} = M_{cl}/(c-1)$. In this experiment, M_{cl} are 3,500 samples.
- **D-GSM (ours)**: The base model applied with the entire proposed approach. The memory usage for the r^{th} past scenario is described in (13), M_{cl} are also set as 3,500 samples.

[1] https://pytorch.org

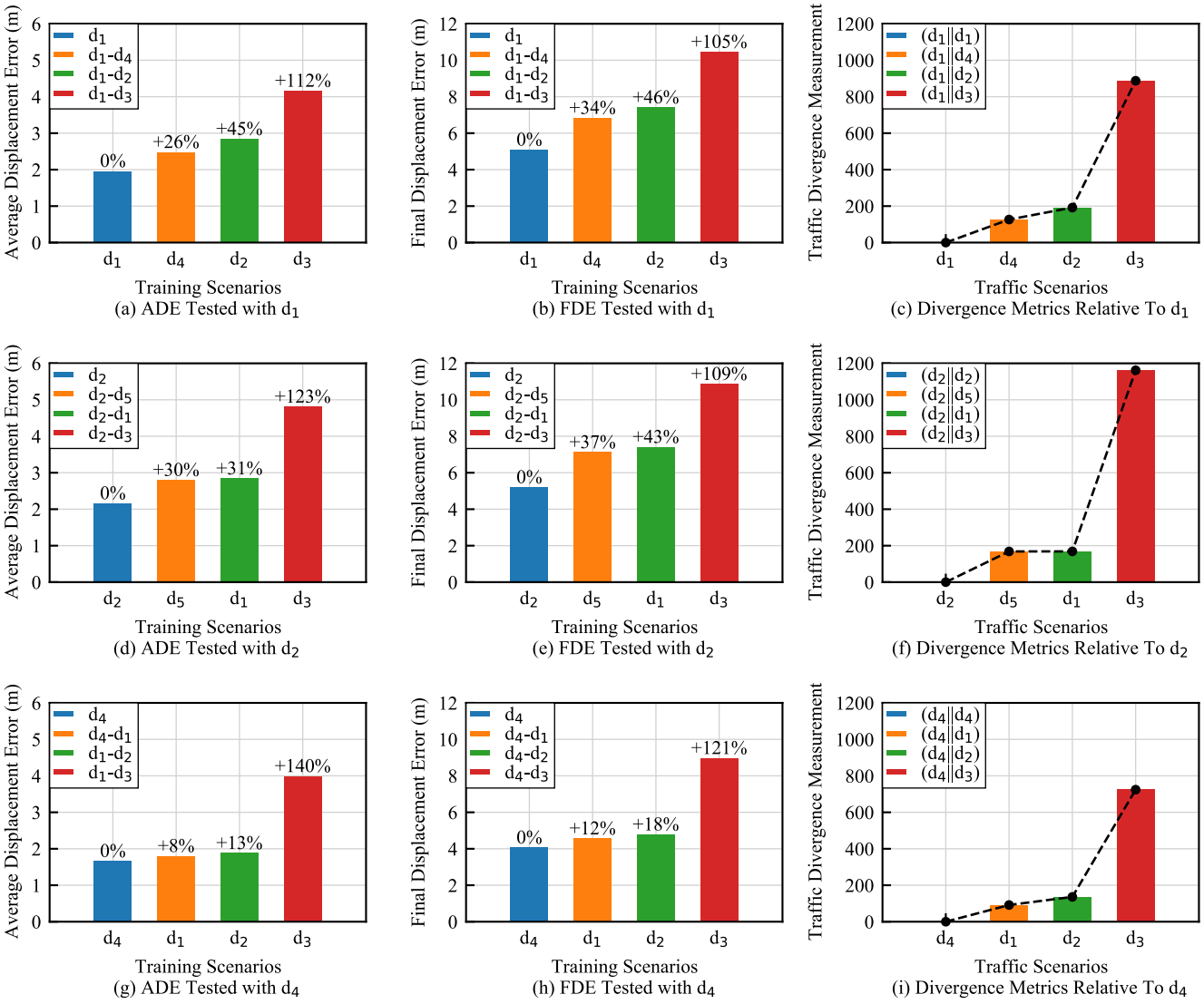


Fig. 9. Catastrophic forgetting compared with the measurement of traffic divergence.

- **Joint-Training**: Vanilla base model with joint training. It is a non-continual learning setting that does not follow the storage limitation assumptions of continual learning. Data from all scenarios are available at once. Models are trained with mixed data from all scenarios.

In addition to the experiments described above, a group of supplementary experiments are conducted to investigate the affects of the scenario definition, presented in Appendix B.

C. Experimental Results and Analysis

1) Experiment I: In the first experiment, models are tested on the past scenarios that have been observed. Experimental results of the first experiment are shown in Fig. 9. Taking the first row in Fig. 9 for example, (a) and (b) are ADE and FDE tested on scenario d_1 after observing the last scenario in continuous scenarios $S_1 = \{d_1, d_4\}$, $S_2 = \{d_1, d_2\}$, and $S_3 = \{d_1, d_3\}$. The blue bars in (a) and (b) are baselines showing the testing performance of the model trained on d_1. (c) shows the weighted-CKLD between new scenarios and the first learned scenario d_1. It can be found that, compared to baselines, ADE and FDE increase in all settings after learning the new scenario. The increased prediction errors reveal the catastrophic forgetting of trajectory predictions in continuous scenarios. The highest increment of ADE and FDE occurs in the setting $S_3 = \{d_1, d_3\}$. After learning scenario d_3, ADE and FDE increase by 112% and 105%. Compared to measurements of the traffic divergence represented by weighted-CKLD in (c), it can be found that the largest weighted-CKLD also corresponds to the setting $S_3 = \{d_1, d_3\}$.

The results are similar in the second rows (d)-(f) and third rows (g)-(i). These results are explainable intuitively. First, d_3 is a merging type scenario from a highway in China, while d_1, d_2, and d_4 are scenarios belonging to urban areas in the USA. Thus, d_3 can be regarded as a divergent scenario to others. As expected, the highest weighted-CKLDs are results between d_3 and other scenarios in all groups of Experiment I.

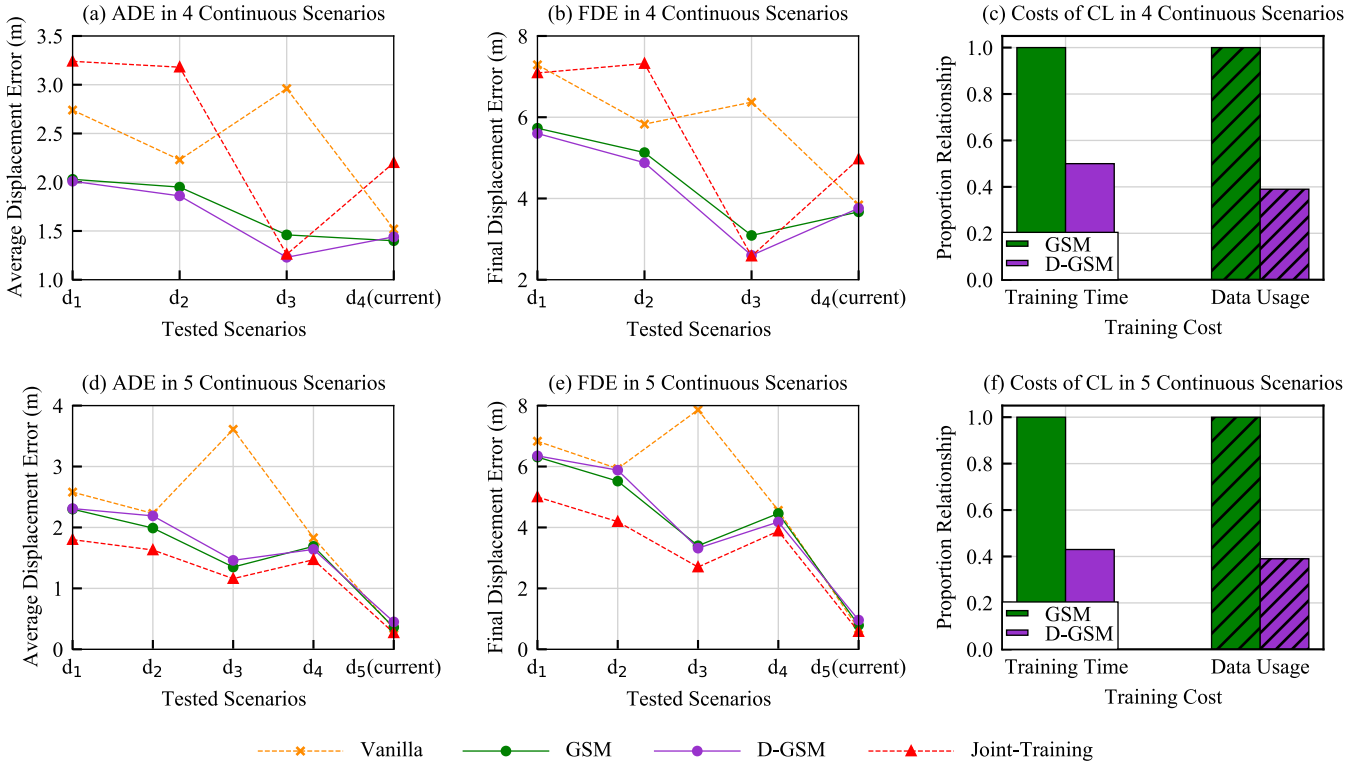


Fig. 10. Trajectory predictions in continuous scenarios: The orange cross markers with dash lines represent the predicting performance vanilla social-STGCNN model (base model). The red triangle markers with dash lines represent the performance of base model with joint-training, which has fully access to all traffic data. And the solid lines are proposed continual learning method, where the green one represents the GSM (ours) and the purple one represents the D-GSM (ours).

TABLE III
AVERAGE PREDICTING ADE(M)/ FDE(M) IN CONTINUOUS SCENARIOS WITH DIFFERENT NUMBER OF MEMORY DATA

Continuous Scenarios	0 Per Task	100 Per Task	500 Per Task	1,000 Per Task
$\{d_1, d_2\}$	2.33/5.89	2.29/5.65	2.12/5.51	**1.99/5.14**
$\{d_1, d_2, d_3\}$	3.61/8.43	2.30/5.46	2.01/4.84	**1.87/4.58**
$\{d_1, d_2, d_3, d_4\}$	2.36/5.83	1.91/4.75	1.78/4.54	**1.70/4.33**

Moreover, compared with errors and weighted-CKLD, experimental results indicate that larger traffic divergence brings higher error increments.

2) Experiment II: The second experiment compares the proposed CL approach using different memory data for continual learning. Table III shows the average ADE and FDE in three groups of continuous scenarios. The baseline is the vanilla base model which does not use memory data (0 memory data per task) and has the highest average errors (both ADE and FDE) in three tested groups. The model using the most memory data in this experiment has the lowest errors among the three groups of continuous scenarios. It can also be found that in each group, with the increment of memory data allocated to calculate the losses of past scenarios, the average errors decline. These results indicate that using more memory data for losses (14) may improve the predicting performance in continuous scenarios. However, using more memory data brings higher training costs. The third experiment will discuss more on the cost, where the training cost of two proposed model settings is compared.

3) Experiment III: The third experiment evaluates the performance of the proposed approach. First, models are continually trained in continuous scenarios. Then models are tested with all scenarios that have learned. This experiment includes three groups of continuous scenarios.Table IV shows detailed results. Compared to vanilla base model, the proposed GSM and D-GSM models have lower ADE and FDE among three groups of continuous scenarios. Besides, joint-training settings are regarded as the best possible performance in continual learning [25], [26] since all training data are accessible at once in joint training. Joint-training model has the best performance in the continuous scenarios $S_{five} = \{d_1, d_2, d_3, d_4, d_5\}$. However, we also found that in another two groups of continuous scenarios, joint-training models are not the best. The proposed models outperform the joint-training in $S_{three} = \{d_1, d_2, d_3\}$ and $S_{four} = \{d_1, d_2, d_3, d_4\}$. Here, take results in S_{three} as example to briefly discuss this "unexpected" phenomenon: It may owe to the memory mechanism of the proposed approach and the selected scenarios. In S_{three}, d_3 is a highway merging scenario collected in China, while other scenarios are from urban environments in the USA. d_3 has a larger divergence relative to others. The total number of scenarios is 3, which means that the data of d_3 has one-third of the training data for joint-training. Besides, the joint-training just mixed all data together to train the model. Due to the divergence, data from d_1 and d_2 may "weaken" the performance on d_3. Meanwhile, data from d_3 interrupt the performance on d_1 and d_2. However, since the proposed approach purely uses memory data to constrain the loss

TABLE IV
VEHICLE TRAJECTORY PREDICTING PERFORMANCE (ADE / FDE) IN CONTINUOUS SCENARIOS

Continuous Scenarios	Tested Scenario	Vanilla	Joint Training	GSM (ours)	D-GSM (ours)
Continuous Scenarios S_{three}: d_1, d_2, d_3	The 1^{st} past scenario: d_1	4.82 / 11.53	3.15 / 7.80	**2.55** / 6.36	2.67 / **6.14**
	The 2^{nd} past scenario: d_2	5.29 / 12.15	2.69 / **6.43**	**2.96** / 6.84	3.05 / 6.91
	Current scenario: d_3	**0.73** / **1.62**	1.23 / 2.54	**0.80** / **1.74**	0.90 / 1.97
	(Average)	(3.61 / 8.43)	(2.36 / 5.59)	(**2.10** / **4.98**)	(2.21 / 5.01)
Continuous Scenarios S_{four}: d_1, d_2, d_3, d_4	The 1^{st} past scenario: d_1	2.74 / 7.29	3.24 / 7.09	2.03 / 5.73	**2.01** / **5.60**
	The 2^{nd} past scenario: d_2	2.23 / 5.83	3.18 / 7.32	1.95 / 5.13	**1.86** / **4.88**
	The 3^{rd} past scenario: d_3	2.96 / 6.37	1.26 / **2.58**	1.46 / 3.09	**1.23** / 2.60
	Current scenario: d_4	1.52 / 3.84	2.20 / 4.97	**1.40** / **3.67**	1.44 / 3.76
	(Average)	(2.36 / 5.83)	(2.47 / 5.49)	(1.71 / **4.04**)	(**1.63** / 4.21)
Continuous Scenarios S_{five}: d_1, d_2, d_3, d_4, d_5	The 1^{st} past scenario: d_1	2.58 / 6.83	**1.80** / **5.00**	2.30 / 6.31	2.31 / 6.35
	The 2^{nd} past scenario: d_2	2.23 / 5.93	**1.63** / **4.19**	1.99 / 5.52	2.19 / 5.88
	The 3^{rd} past scenario: d_3	3.61 / 7.86	**1.16** / **2.70**	1.35 / 3.40	1.46 / 3.32
	The 4^{th} past scenario: d_4	1.83 / 4.56	**1.47** / **3.88**	1.69 / 4.46	1.64 / 4.18
	Current scenario: d_5	0.34 / 0.67	**0.27** / **0.58**	0.36 / 0.79	0.45 / 0.96
	(Average)	(2.12 / 5.17)	(**1.27** / **3.27**)	(1.54 / 3.05)	(1.61 / 4.13)

TABLE V
CALCULATED CKLDS BETWEEN DIFFERENT SCENARIOS

$CKLD(d\|\|d')$ (d \ d')	d_1	d_2	d_3	d_4	d_5
d_1	0	151.21	87.79	65.45	121.64
d_2	209.30	0	88.88	121.10	132.71
d_3	1230.01	1622.34	0	987.29	1110.29
d_4	152.44	171.08	109.58	0	97.99
d_5	269.04	183.51	93.31	164.99	0

increment on past scenarios, it performs well both in the current and past scenarios.

When the number of scenarios increases as in S_{five}, on the one hand, highly divergent scenario d_3 has a lower weight in the whole continuous scenarios. On the other hand, decreasing memory data may reduce the advantage of the proposed methods. As a result, in S_{five}, joint-training has the best performance. In summary, these interesting results indicate that the advantage and limitations of the proposed approach are relative to the memory room and the scenario data. There can be some ablative studies to explore more characteristics of the proposed approach in future works.

The ADE and FDE are also detailed in Table IV. Besides, CKLDs utilized to calculate weighted-CKLDs for D-GSM are shown in Table V. These experimental results show that the proposed approach improves the performance of vehicle trajectory prediction in continuous scenarios. The catastrophic forgetting of base models is alleviated.

Moreover, the proposed D-GSM aims to improve training efficiency by allocating different memory data to diverse previous tasks. In Fig. 10, the green dot markers with solid lines represent the proposed approach without dynamic memory (GSM). The purple dot markers with solid lines represent the proposed D-GSM. Yellow cross markers and red triangle markers with dash lines represent vanilla base model without continual learning strategy and base model with joint-training. Comparisons of training time and the data usage are also demonstrated in Fig. 10. Since training time depends on the scale of data and number of epochs, the last column of Fig. 10 shows the proportional relationships of training time and data usage for two settings. The green bars in Fig. 10 represent the cost of GSM, and the purple bars represent the D-GSM. (c) show the comparisons in continuous scenarios S_{four}, and (f) shows the comparisons in S_{five}. The average training time per epoch of GSM is set as 1, which corresponds to approximate 900s. Similarly, for a clear comparison, usage of memory data for GSM is also set as 1, corresponding to 3,500 samples. It is easy to understand that the ratio of time cost is close to the ratio of memory data usage since more data bring more computations. In three groups of continuous scenarios, the time cost and data cost of D-GSM are lower than GSM. The costs are approximate 0.4 to 0.5 times in S_{four} and S_{five}. It can be found that D-GSM models have a lower time cost without reducing much accuracy.

V. CONCLUSION

This paper proposes a novel continual learning approach termed D-GSM for interactive behavior learning of AVs. D-GSM constrains the loss increment on old tasks when learning a new task. With the help of memory data, the updating strategy for trainable parameters considers both data from the current and previous scenarios at each optimizing step. Therefore, compared to the non-CL approaches, which only focus on learning the current task, the proposed approach mitigates the catastrophic forgetting of interactive behavior learning in continuous scenarios without re-training. Moreover, a novel metric named weighted-CKLD is proposed to measure the traffic divergence between interactive scenarios. By utilizing the measurement of traffic divergent, a dynamic memory for the proposed CL approach is developed to improve the training efficiency. Based on various scenarios datasets from INTERACTION dataset [33], experiments are conducted to evaluate the proposed approach. Experimental results show that D-GSM outperforms non-CL models in continuous scenarios. Meanwhile, the developed dynamic memory improves the training efficiency by using less memory data.

TABLE VI
PERFORMANCE (ADE / FDE) IN CONTINUOUS SCENARIOS (TRAINING WITH NOISE DATA)

Continuous Scenarios	Tested Scenario	Vanilla (base)	D-GSM (ours)
S_{three} : $\{d_1, d_2, d_3\}$	d_1	4.04 / 9.86	**3.79 /8.24**
	d_2	4.33 / 10.01	**4.13 / 8.98**
	d_3	1.01 / 1.96	**1.19 / 2.44**
	(Average)	(3.13/ 7.28)	(**3.04 / 6.55**)
S_{four} : $\{d_1, d_2, d_3, d_4\}$	d_1	**2.85** /7.45	2.93 / **6.99**
	d_2	**3.07** / 7.53	3.09 / **7.04**
	d_3	4.62 / 10.68	**3.53/ 6.58**
	d_4	**1.82 / 4.37**	2.17 / 5.21
	(Average)	(3.09 / 7.51)	(**2.93/ 6.46**)
S_{five} : $\{d_1, d_2, d_3, d_4, d_5\}$	d_1	2.91 / 7.37	**2.89 / 7.26**
	d_2	3.30/ 8.37	**2.68 / 6.45**
	d_3	2.11 / 4.40	**1.38/ 2.76**
	d_4	**2.18** /5.34	2.25/ **5.29**
	d_5	0.98/ 1.67	**0.64 / 0.96**
	(Average)	(2.30/ 5.43)	(**1.97 / 4.54**)

The proposed approach strengthens the practicability of AVs, which is beneficial to improve the road safety and traffic efficiency of ITS. Besides, investigations on the uncertainty in traffic or explorations for different factors of traffic divergence may be the potential research area of continual interactive behavior learning for ITS.

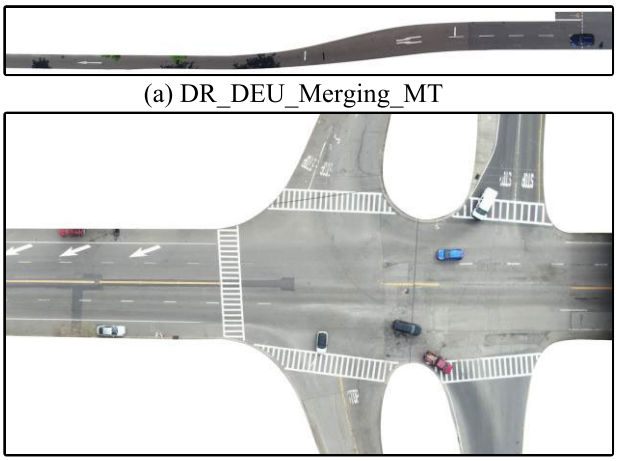
APPENDIX A
EXPERIMENTS WITH NOISE DATA

One of the simplifications in the experiments described in Section IV is that the training data are smoothed to obtain smooth motions of the vehicles. Compared to the smoothed data, the raw data collected in the real world are noisier. An experiment based on data with Gaussian noise is conducted to investigate the impacts of this simplification. In detail, first, the noise data are randomly sampled from a Gaussian distribution with a 0.1 standard deviation. Then, the sampled noise is added to all the training data. Based on the training data with Gaussian noise, the performance of the vanilla Social-STGCNN model (baseline) and the proposed D-GSM model are tested in continuous scenarios. The models are tested in each scenario that has been learned after continually learning in the continuous scenarios.

The experimental results are shown in Table VI. It can be found that, among three groups of continuous scenarios (S_{three}, S_{four}, and S_{five}), the proposed D-GSM have lower ADE and FDE in most cases, which alleviates the catastrophic forgetting of the vanilla model, as the experiments using the smoothed data demonstrated in Section IV. However, compared to the Table IV, the absolute values of ADE and FDE in Table VI are higher. These results indicate that the data with noise may reduce the prediction accuracy, while the proposed model still achieves the goal of continual learning.

APPENDIX B
EXPERIMENTS WITH A DIFFERENT DEFINITION OF SCENARIOS

The definition of the scenario is critical for the continual learning tasks of trajectory prediction. To explore whether the current definition of the scenario affects the performance of the proposed approach, experiments with newly defined scenarios are conducted to evaluate the models comprehensively.



(a) DR_DEU_Merging_MT

(b) DR_USA_Intersection_GL

Fig. 11. The maps of additional datasets used in the supplementary experiments.

A. New Definition Of Scenarios

The current definition of "scenario" is introduced in Section II and Section IV. In detail, different scenarios are represented by specific datasets collected from divergent locations, where the labels of datasets are used to distinguish scenarios. Even though two datasets collected from the same type of traffic environment but from different regions are regarded as two distinct "scenarios." For example, datasets d_2 and d_4 are collected from two urban roundabouts, which are considered as two scenarios.

In contrast to the current definition, a new definition is more macroscopic, where the same type of traffic environment is regarded as a scenario. For example, according to the new definition, the datasets d_2 and d_4 can be classified into the same scenario since they are collected from urban roundabouts. Specifically, three different scenarios, including 1) *Urban Intersection*, 2) *Urban Roundabout*, and 3) *Ramp Merging*, are defined in the supplementary experiments.

B. Experimental Settings

Additional datasets from INTERACTION datasets [33] are used to construct the continuous scenarios based on the new definition. As shown in Fig. 11, a dataset named "DR_DEU_Merging_MT" collected from a merging environment in Germany, and a dataset named "DR_USA_Intersection_GL" collected from an urban intersection in the U.S.A, are adopted in the supplementary experiments.

Referring to Section IV-A, newly defined scenarios are constructed by combining different datasets. In detail, the dataset d_1 ("DR_USA_Intersection_MA") is combined with the "DR_USA_Intersection_GL" to represent the *Urban Intersection* scenario. Datasets collected in roundabouts including d_2 ("DR_USA_Roundabout_FT") and d_4("DR_USA_Roundabout_EP") are mixed to represent *Urban Roundabout* scenario. Moreover, since

TABLE VII
PERFORMANCE (ADE / FDE) IN CONTINUOUS SCENARIOS (NEW SCENARIO DEFINITION)

Continuous Scenarios	Tested Scenario	Vanilla (base)	GSM (ours)	D-GSM (ours)
S_{II}	Intersection	1.99 / 5.12	**1.78 / 4.42**	- / -
	Roundabout	1.93 / 4.84	**1.74 / 4.46**	- / -
	(Average)	(1.96 / 4.98)	(**1.76 / 4.44**)	(- / -)
S_{III}	Intersection	3.35 /7.54	**3.08 / 6.29**	3.21 / 7.18
	Roundabout	4.61 / 9.94	4.77 / 9.20	**3.25 / 7.39**
	Merging	1.88 / 4.14	1.88 / 4.04	**1.06 / 2.19**
	(Average)	(3.28 / 7.21)	(3.24 / 6.51)	(**2.51 / 5.59**)

the dataset d_3 ("DR_CHN_Merging_ZS") and the "DR_DEU_Merging_MT" are collected in merging region and sharing the similar "zipper" rule [33], they are combined to represent *Ramp Merging* scenario.

Furthermore, two groups of continuous scenarios including $S_{II} = \{$*Urban Intersection*, *Urban Roundabout*$\}$, and $S_{III} = \{$*Urban Intersection*, *Urban Roundabout*, *Ramp Merging* $\}$ are devised for the evaluation of the proposed approach. Note that the total numbers of training samples are consistent with the experiments demonstrated in Section IV. The ratio of combined datasets in each newly defined scenario is approximately 1:1. The baseline model (vanilla) and the proposed models (GSM and D-GSM) are continually trained in S_{II} and S_{III}, and tested in each learned scenario.

C. Experimental Results and Analysis

The performance of the proposed approach is evaluated in continuous scenarios under the new scenario definition. As shown in Table VII, the prediction accuracy of the base model (vanilla) and the models applied with the proposed approach (GSM and D-GSM) are compared in newly defined continuous scenarios. In continuous scenarios S_{II},[2] ADE and FDE of the proposed GSM are lower than the base model among all tests. Similarly, in S_{III}, GSM and D-GSM outperform the base model. These experimental results show that the proposed approach achieves the primary goal of alleviating catastrophic forgetting in continuous scenarios in the new scenario definition.

Moreover, Fig. 12 demonstrates the FDE and the learning cost of D-GSM and GSM in S_{III}. As introduced in Section III, D-GSM dynamically allocates memory data based on the measurement of past scenarios. In this experiment, the divergence between the constructed new scenarios is measured using the proposed weighted CKLD. Then, D-GSM applies its dynamic memory allocation strategy to improve learning efficiency using fewer data based on the divergence measurement. The experimental results show that the proposed D-GSM achieves the best performance with less training time and data usage than GSM.

In summary, compared to the experimental results demonstrated in Section IV, the different scenario definitions may affect the absolute values of prediction errors. However, the new definition in the supplementary experiments brings little effect on the main characteristics of the proposed approach, including the ability of CL in continuous scenarios and the advantages of the dynamic memory allocation strategy.

[2]The D-GSM allocates different memory resources in different past scenarios. Since there is only one past scenario (i.e., Intersection) in S_{II}, the CL strategies of GSM and D-GSM are the same in this case.

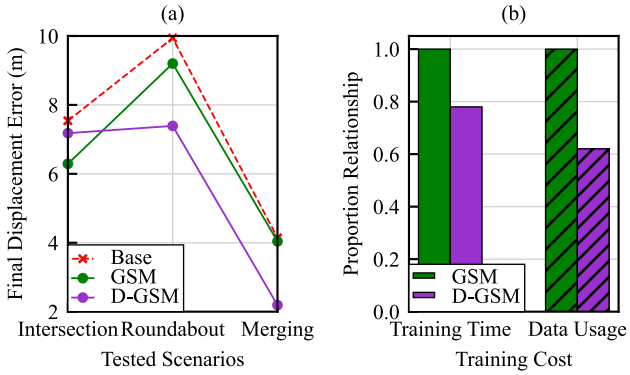


Fig. 12. The performance of the proposed model in newly defined continuous scenarios.

APPENDIX C
DEFINITION OF MINIMUM TIME-TO-CONFLICT-POINT

Referring to [33], the density of interactive behaviors of a scenario can be represented by minimum time-to-conflict-point ($\triangle TTCP_{min}$). $\triangle TTCP_{min}$ is a metric to describe the relative states between two moving vehicles in a scenario, where the paths of the two vehicles share a conflict point but without any forced stop. More details about the concept of the conflict points can be found in [33]. Assuming that $TTCP_i^t = \triangle l_i^t / v_i^t (i = 1, 2)$ is the traveling time to the conflict point of each vehicle in the interactive pairs [37]. v_i^t and $\triangle l_i^t$ are, respectively, the speed of the i^{th} vehicle and its distance to the conflict point along the path at time t. Then, $\triangle TTCP_{min}$ is defined as:

$$\triangle TTCP_{min} = \min_{t \in [T_{start}, T_{end}]} \left(TTCP_1^t - TTCP_2^t \right) \quad \text{(A.1)}$$

where T_{start} and T_{end} are, respectively, the starting time index when both vehicles appear and the crossing time index when one of the vehicles passes the conflict point. If $\triangle TTCP_{min} \leq 3$s, then it is defined that interaction exists [33], [37].

APPENDIX D
BASIC REQUIREMENT FOR CALCULATIONS OF CONDITIONAL KULLBACK-LEIBLER DIVERGENCE

As introduced in Section III-B, difference of spatiotemporal dependency among vehicles are used to represent divergence between interactive scenarios. In the implementation, the spatiotemporal dependency is formulated as the conditional probability density function (CPDF) over historical and future trajectories of vehicles, and CKLD presented in (5) is used to measure the distance between two CPDFs. Referring to (5), p_1 and p_2 are assumed as GMMs, which are firstly estimated by MDN [30]. Then, Monte-Carlo sampling is applied to calculate the original KLD between estimated GMMs. Finally, CKLD is calculated based on KLD, as formulated in (10).

The calculation described above takes memory data stored in the scenario repository as inputs. According to

TABLE VIII
TRAINING LOSS OF GAUSSIAN MIXTURE MODELS (GMMS) ESTIMATION

Scenario \ Gaussian Number	1	5	10	20
d_1	108.50	45.00	32.53	**30.75**
d_2	119.20	74.10	59.34	**46.25**
d_3	96.00	75.70	34.40	**30.00**
d_4	90.00	47.55	35.44	**26.40**
d_5	82.40	12.75	12.05	**6.25**

Section III-A, memory storage space is not infinite. The memory data for each observed scenario decreases when a new scenario comes. However, insufficient data may lead to errors during the calculation of CKLD. To discuss this problem seriously, the components of GMMs, the data amount for the CKLD calculation, and the reliability of the calculated CKLD are investigated in the supplementary experiments.

A. The Assumption of GMMs

The distributions of the scenario data are assumed as Gaussian Mixture Models (GMMs) in this work. In detail, GMMs are mixed by Gaussian distributions with different parameters. The component number of Gaussian distributions in GMMs is a hyperparameter. A simple assumption of GMMs with fewer Gaussian distributions may ease the computational burden of GMMs estimation. However, the simple assumption may also be a lack of capability to estimate a GMM precisely.

In the supplementary experiments, the training losses of MDN to estimate GMMs with different assumptions are recorded and compared. The MDN to estimate GMMs is trained with 1,000 epochs. The training process of MDE is to minimize the loss function, i.e., the negative log-likelihood [30]. Table VIII demonstrates the loss of the final epoch in different GMM assumptions with 6,000 cases of training data. In the simplest assumption, the GMMs are assumed to contain only one Gaussian distribution. The maximum component number of GMMs is 20, as in our works introduced in Section IV. From the comparison, it can be found that the loss reduces with the increase in the component number of GMMs. The lower loss indicates a more precise estimation of data distribution. Based on the experimental results, 10 or more Gaussian distribution components in GMMs are suggested in this study.

B. Basic Requirement for CKLD Calculation

There are two forms of errors when calculating CKLD by insufficient data: 1) The amount of data is too small to calculate values of CKLD. 2) The calculated CKLD may not be precise. Considering these potential errors, the requirement of obtaining values of CKLD, which are reliable to guide the dynamic allocation strategy in the proposed CL approach, is investigated in the supplementary experiments.

1) Data Amount and Process: All scenarios have the same amount of data for the calculation of CKLDs. Processed cases are 60,000 in total. It should be noted that data process refers to the condition $\mathbf{X}_{cond}^{N}$ and $\mathbf{Y}$ described in Section III-B, which differs from the processing of trajectories for prediction in this work. Besides, the dimensions of condition $\mathbf{X}_{cond}^{N}$ for CKLD calculation are different from the processed trajectories to be predicted. The exact amount of raw data can be processed into different cases for CKLD and trajectory to predict, respectively. Thus, we use "cases" to distinguish "samples," which are used in the paper to represent the processed trajectories for prediction.

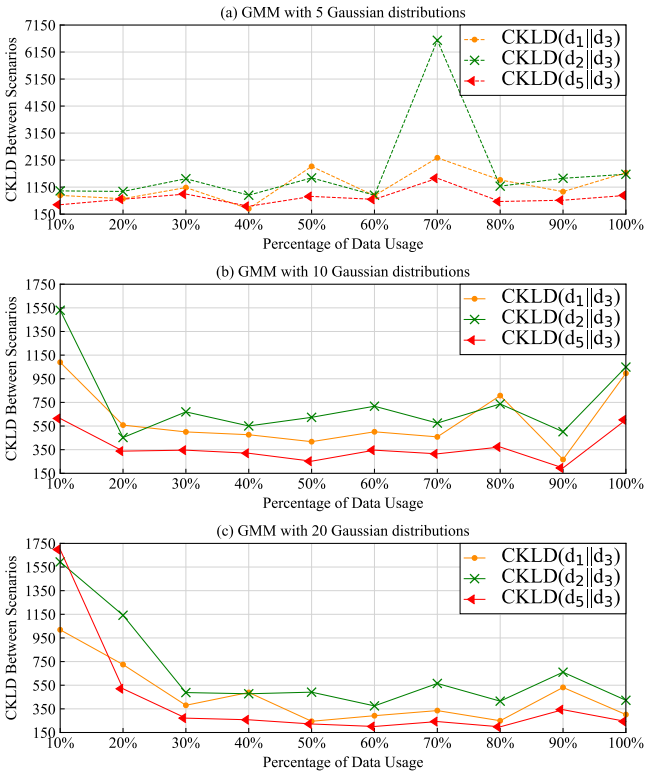


Fig. 13. The conditional Kullback-Leibler divergence (CKLD) between scenario d_3 and $d_i (i = 1, 2, 5)$ are calculated using different data amounts. The maximum number of processed cases for this experiment is 60,000, which is denoted as 100% data usage.

2) Experimental Settings: The supplementary experiments are devised to find 1) the floor of data amount to calculate CKLD and 2) the thresholds of the data amount to obtain relatively reliable CKLD under different GMM assumptions. Based on the discussions in Appendix D-A, three groups of assumptions representing different complexities, including a GMM containing 5, 10, and 20 Gaussian distributions, are discussed. Under these assumptions, CKLD values among d_1 to d_5 scenarios are calculated in pairs using various amounts of data. Considering the limited length of the article, only the main experimental results are demonstrated in the following.

3) Experimental Results and Analysis: As shown in Fig. 13, CKLDs (the y-axis) between scenarios $d_i(1, 2, 5)$ and d_3, calculated by different data amount (the x-axis) under assumptions including GMM with 5 (Fig. 13 (a)), 10 (Fig. 13 (b)), and 20 (Fig. 13 (c)) Gaussian distributions are compared. Note that the scale of the y-axis in Fig. 13 (a) is much higher than (b) and (c). The actual fluctuation of the CKLD curve in Fig. 13 (a) is drastic, indicating the unsteadiness of the results. This phenomenon may be due to the imprecise estimation of GMMs, as discussed in Section D-A. The following analysis

TABLE IX
BASIC REQUIREMENT OF DATA AMOUNT FOR CKLD CALCULATIONS

Gaussian Number	Floor Of Calculability	Recommended Threshold
10	3,900 (data cases)	$>$ 12,000 (data cases)
20	6,000 (data cases)	$>$ 18,000 (data cases)

TABLE X
D-GSM PERFORMANCE (ADE / FDE) WITH DIFFERENT SCENARIO DIVERGENCE MEASUREMENTS IN CONTINUOUS SCENARIOS S_{three}

Compared Metrics	Tested Scenario	Measured by 70%	Measured by 90%	Measured by 100%
Prediction Accuracy	d_1	2.57 / 6.25	2.65 / 6.78	2.67 / 6.14
	d_2	2.80 / 6.24	2.88 / 6.68	3.05 / 6.91
	d_3	0.92 / 1.96	0.74 / 1.64	0.90 / 1.97
	(Average)	(2.10 / 4.87)	(2.09 / 5.03)	(2.21 / 5.01)
Training Cost	Epoch Time	753 s	786 s	780 s

will focus on the experimental results of GMMs with 10 and 20 Gaussian distributions to obtain an insightful conclusion for the practical application.

From the experimental results demonstrated in Fig. 13 (b) and (c), it can be found that the CKLD results are relatively stable using data more than 20% (12,000 processed cases) for the GMM with 10 Gaussian distributions and using data more than 30% (18,000 processed cases) for GMM with 20 Gaussian distributions. These thresholds are recommended as a reference when devising the CL memory system applied to the proposed method. Furthermore, we decline the data amount gradually during the experiment to find the floor, i.e., the lower bound, of CKLD calculability. When the declined data amount is less than the "floor," null values appear. In other words, the data less than the "floor" are insufficient to estimate the GMMs, and the CKLD cannot be obtained. The experimental results are summarized in Table IX. The floor of calculability for GMM with 10 Gaussian distributions is 3,500 and 6,000 for the GMM with 20 Gaussian distributions.

C. Discussion on the Reliability of Calculated CKLD

In the supplementary experiment, the relatively steady results are regarded as reliable results. Since using GMM with 20 Gaussian distribution is the most practical assumption in this work, the experimental results of this group are highlighted for more discussions. Note that the CKLD values increase at 70% and 90%. Are these increased values can be trusted to apply to the proposed CL approach? To answer this question, we applied these CKLD results to the proposed D-GSM. Table X shows the CL performance of D-GSM models guided by different CKLD-based measurements, including CKLDs calculated by 70%, 90%, and 100% amount of data. The experimental results show that the prediction accuracy and the training cost are very close in these three groups, which indicate that the increased CKLDs have little impacts on the performance of the proposed CL approach.

In summary, the floor of calculability and the threshold to obtain relatively stable results are provided as the basic requirements of CKLD calculations. However, since the proposed traffic divergence measurement is a learning-based numerical method, the specific requirement in different problems may be affected by the data used with specific data processes. We hope the conclusions in the supplementary experiments can provide a reference in related research areas, and welcome researchers to explore further in the traffic divergence measurement studies.

REFERENCES

[1] M. N. Azadani and A. Boukerche, "Driving behavior analysis guidelines for intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6027–6045, Jul. 2022.

[2] F. Leon and M. Gavrilescu, "A review of tracking and trajectory prediction methods for autonomous driving," *Mathematics*, vol. 9, no. 6, p. 660, Mar. 2021.

[3] Z. Huang, H. Liu, J. Wu, and C. Lv, "Differentiable integrated motion prediction and planning with learnable cost function for autonomous driving," 2022, *arXiv:2207.10422*.

[4] W. Zhan, A. de La Fortelle, Y.-T. Chen, C.-Y. Chan, and M. Tomizuka, "Probabilistic prediction from planning perspective: Problem formulation, representation simplification and evaluation metric," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1150–1156.

[5] M. Brännström, E. Coelingh, and J. Sjöberg, "Model-based threat assessment for avoiding arbitrary vehicle collisions," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 658–669, Sep. 2010.

[6] A. Polychronopoulos, M. Tsogas, A. J. Amditis, and L. Andreone, "Sensor fusion for predicting vehicles' path for collision avoidance systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3, pp. 549–562, Sep. 2007.

[7] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH J.*, vol. 1, no. 1, pp. 1–14, Dec. 2014.

[8] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behavior prediction for autonomous driving applications: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 33–47, Jan. 2022.

[9] 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. (CVPR)*, Jun. 2016, pp. 961–971.

[10] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1549–15498.

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

[12] Z. Li, C. Lu, Y. Yi, and J. Gong, "A hierarchical framework for interactive behaviour prediction of heterogeneous traffic participants based on graph neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9102–9114, Jul. 2022.

[13] J. Sun, Q. Jiang, and C. Lu, "Recursive social behavior graph for trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 657–666.

[14] Z. Li, J. Gong, C. Lu, and J. Xi, "Importance weighted Gaussian process regression for transferable driver behaviour learning in the lane change scenario," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12497–12509, Nov. 2020.

[15] Z. Li, J. Gong, C. Lu, and J. Li, "Personalized driver braking behavior modeling in the car-following scenario: An importance-weight-based transfer learning approach," *IEEE Trans. Ind. Electron.*, vol. 69, no. 10, pp. 10704–10714, Oct. 2022.

[16] 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, Jun. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253519307377

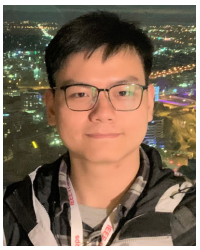
[17] Z. Chen and B. Liu, "Lifelong machine learning," *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 12, no. 3, pp. 1–207, 2018.

[18] A. Pentina and C. H. Lampert, "Lifelong learning with non-iid tasks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1540–1548. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/2969239.2969411

[19] A. Cossu, A. Carta, V. Lomonaco, and D. Bacciu, "Continual learning for recurrent neural networks: An empirical evaluation," *Neural Netw.*, vol. 143, pp. 607–627, Nov. 2021.
[20] M. Perkonigg et al., "Dynamic memory to alleviate catastrophic forgetting in continual learning with medical imaging," *Nature Commun.*, vol. 12, no. 1, pp. 1–12, Sep. 2021.
[21] G. M. van de Ven, H. T. Siegelmann, and A. S. Tolias, "Brain-inspired replay for continual learning with artificial neural networks," *Nature Commun.*, vol. 11, no. 1, pp. 1–14, Aug. 2020.
[22] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 2994–3003. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/3294996.3295059
[23] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6470–6479. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/3295222.3295393
[24] D. Kudithipudi et al., "Biological underpinnings for lifelong learning machines," *Nature Mach. Intell.*, vol. 4, no. 3, pp. 196–210, 2022.
[25] H. Ma, Y. Sun, J. Li, M. Tomizuka, and C. Choi, "Continual multi-agent interaction behavior prediction with conditional generative memory," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 8410–8417, Oct. 2021.
[26] P. Bao, Z. Chen, J. Wang, D. Dai, and H. Zhao, "Lifelong vehicle trajectory prediction framework based on generative replay," 2021, *arXiv:2111.07511*.
[27] Z. Li, J. Gong, C. Lu, and Y. Yi, "Interactive behavior prediction for heterogeneous traffic participants in the urban road: A graph-neural-network-based multitask learning framework," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 3, pp. 1339–1349, Jun. 2021.
[28] Z. Li et al., "An ensemble learning framework for vehicle trajectory prediction in interactive scenarios," 2022, *arXiv:2202.10617*.
[29] B. Yu, Y. Lee, and K. Sohn, "Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (GCN)," *Transp. Res. Part C: Emerg. Technol.*, vol. 114, pp. 189–204, May 2020.
[30] C. M. Bishop, "Mixture density networks," Aston Univ., Birmingham, U.K., Tech. Rep. NCRG/94/004, 1994. [Online]. Available: https://publications.aston.ac.uk/id/eprint/373/
[31] J. J. Langille and R. E. Brown, "The synaptic theory of memory: A historical survey and reconciliation of recent opposition," *Frontiers Syst. Neurosci.*, vol. 12, p. 52, Oct. 2018.
[32] P. S. B. Finnie and K. Nader, "The role of metaplasticity mechanisms in regulating memory destabilization and reconsolidation," *Neurosci. Biobehav. Rev.*, vol. 36, no. 7, pp. 1667–1707, Aug. 2012.
[33] W. Zhan et al., "Interaction dataset: An international, adversarial and cooperative moTION dataset in interactive driving scenarios with semantic maps," 2019, *arXiv:1910.03088*.
[34] B. R. Kiran et al., "Deep reinforcement learning for autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022.
[35] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Hoboken, NJ, USA: Wiley, 2006.
[36] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14412–14420.
[37] W. Zhan, L. Sun, D. Wang, Y. Jin, and M. Tomizuka, "Constructing a highly interactive vehicle motion dataset," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 6415–6420.

Yunlong Lin received the B.S. degree in mechanical engineering from the Beijing Institute of Technology (BIT), Beijing, China, in 2022, where he is currently pursuing the Ph.D. degree. His research interests include interactive behavior modeling, continual learning, and decision-making of intelligent vehicles.

Zirui Li (Student Member, IEEE) received the B.S. degree from the Beijing Institute of Technology (BIT), Beijing, China, in 2019, where he is currently pursuing the Ph.D. degree in mechanical engineering. From June 2021 to July 2022, he was a Visiting Researcher with the Delft University of Technology (TU Delft). Since August 2022, he has been a Visiting Researcher with the Chair of Traffic Process Automation with the Faculty of Transportation and Traffic Sciences "Friedrich List," TU Dresden. His research interests include interactive behavior modeling, risk assessment, and the motion planning of automated vehicles.



Cheng Gong (Graduate Student Member, IEEE) received the B.S. degree in mechanical engineering from the Beijing Institute of Technology, China, in 2020, where he is currently pursuing the Ph.D. degree. His research interests include intelligent vehicles, motion planning and control, machine learning, and life-long learning.

Chao Lu (Member, IEEE) received the B.S. degree in transport engineering from the Beijing Institute of Technology (BIT), Beijing, China, in 2009, and the Ph.D. degree in transport studies from the University of Leeds, Leeds, U.K., in 2015. In 2017, he was a Visiting Researcher with the Advanced Vehicle Engineering Centre, Cranfield University, Cranfield, U.K. He is currently an Associate Professor with the School of Mechanical Engineering, BIT. His research interests include intelligent transportation and vehicular systems, driver behavior modeling, reinforcement learning, and transfer learning and its applications.

Xinwei Wang (Member, IEEE) received the Ph.D. degree from Beihang University, China, in 2019. He is currently a Lecturer (an Assistant Professor) with the Queen Mary University of London (QMUL), U.K. He was a Post-Doctoral Researcher with TU Delft, The Netherlands, from 2020 to 2022. Prior to that, he was a Post-Doctoral Researcher with QMUL from 2019 to 2020. Over the years, he has integrated artificial intelligence and systems engineering for risk assessment, motion planning, and decision making in intelligent systems. He was a recipient of Marie Sklodowska-Curie Actions Co-Fund Fellowship in 2022 and IEEE ITSS Young Professionals Travelling Fellowship in 2022. He has authored over 20 papers, including those in *TR Part C*, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE TRANSACTIONS ON FUZZY SYSTEMS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS.

Jianwei Gong (Member, IEEE) received the B.S. degree from the National University of Defense Technology, Changsha, China, in 1992, and the Ph.D. degree from the Beijing Institute of Technology (BIT), Beijing, China, in 2002. Between 2011 and 2012, he was a Visiting Scientist with the Robotic Mobility Group, Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently a Professor with the School of Mechanical Engineering, BIT. His research interests include intelligent vehicle environment perception and understanding, decision making, path/motion planning, and control.