Continual Learning for Motion Prediction Model via Meta-Representation Learning and Optimal Memory Buffer Retention Strategy

DaeJun Kang[1] Dongsuk Kum[2] Sanmin Kim[2]
[1]Korea Automotive Technology Institute
[2]Korea Advanced Institute of Science and Technology
djkang@katech.re.kr, {dskum,sanmin.kim}@kaist.ac.kr

Abstract

Embodied AI, such as autonomous vehicles, suffers from insufficient, long-tailed data because it must be obtained from the physical world. In fact, data must be continuously obtained in a series of small batches, and the model must also be continuously trained to achieve generalizability and scalability by improving the biased data distribution. This paper addresses the training cost and catastrophic forgetting problems when continuously updating models to adapt to incoming small batches from various environments for real-world motion prediction in autonomous driving. To this end, we propose a novel continual motion prediction (CMP) learning framework based on sparse meta-representation learning and an optimal memory buffer retention strategy. In meta-representation learning, a model explicitly learns a sparse representation of each driving environment, from road geometry to vehicle states, by training to reduce catastrophic forgetting based on an augmented modulation network with sparsity regularization. Also, in the adaptation phase, We develop an Optimal Memory Buffer Retention strategy that smartly preserves diverse samples by focusing on representation similarity. This approach handles the nuanced task distribution shifts characteristic of motion prediction datasets, ensuring our model stays responsive to evolving input variations without requiring extensive resources. The experiment results demonstrate that the proposed method shows superior adaptation performance to the conventional continual learning approach, which is developed using a synthetic dataset for the continual learning problem.

1. Introduction

For autonomous driving, reliable motion prediction in various driving environments is essential. However, while current motion prediction studies perform well on given benchmark datasets, their performance deteriorates significantly in changed environments, for example, new driving patterns due to different road types never seen before [1]. Since the driving environment is non-stationary and changes over time and place, it is necessary to update the motion prediction model seamlessly under changing environments to maintain reliable performance as shown in Fig. 1.

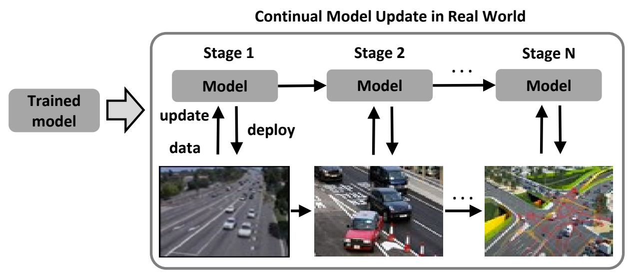

Figure 1. Continual model update problem for scalability to changing environments.

Joint training described in [25], as an example is a representative conventional method for such a model update. This method re-trains the model with all existing datasets and data from new environments upon each update. Therefore, it guarantees decent performance across all environments by embracing the entirety of the data distribution. However, it demands substantial training resources, which escalate as data volume increases due to the redundant training inherent in the constant update process. Moreover, performance degradation in data-scarce environments is a notable challenge stemming from the learning's bias toward more abundantly represented data. Conversely, transfer learning [28] adopts a more streamlined approach by utilizing solely the data from new environments for learning, significantly enhancing the efficiency of the model update process. However, this method encounters a notable limitation: it struggles to retain previously acquired knowledge while adjusting to a specific target environment, posing a challenge to preserving learned experiences.

To tackle the above problem, continual learning, which focuses on updating models in response to sequences of data from varying distributions, has emerged as a promising area of research. While numerous studies are in progress, the majority are focused on foundational research domains, like image classification [42]. Remarkably, the domain of motion prediction remains largely unexplored. While a few of studies have been studied adaptations between specific road types [41] or datasets [19], the field has yet to embrace and expand upon research into a continual model update framework.

Therefore, this study addresses the significant challenge of performance degradation in motion prediction models due to changes in driving environments. We introduce a groundbreaking Continual Motion Prediction (CMP) learning framework designed to mitigate catastrophic forgetting of prior knowledge while efficiently assimilating new environmental data. Our innovative CMP framework unfolds in two strategic phases: first, a generic meta-representation is cultivated during the pre-training phase, followed by implementing an optimal memory buffer retention strategy during the adaptation phase. This dual-step approach promises a robust solution to the perennial issue of balancing past learning with new experiences.

In preparation for the continual model update, our motion prediction model undergoes a vital initial phase termed pre-training. During this stage, the model is trained to strategically acquire sparse representations to mitigate interference from subsequent updates with new data. By mastering distinct representations from each input distribution within the constraints of a limited model capacity, we aim to minimize interference from disparate inputs' overlap. To achieve this objective, we undertake representation learning, targeting the reduction of catastrophic forgetting loss induced through simulated scenarios of continual model updates based on a modulation network with sparsity regularization. This approach ensures our model is primed to seamlessly adapt to evolving datasets while preserving essential knowledge from past experiences. In the *adaptation phase*, our model integrates and updates with new environmental data at each stage, ensuring continuous and seamless refinement. At every stage, carefully chosen data is preserved within the memory buffer with limited capacity, then merged with incoming data in the following stage, ensuring a sophisticated and seamless adaptation process. In the merging process, we implement an optimal memory buffer retention strategy encompassing the entire data spectrum. This approach selectively curates data distinct from the current memory contents for storage, optimizing the use of limited buffer space at each stage to ensure a comprehensive and diverse data representation. By embracing the full spectrum of the data distribution using a minimal number of data, this strategic approach eliminates the redundancy of reprocessing familiar data and effectively minimizes gradient interference, safeguarding against catastrophic forgetting. Through these two innovative methods, the motion prediction model achieves continual learning, adeptly adapting to evolving driving conditions while maintaining peak performance in previous environments.

The main contributions of this study can be summarized as follows:

- We pioneered the problem formulation to facilitate the continual updating of the motion prediction model, specifically tailored to meet the demands of commercial autonomous driving AI.
- We proposed the novel generic model representation learning method that reveals the inherent sparsity within each scene's representation, effectively mitigating catastrophic forgetting.
- Recognizing the inefficiencies of successive joint training in real-world applications, we introduced a groundbreaking optimal memory buffer retention strategy. This innovative approach, rooted in adaptive sampling, ensures efficient continual model updates by effectively managing gradient interference between previous and current stages.

We constructed the virtual continual motion prediction model update problem on nuScenes dataset and verified the proposed method based on the designed problem formulation.

2. Related Work

Motion Prediction. Conventional studies have predominantly emphasized refining knowledge representation of the surrounding environment through deep learning [29, 31, 46, 51–53]. Interaction-aware motion prediction model [9–11, 13, 39] have been studied to consider the interaction of surrounding vehicles. Recently, the focus has shifted towards forecasting the future trajectory of the ego-vehicle [2, 36, 38]. Studies have also explored leveraging infrastructure-related data in motion prediction, employing Convolutional Neural Network(CNN) [4, 8, 12, 14, 24, 27, 30, 35, 49] and the Graph Neural Network(GNN) [12, 24] to encode HD-map information through knowledge representation and concatenation techniques. Furthermore, research [40] has delved into utilizing lane information to predict multiple trajectories, accommodating the potentiality of diverse future paths.

While these studies have demonstrated progressively improving performance, their validity is confined to specific datasets, and they have yet to address testing scenarios in new environments. [1] reveals that the motion prediction model works in a trained environment but often fails in diverse driving scenarios generated by the real world. A recent study has explored the efficacy of transfer learning between different road types for adapting to new environments [41]. Also, the study has conducted domain adaptation between benchmark datasets [19]. However, the scope of these environments remains limited compared to the vast array of real-world driving scenarios. Consequently, they fail to ensure the diversity necessary for accurate motion prediction in genuine environments. Extended research into continual domain adaptation is imperative to accurately reflect the diverse driving scenarios encountered in real-world settings.

Continual Learning. An adaptation process becomes essential for effective motion prediction in light of the dynamically evolving driving environment. Continual learning specifically addresses challenges encountered when updating the model throughout this adaptation process. There are three approaches to continual learning: regularization-based [22, 25, 44, 48], replay-based [5, 6, 16, 17, 21, 26, 33, 37], and dynamic model architecture [7, 15, 32, 34, 43, 45, 47, 50]. Regularization-based methods ensure performance retention over previous tasks by using update constraints on network weights that greatly impact previous tasks. Additionally, studies have been conducted to learn representation sparsity to maintain activation of only a minimal subset of representation vectors at each input, thereby enhancing efficiency. Replay-based methods store samples from past tasks using a memory buffer and utilize it to model updates. Utilizing this memory buffer enables the model to learn historical data along with sequential inputs, ensuring performance across previous environments during the adaptation process. While reservoir sampling is a primary method for the memory buffer, its uniform data sampling across stages would result in suboptimal buffer status in real environments. The dynamic model architecture dynamically activates neurons or layers to accommodate incremental classes or tasks. While this method effectively addresses catastrophic forgetting through learning via selective model expansion, its compatibility with embedded systems is limited due to constraints in model size dictated by hardware capacity.

Furthermore, given that the approaches above primarily focus on validating concepts using fundamental deep learning problems like MNIST classification, it is imperative to conduct additional research into problem formulation and learning methodologies tailored to applications such as motion prediction.

3. Problem Formulation

This study frames the challenge of adapting the motion prediction model across various driving environments, efficiently updating the model with sequential data from each environment. Therefore, we devise a virtual continual learning scenario, simulating consecutive updates to the motion prediction model. The scenes dataset features a rich collection of scenes from diverse locations, with each scene

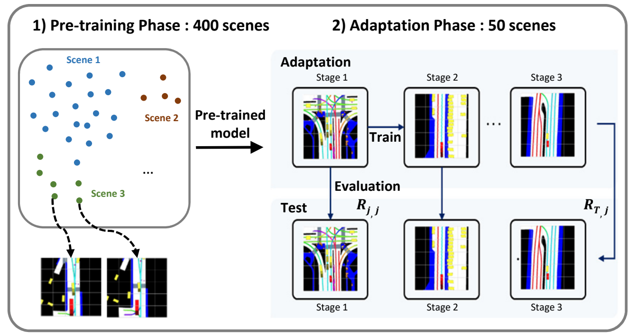

Figure 2. Problem formulation for scalability in motion prediction. One circle is a training example. The color denotes a scene. Each scene has a motion prediction sequence for one or more target vehicles

offering sequences for one or more target vehicles, organized into ten to thirty tailored motion prediction training examples. In our virtual scenario, we treat data from each scene as sequentially incoming, setting our objective to train the motion prediction model scene by scene. We establish a sequence of tasks $[\tau_1, \tau_2, \cdots]$, each trained sequentially with their respective incoming data $[D_1, D_2, \cdots]$. As the scene shifts, retraining the model for new locations is crucial, simultaneously minimizing catastrophic forgetting of prior tasks. Given the significant influence of map details and traffic conditions on motion prediction models, this approach is well-suited for continual learning scenarios within motion prediction applications. In this scenario, sequential data entry with varying distributions disrupts the independent and identically distribution (IID) sampling conditions essential for stochastic gradient descent. Consequently, when the model updates based on sequential tasks, optimizing for a new task can lead to performance declines in previous tasks due to the gradient directions of model loss. Hence, our goal is to design the CMP learning framework that can minimize catastrophic forgetting and optimize performance across all T tasks encountered up to any time step j throughout the adaptation process. Throughout this process, the model is updated using minimal data for each incoming sequential stage, assessing the efficacy of adaptation. Furthermore, the continual model update process verifies whether performance has degraded in the previous stage. Fig.2 illustrates the overall process and training examples for each task sourced from the NuScenes dataset.

4. Methodology

When updating the motion prediction model, the sequential input of different driving environments often leads to overlapping representations and misalignment in gradient direction, resulting in catastrophic forgetting. To address this challenge comprehensively, we propose two key strategies: meta-representation learning in the pre-training phase (Sec-

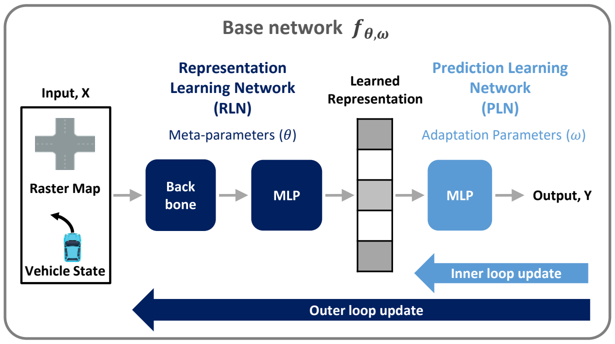

Figure 3. Base model architecture for the meta-representation in motion prediction.

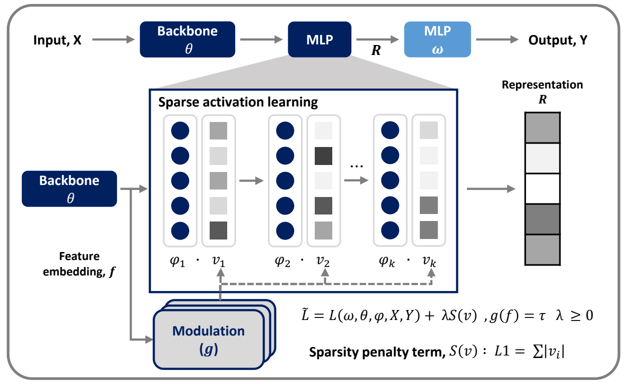

Figure 4. Modulation via regularization for sparse representation.

tion 4.1) and an optimal memory buffer retention strategy in the adaptation phase (Section 4.2).

4.1. Meta Representation Learning in Training

Our meta-representation learning approach focuses on cultivating sparse representations tailored to each input distribution that are robust to catastrophic forgetting. This approach effectively mitigates interference between diverse inputs while facilitating the rapid assimilation of new associations.

Meta-Representation Learning for Motion Prediction. Inspired by Online-aware Meta-Learning (OML) [20], we employ a base model architecture comprising a Representation Learning Network (RLN) and a Prediction Learning Network (PLN), as depicted in Fig. 3. The RLN encodes vehicle states and map information, while the PLN decodes the learned representations into the output.

To learn representations resilient to catastrophic forgetting, we simulate a continual learning process during the pre-training phase. This involves a two-step training process consisting of inner and outer loops. In the inner loop update, we sequentially sample the inner loop training sequences $\{S_1^I, S_2^I, \cdots, S_k^I\}$ from the pre-training data D where $S_i^I = \{s_{i,1}^I, \cdots s_{i,m}^I\}$. Here S^I and s^I denote a sequence and sample, respectively. Subsequently, the PLN undergoes k serial updates using these sequences while the parameters in the RLN remain frozen. Following this, we randomly sample outer loop training samples $S^O = \{s_1^O, \cdots, s_l^O\}$, and both the RLN and PLN are updated concurrently in the outer loop. Note that we set $l > mk$ to provide sufficient data for the RLN update.

This approach enables us to emulate a continual learning process during the pre-training phase, guiding the model to learn representations robust to catastrophic forgetting. By updating the model towards retaining learned representations while adapting to new data, we ensure its adaptability to changing input landscapes. Algorithm 1 describes our pioneering approach to meta-representation learning for the continual update of the motion prediction model.

Sparsity Regularization with Modulation Network. Ideally, activating a minimal subset of neurons in response to each unique input can significantly reduce catastrophic forgetting. This is because updates to the model will only adjust a limited set of weights per input, preserving the integrity of learned representations. Hence, we aim to achieve representations devoid of inactive neurons across all data distributions while ensuring sparsity in the neural response to individual inputs. In addition to leveraging meta-representation learning, we enhance the representation learning framework by integrating and training a modulation network to incentivize explicitly sparse representation. As shown in Fig. 4, the modulation network shares the feature vector from the backbone of RLN and uses it as an input. It transfers the input feature vector to the final layers of existing RLN through the modulation network. In a network structured in Fig. 4, we apply the L1 loss to the modulation network output vector (v) and add it to the original loss term to penalize. We enhance the model's representation by applying L1 loss to the modulation network's output

Algorithm 1 Meta-representation learning process

Input: D: Pre-training data, α,β: Each learning rate for inner loop and outer loop, m: No. of inner gradient steps per update, θ: RLN parameters, ω: PLN parameters

Random initialize θ
while do
  Random initialize ω
  $\{S_1^i, S_2^i, \cdots, S_k^i\} \sim D$, Sample data sequences
  $\omega_0 = \omega$
  for $i = 1, 2, \cdots, k$ **do**
    $\omega_i = \omega_{i-1} - \alpha \nabla_{\omega_{i-1}} L(f_{\theta,\omega_{i-1}}, S_i^I)$
  end for
  $S^O \sim D$, Sample data for outer loop training
  $\theta = \theta - \beta \nabla_\theta L(f_{\theta,\omega_k}, S^O)$
end while

vector (v), integrating it with the original loss to introduce a sparsity penalty. The application of L1 loss naturally encourages the emergence of sparse vectors. Consequently, this method directly induces learning of sparse representations determining neuron activation within the RLN layer through the dot product of the modulation network's output with the existing RLN.

4.2. Optimal Memory Buffer Retention Strategy

In section 1, we pointed out that there are limitations in storing and using many data samples due to resource limitations and cost issues. Thus, we use a memory buffer to keep some portion of previous data and restrain catastrophic forgetting due to gradient interference using it. Most studies apply a reservoir sampling algorithm [23] to store sequentially incoming data in a memory buffer at an equal ratio. The reservoir sampling simply samples $D_i \in X_i$ with probability $\frac{1}{i}$, remove D_i with probability $\frac{(i-1)}{i}$ at stage i. $[X_1, X_2, \cdots, X_N]$, $[D_1, D_2, \cdots, D_N]$ represents samples in the memory buffer at stage N and stream data for each stage. According to this algorithm, the probability that a random jth sample is in the memory buffer remains uniformly $\frac{1}{N}$. However, the significant challenge of the real scenario is that data distributions are non-stationary during the adaptation process, and there is no prior information about these distribution shifts. Therefore, if data is stored in the memory buffer at the same ratio in each stage, we can not hold the memory buffer in an optimal state that reflects the entire data distribution. In this study, we apply the strategy using the difference in data distribution between the memory buffer and the current stage to maintain the memory buffer in an optimal state that covers the entire data distribution. To efficiently determine how much the data distribution changes, we consider the similarity of the representation vector. The similarity decreases when the model receives data from the shifted data distribution at any stage. Using this tendency, we design algorithm 2 to detect the data distribution shift whenever the similarity is below some pre-defined threshold.

Algorithm 2 Data distribution shift detection (Adaptation process)

Input: Stream data D, Representation Learning Network f_θ

```
R=[], Initialize memory buffer
while do
    D_i ~ D, Sample a data sequence at stage i
    D_a ← D_i, Batch of sequential data from D_i
    b_m ← Sample(R), Batch from memory buffer
    sim similarity(f_θ(D_a), f_θ(b_m))
    if sim ≦ threshold then
        R ← D_a, Update memory buffer with reservoir sampling
    end if
end while
```

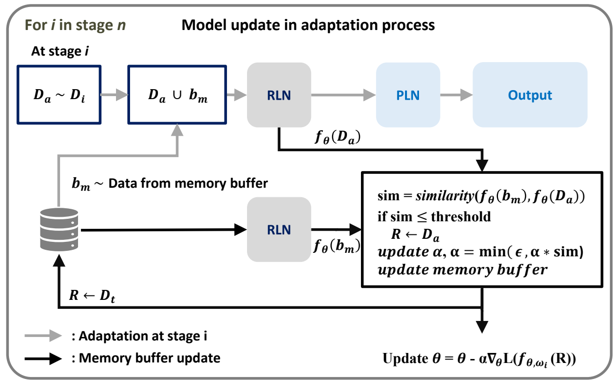

Figure 5. The overall continual model update process.

4.3. Overall Continual Model Update Framework

In the entire adaptation phase, each stage data is received and divided into adaptation and evaluation data, dataD_a for adaptation is combined with memory buffer data b_m to update the model by additional learning. During the learning process, the memory buffer is updated through adaptive sampling in section 4.2. Then, the representation learning network is trained in section 4.1 through the updated memory buffer data and continually proceeds to the next stage. The overall continual model update process is described in Fig.5.

5. Experiment

5.1. Experimental Setup

This study evaluates the adaptation and sustainability performance during sequential model updates. We validate the optimization performance of the overall framework by integrating a pre-trained model into the adaptation process during testing.

We train and evaluate our approach on nuScenes benchmark [3]. Initially, we utilize 400 scenes to establish a robust meta-representation during the pre-training phase. Then, we sequentially update the trained model across the remaining 50 scenes, simulating the continual model update process during motion prediction model deployment, as illustrated in Fig. 2. In the adaptation phase, test data is continuously streamed at each stage from various environments, as depicted in Fig. 6. The incoming data is divided into adaptation (50 samples including samples from the memory buffer) and evaluation (20 samples) sets, wherein the model undergoes updates based on adaptation data and is subsequently evaluated on separate data not utilized in adaptation training.

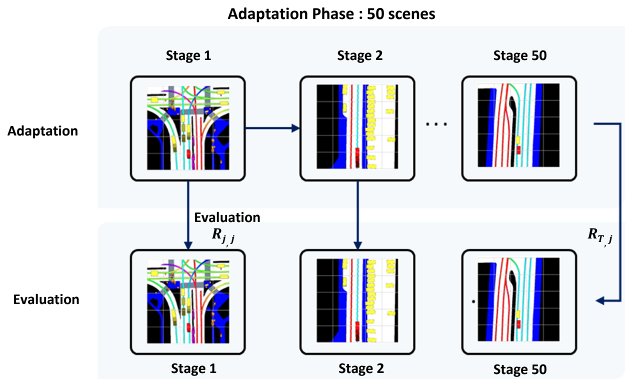

Figure 6. Continual model update scenario in the adaptation phase.

Metrics

- $R_{i,j}$ is the accuracy of the model at stage T_j after observing the last sample from stage T_i.
- Retained accuracy (RA): The average accuracy at stage T is then defined as, $A_T = \frac{1}{T}\sum_{j=1}^{T} R_{T,j}$.
- Backward-transfer and interference (BTI) : Forgetting, BTI = $\frac{1}{T}\sum_{j=1}^{T} R_{T,j} - R_{j,j}$.

We adopt average displacement error(ADE) in motion prediction as the model accuracy metric. In Fig. 6, R_{ij} signifies the ADE performance of the model at the jth stage, following adaptation updates made in the ith stage. We utilize the BTI metric to assess the average accuracy shift at each stage, from initial learning to the conclusion of training. Because ADE stands for path prediction error, smaller values of RA and BTI serve as indicators of a successful continual model update process.

The proposed method is compared and evaluated using the following baselines in Fig. 7. The baselines are listed below:

A) Joint training: The model is trained using all available data (dataset from training and target tasks). The pre-training and adaptation processes are not separate.

B) Pre-training (standard-supervised), C) Pre-training (OML) [20], and D) Pre-training (MAML-Rep) [20]: The model is trained in the pre-training phase using each training method. Then it is continually fine-tuned at each stage using the incoming data sequence from the adaptation phase. MAML-Rep is also a representation learning method. Unlike OML, which leverages sequential updates in the inner loop to induce catastrophic forgetting effects, MAML-Rep uses a batch of data for updates in the inner loop.

E) Elastic Weight Consolidation (EWC) [22]: The model is trained in the pre-training phase using EWC. Then, it is continually updated for the incoming sequential stage based on EWC. EWC is a representative regularization-based learning method for continual learning that constrains updating parameters important for previous tasks.

F) Scratch: The model is updated with only data for the incoming sequential stage in the adaptation phase.

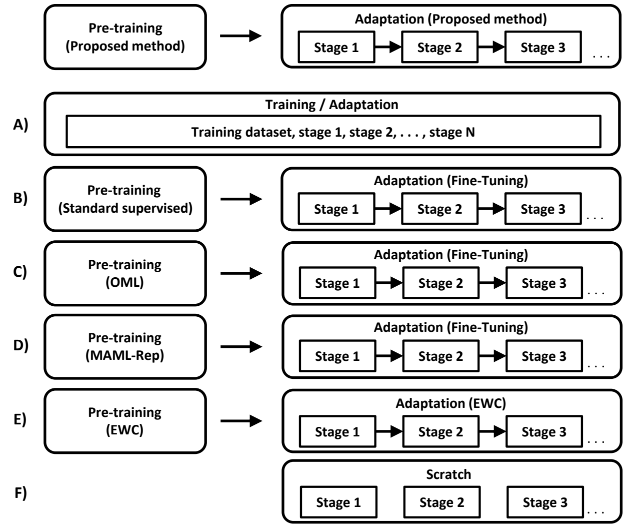

Figure 7. Training process of baselines.

Implementation details. We utilize DenseNet-121 [18] as the backbone for the Representation Learning Network (RLN), while employing Multi-Layer Perceptron (MLP) layers for the Prediction Learning Network (PLN). Notably, our approach remains agnostic to the specific model architecture, ensuring flexibility and applicability across various frameworks. During the pre-training phase, we configure $k = 10$ and $m = 1$, conducting 5 epochs of training for the inner loop with a learning rate of 1e-2. For the outer loop, we set $m = 16$ and conduct 10 epochs of training with a learning rate of 1e-3. In the adaptation phase, the model undergoes training for 10 epochs per stage with a learning rate of 1e-3. We employ the Adam optimizer throughout the training process and train the model with 4 NVIDIA RTX3080ti GPUs. To ensure robustness and reliability of our results, experiments are conducted using five random seeds, with data composition altered at each stage to provide comprehensive evaluation and validate the effectiveness of our approach.

5.2. Evaluation on Continual Update Scenario

Catastrophic forgetting. Table. 1 reports that the pre-trained model based on the proposed learning method shows superior RA and BTI performance than other pre-trained models throughout overall memory buffer sizes. Notably, it represents extremely low catastrophic forgetting when the memory buffer is 40. While traditional learning methods exhibit enhanced performance with larger memory buffer sizes, our proposed approach excels even with smaller ones. The performance continues to improve in

Table 1. Overall adaptation performance analysis. M denotes the memory buffer size.

Method	RA / BTI		
	M=20	M=30	M=40
Joint-training	1.92±0.03		
Standard supervised	2.87±0.11 / 0.60±0.30	2.89±0.52 / 0.49±0.54	2.42±0.08 / 0.16±0.13
OML	2.75±0.03 / 0.54±0.01	2.52±0.29 / 0.31±0.15	2.35±0.06 / 0.14±0.11
EWC	2.88±0.15 / **0.42±0.17**	2.77±0.33 / 0.54±0.22	2.31±0.17 / 0.36±0.76
MAML-Rep	2.79±0.09 / 0.73±0.49	2.54±0.12 / 0.77±0.89	2.36±0.47 / 0.23±0.44
Proposed method	**2.24±0.23** / 0.51±0.14	**2.16±0.31 / 0.24±0.17**	**2.06±0.11 / 0.13±0.09**

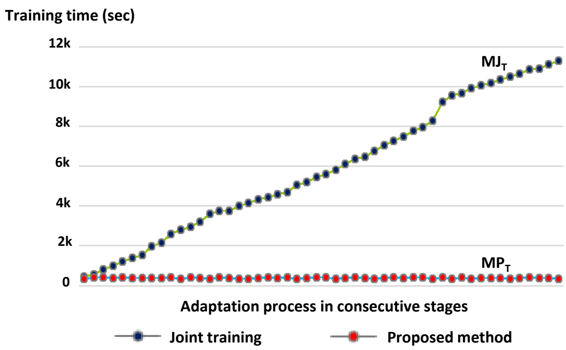

Figure 8. Comparison in training time between joint training and the proposed method.

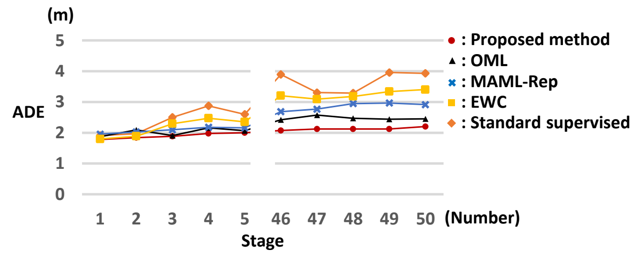

Figure 9. Adaptation performance during continual model updates.

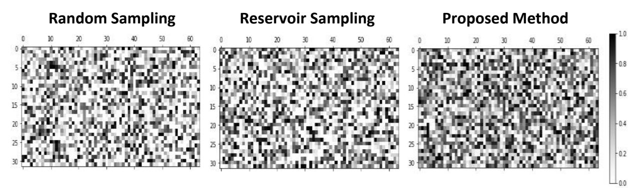

Figure 10. Average activation of the representation vector in the dataset of the memory buffer through each sampling method. Dead neuron percentage: 31%(random sampling), 24%(reservoir sampling), 11%(proposed method)

correlation with the memory buffer size, showcasing the effectiveness of our adaptive sampling method. Through the above test scenarios, we ensured that the methods proposed in the pre-training and adaptation process could cope with continual learning scenarios in the real world.

Adaptation time. Joint training is unsustainable because it requires much resource consumption to collect, store, and process data. We compare the adaptation time of the proposed method with joint training and find that joint training is not feasible in real scenarios. Training time(second) in the adaptation process in each method is defined as model update time until the model is fully trained at each adaptation stage (MU_j: time consumed for the model update using joint training method, MU_p: time consumed for the model update using the proposed method). Fig. 8 shows the difference in model update time of the two methods. It reports that joint training needs more model update time as an adaptation process progresses than the proposed method. Since the motion prediction model used for autonomous driving must be extended to infinitely different environments, we are confident that the proposed method can greatly help the AI model update process of autonomous driving.

Adaptation performance during model update. Leveraging a pre-trained model, we continually updated the model using sample data at each task stage, demonstrating superior adaptation performance. Unlike traditional continual learning challenges, where synthetic datasets provide clear task shifts, the real-world driving data presents blurred boundaries between stages, leading to performance variability. Despite this, our methodology stands out, consistently delivering stable and robust performance, evidenced by a remarkably even trend across the adaptation phase as shown in Fig. 9.

Performance depending on adaptation strategy. In the existing benchmark dataset for developing continual learning methods, the data distribution is given discretely in the data sequence, so there is no concern regarding the sampling method for the memory buffer. However, in the real motion prediction problem, the sampling method to fill the limited memory buffer in an optimal state is necessary because there could be cases where the data distribution is similar depending on driving patterns, even if the scene is different. Therefore, we proposed an adaptive sampling

Table 2. Performance analysis depending on adaptation strategy without pre-trained model. M denotes the memory buffer size.

Adaptation strategy	Memory buffer	RA / BTI		
		M=20	M=30	M=40
Scratch			3.94±1.09 / 1.83±1.10	
Random sampling	✓	3.42±0.51 / 1.44±0.81	3.22±0.42 / 1.29±0.54	3.23±0.51 / 0.97±0.66
Reservoir sampling	✓	2.54±0.25 / 0.96±0.35	2.54±0.08 / 0.56±0.21	2.48±0.14 / 0.45±0.46
Reservoir+adaptive sampling	✓	2.45±0.12 / 0.85±0.18	2.48±0.49 / 0.47±0.43	2.25±0.13 / 0.22±0.14

Table 3. Performance analysis depending on pre-trained model without model update.

Pre-training	RA
Standard supervised	4.96±0.13
OML	3.24±0.19
Proposed method	**2.57±0.09**

method through data distribution shift detection showed better adaptation performance than the existing sampling method. To verify the effectiveness of the proposed adaptive sampling method, we conducted an experiment updating the model with each adaptation strategy in Table. 2 without a pre-trained model. Compared to Scratch without memory replay, memory replay surely improves both RA and BTI. Also, we see that the reservoir sampling method, which stores data from each stage in identical proportion, positively affects RA improvement while reducing BTI than the random sampling method. In addition, instead of taking data in equal proportions at each stage, the proposed method shows improved performance in both RA and BTI than reservoir sampling, which is widely used in continual learning.

5.3. Ablation Study

Effect of the optimal memory buffer retention strategy. The ideal state for the memory buffer is one where the data distribution closely resembles the overall data distribution of the adaptation phase. This alignment can help optimize model updates by ensuring data fidelity and facilitating seamless information integration. We demonstrate the effectiveness of our proposed method by evaluating the extent to which each sampling approach achieves these optimal states. This can be elucidated by examining the mean activation of the representation vector across the dataset stored in the memory buffer using each sampling method. The average activation represents a composite of representations from all data within the memory buffer. We illustrate it by normalizing and visually flattening it in two dimensions to improve clarity, as depicted in Fig. 10 Our demonstration of data distribution coverage within the memory buffer, depicted through the average activation map, provides valuable insights into the efficiency of various sampling methods. The proposed sampling method leverages a broader representation space in average activation compared to conventional random and reservoir sampling methods. Notably, our proposed method exhibits a significantly lower dead neuron percentage of (11%), outperforming reservoir sampling (24%) and random sampling (31%). This expansion encompasses a wider range of data distributions, ensuring the optimal state of the entire memory buffer.

Generalization performance in the pre-training phase. Each learning method demonstrates varying performance levels during the adaptation phase. Table. 3 reports that our proposed method, even without additional updates, achieves a remarkable RA result comparable to joint training, which entails training all data from the adaptation stage. Compared to OML, the proposed method, which leverages a modulation network for explicitly learning sparse representation, significantly enhances the performance in environments with complex input data distribution. This outcome can be attributed to sparse representation, with instance sparsity percentages for each learning method measured at 15%, 22%, and 58%, respectively (proposed method, OML, standard supervised).

6. Conclusion

This study addresses the critical challenge of scaling and deploying motion prediction models across diverse driving environments. Given the myriad driving scenarios shaped by unique road and traffic conditions, achieving scalability is crucial for the commercial viability of autonomous driving. Thus, our proposed learning framework is indispensable for these models' sustainable development and deployment, marking a significant stride toward commercialization.

7. Acknowledgment

This work was supported by Korea Internet & Security Agency (KISA) grant funded by the Korea government (PIPC) (No. 1781000009) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MIST)(2022R1A2C200494412).

References

[1] Mohammadhossein Bahari, Saeed Saadatnejad, Ahmad Rahimi, Mohammad Shaverdikondori, Amir Hossein Shahidzadeh, Seyed-Mohsen Moosavi-Dezfooli, and Alexandre Alahi. Vehicle trajectory prediction works, but not everywhere. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17123–17133, 2022. 1, 2

[2] Ershad Banijamali, Mohsen Rohani, Elmira Amirloo, Jun Luo, and Pascal Poupart. Prediction by anticipation: An action-conditional prediction method based on interaction learning, 2020. 2

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 5

[4] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *Conference on Robot Learning*, pages 947–956. PMLR, 2018. 2

[5] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018. 3

[6] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, P Dokania, P Torr, and M Ranzato. Continual learning with tiny episodic memories. In *Workshop on Multi-Task and Lifelong Reinforcement Learning*, 2019. 3

[7] Corinna Cortes, Xavier Gonzalvo, Vitaly Kuznetsov, Mehryar Mohri, and Scott Yang. Adanet: Adaptive structural learning of artificial neural networks. In *International conference on machine learning*, pages 874–883. PMLR, 2017. 3

[8] Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, Tsung-Han Lin, Thi Nguyen, Tzu-Kuo Huang, Jeff Schneider, and Nemanja Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2090–2096. IEEE, 2019. 2

[9] Shengzhe Dai, Li Li, and Zhiheng Li. Modeling vehicle interactions via modified lstm models for trajectory prediction. *IEEE Access*, 7:38287–38296, 2019. 2

[10] Nachiket Deo and Mohan M Trivedi. Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1179–1184. IEEE, 2018.

[11] Nachiket Deo and Mohan M Trivedi. Convolutional social pooling for vehicle trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1468–1476, 2018. 2

[12] Frederik Diehl, Thomas Brunner, Michael Truong Le, and Alois Knoll. Graph neural networks for modelling traffic participant interaction. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 695–701. IEEE, 2019. 2

[13] Wenchao Ding, Jing Chen, and Shaojie Shen. Predicting vehicle behaviors over an extended horizon using behavior interaction network. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8634–8640. IEEE, 2019. 2

[14] Nemanja Djuric, Vladan Radosavljevic, Henggang Cui, Thi Nguyen, Fang-Chieh Chou, Tsung-Han Lin, Nitin Singh, and Jeff Schneider. Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2095–2104, 2020. 2

[15] Timothy J Draelos, Nadine E Miner, Christopher C Lamb, Jonathan A Cox, Craig M Vineyard, Kristofor D Carlson, William M Severa, Conrad D James, and James B Aimone. Neurogenesis deep learning: Extending deep networks to accommodate new classes. In *2017 international joint conference on neural networks (IJCNN)*, pages 526–533. IEEE, 2017. 3

[16] Tyler L Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. Remind your neural network to prevent catastrophic forgetting. In *European Conference on Computer Vision*, pages 466–483. Springer, 2020. 3

[17] Geoffrey E Hinton and David C Plaut. Using fast weights to deblur old memories. In *Proceedings of the ninth annual conference of the Cognitive Science Society*, pages 177–186, 1987. 3

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 6

[19] Boris Ivanovic, James Harrison, and Marco Pavone. Expanding the deployment envelope of behavior prediction via adaptive meta-learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7786–7793. IEEE, 2023. 2, 3

[20] Khurram Javed and Martha White. Meta-learning representations for continual learning. *Advances in neural information processing systems*, 32, 2019. 4, 6

[21] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 3

[22] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 3, 6

[23] Kim-Hung Li. Reservoir-sampling algorithms of time complexity o (n (1+ log (n/n))). *ACM Transactions on Mathematical Software (TOMS)*, 20(4):481–493, 1994. 5

[24] Xin Li, Xiaowen Ying, and Mooi Choo Chuah. Grip: Graph-based interaction-aware trajectory prediction. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3960–3966. IEEE, 2019. 2

[25] Zhizhong Li and Derek Hoiem. Learning without forgetting.

IEEE transactions on pattern analysis and machine intelligence, 40(12):2935–2947, 2017. 1, 3

[26] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017. 3

[27] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3569–3577, 2018. 2

[28] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009. 1

[29] Seong Hyeon Park, ByeongDo Kim, Chang Mook Kang, Chung Choo Chung, and Jun Won Choi. Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1672–1678. IEEE, 2018. 2

[30] Tung Phan-Minh, Elena Corina Grigore, Freddy A Boulton, Oscar Beijbom, and Eric M Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14074–14083, 2020. 2

[31] Derek J Phillips, Tim A Wheeler, and Mykel J Kochenderfer. Generalizable intention prediction of human drivers at intersections. In *2017 IEEE intelligent vehicles symposium (IV)*, pages 1665–1670. IEEE, 2017. 2

[32] SA Rebuffi, A Kolesnikov, G Sperl, CH Lampert, and icarl. Incremental classifier and representation learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5533–5542, . 3

[33] SA Rebuffi, A Kolesnikov, G Sperl, CH Lampert, and icarl. Incremental classifier and representation learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5533–5542, . 3

[34] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 3

[35] Marcel Schreiber, Stefan Hoermann, and Klaus Dietmayer. Long-term occupancy grid prediction using recurrent neural networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9299–9305. IEEE, 2019. 2

[36] Haoran Song, Wenchao Ding, Yuxuan Chen, Shaojie Shen, Michael Yu Wang, and Qifeng Chen. Pip: Planning-informed trajectory prediction for autonomous driving. In *European Conference on Computer Vision*, pages 598–614. Springer, 2020. 2

[37] Rishabh Tiwari, Krishnateja Killamsetty, Rishabh Iyer, and Pradeep Shenoy. Gcr: Gradient coreset based replay buffer selection for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 99–108, 2022. 3

[38] Ekaterina Tolstaya, Reza Mahjourian, Carlton Downey, Balakrishnan Varadarajan, Benjamin Sapp, and Dragomir Anguelov. Identifying driver interactions via conditional behavior prediction, 2021. 2

[39] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. In *2018 IEEE international Conference on Robotics and Automation (ICRA)*, pages 4601–4607. IEEE, 2018. 2

[40] Jingke Wang, Tengju Ye, Ziqing Gu, and Junbo Chen. Ltp: Lane-based trajectory prediction for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17134–17142, 2022. 2

[41] Letian Wang, Yeping Hu, Liting Sun, Wei Zhan, Masayoshi Tomizuka, and Changliu Liu. Transferable and adaptable driving behavior prediction. *arXiv preprint arXiv:2202.05140*, 2022. 2, 3

[42] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *arXiv preprint arXiv:2302.00487*, 2023. 2

[43] Zhenyi Wang, Li Shen, Tiehang Duan, Donglin Zhan, Le Fang, and Mingchen Gao. Learning to learn and remember super long multi-domain task sequence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7982–7992, 2022. 3

[44] Zhenyi Wang, Li Shen, Tiehang Duan, Donglin Zhan, Le Fang, and Mingchen Gao. Learning to learn and remember super long multi-domain task sequence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7982–7992, 2022. 3

[45] Tianjun Xiao, Jiaxing Zhang, Kuiyuan Yang, Yuxin Peng, and Zheng Zhang. Error-driven incremental learning in deep convolutional neural network for large-scale image classification. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 177–186, 2014. 3

[46] Long Xin, Pin Wang, Ching-Yao Chan, Jianyu Chen, Shengbo Eben Li, and Bo Cheng. Intention-aware long horizon trajectory prediction of surrounding vehicles using dual lstm networks. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1441–1446. IEEE, 2018. 2

[47] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017. 3

[48] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017. 3

[49] Tianyang Zhao, Yifei Xu, Mathew Monfort, Wongun Choi, Chris Baker, Yibiao Zhao, Yizhou Wang, and Ying Nian Wu. Multi-agent tensor fusion for contextual trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12126–12134, 2019. 2

[50] Guanyu Zhou, Kihyuk Sohn, and Honglak Lee. Online incremental feature learning with denoising autoencoders. In *Artificial intelligence and statistics*, pages 1453–1461. PMLR, 2012. 3

[51] Alex Zyner, Stewart Worrall, James Ward, and Eduardo Nebot. Long short term memory for driver intent prediction. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1484–1489. IEEE, 2017. 2

[52] Alex Zyner, Stewart Worrall, and Eduardo Nebot. A recurrent neural network solution for predicting driver intention at unsignalized intersections. *IEEE Robotics and Automation Letters*, 3(3):1759–1764, 2018.

[53] Alex Zyner, Stewart Worrall, and Eduardo Nebot. Naturalistic driver intention and path prediction using recurrent neural networks. *IEEE transactions on intelligent transportation systems*, 21(4):1584–1594, 2019. 2