

Complementary Learning System Empowers Online Continual Learning of Vehicle Motion Forecasting in Smart Cities

Zirui Li¹, Yunlong Lin¹, Guodong Du¹, Xiaocong Zhao², Cheng Gong¹, Chen Lv³, Chao Lu^{1,*}, and Jianwei Gong^{1,*}

¹The School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China.

²The Key Laboratory of Road and Traffic Engineering, Ministry of Education, Tongji University, Shanghai, China.

³The School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore.

*Corresponding authors: chaolu@bit.edu.cn (C. Lu) and gongjianwei@bit.edu.cn (J. Gong)

ABSTRACT

Artificial intelligence underpins most smart city services, yet deep neural network (DNN) that forecasts vehicle motion still struggle with catastrophic forgetting, the loss of earlier knowledge when models are updated. Conventional fixes enlarge the training set or replay past data, but these strategies incur high data collection costs, sample inefficiently and fail to balance long- and short-term experience, leaving them short of human-like continual learning. Here we introduce Dual-LS, a task-free, online continual learning paradigm for DNN-based motion forecasting that is inspired by the complementary learning system of the human brain. Dual-LS pairs two synergistic memory rehearsal replay mechanisms to accelerate experience retrieval while dynamically coordinating long-term and short-term knowledge representations. Tests on naturalistic data spanning three countries, over 772,000 vehicles and cumulative testing mileage of 11,187 km show that Dual-LS mitigates catastrophic forgetting by up to 74.31% and reduces computational resource demand by up to 94.02%, markedly boosting predictive stability in vehicle motion forecasting without inflating data requirements. Meanwhile, it endows DNN-based vehicle motion forecasting with computation efficient and human-like continual learning adaptability fit for smart cities.

Keywords: Smart cities, vehicle motion forecasting, continual learning, complementary learning system.

Introduction

With advances in deep learning and high-performance computing, artificial intelligence (AI) has achieved remarkable breakthroughs^{1–3}. To boost operational efficiency and reduce human workload, many data-driven algorithms have been embedded in smart cities^{4–7}. Vehicle motion forecasting⁸, as a typical technology of intelligent vehicles application in smart cities, plays an important role in alleviating traffic congestion⁹, protecting the safety of vulnerable road users¹⁰, improving driving comfort¹¹ and reducing accident rates¹². Because naturalistic scenarios evolve rapidly and future conditions are difficult to predict, these systems must learn and adapt continuously throughout their lifecycles¹³. Nowadays, richer collected data and stronger processing capabilities allow most AI modules to be trained on ever-larger datasets to improve performance. This trend, however, raises two challenges: storing and maintaining vast sensor archives is costly, and collecting all possible cases in a single effort is unrealistic. A common workaround is to train an initial model and then update it incrementally as new data arrive¹⁴. From the perspective of DNN, such continual updates can trigger catastrophic forgetting, the gradual loss of proficiency on previously learned tasks¹⁵.

To maintain the memory stability of DNN and mitigate catastrophic forgetting under real-world data streams, researchers have advanced continual learning (CL) techniques^{16,17}. CL seeks to endow models with the dual capacity to assimilate new tasks while retaining competence on previously learned ones during continuous updates. Within the CL framework, assorted paradigms and algorithms have emerged, each tailored to specific assumptions, such as whether task labels are supplied, how much memory can be stored, the desired learning efficiency, and the number of classes involved¹⁸. These assumptions give rise to replay-based, generative-based and dynamic architecture strategies¹⁸. Most current CL methods, however, are validated only on simplified benchmarks¹⁹, restricting their usefulness in safety-critical transportation systems¹³. In self-driving vehicles (SVs), DNN-based motion forecasting, which predicts the future trajectories of surrounding agents and feeds this information

⁰Under formal peer review at Nature Computational Science

directly into the planning stack, is a linchpin of overall safety and performance. Enhancing the CL capacity of this component can preserve prediction fidelity in the face of non-stationary traffic patterns, thereby protecting vulnerable road users, improving traffic flow and reducing conflict incidents¹².

However, there are two major shortcomings in current attempts. First, there is a high reliance on task identifiers to indicate boundaries between tasks, which are hard to define accurately or meaningfully; second, memory samples from the learned tasks are typically selected randomly, lacking a refined maintenance and efficient utilization strategy. In this article, to address the first issue, we propose a modified paradigm for DNN-based modules with online task-free continual learning (OTFCL) ability. This paradigm consists of four key components: data stream, memory storage, model update strategies, and performance evaluation. Similar to the learning process of the human brain, OTFCL is required to integrate new experiences with prior knowledge to update its cognitive abilities and form memories. When the model meets cases it has learned, the performance can reflect whether catastrophic forgetting happens. Specifically, the characteristics of each component in the OTFCL paradigm are as follows:

- **Data stream:** The data stream in CL is a dynamic, evolving, and potentially non-repetitive flow of information. The DNN-based models need to adapt to the new distributions in the data stream while retaining old knowledge from past information.
- **Memory storage:** The memory storage refers to the mechanisms used to retain and recall knowledge from previous tasks or experiences to mitigate catastrophic forgetting. It plays a critical role in CL as models are typically limited in terms of how much past data they can store or replay.
- **Model update strategies:** The strategy is the algorithm for weight update in DNN by complying with the constraints in the memory limit, the learning setting, and the related assumption. The goal of the strategy is to improve the performance expectations within OTFCL.
- **Performance evaluation:** Besides the final accuracy after the whole learning process, the evaluation metrics need to measure the level of forgetting, which is also termed memory stability.

In the OTFCL paradigm, any method that can leverage the data stream to update models can be incorporated, such as continually training without recalling past samples and updating the model with human feedback. In the fields of computer vision and reinforcement learning, several algorithms have been proposed¹⁸, including gradient projection, rehearsal replay, generative replay, and the integration of these approaches. Experimentally, we found that the rehearsal replay-based methods performed better than others²⁰. However, we also identified that during memory replay, the reliance on random sampling with equal probability lacks effective memory management, which is the second shortcoming. It can result in important samples, that may be critical for mitigating forgetting, not being selected during the historical sample retrieval process. Additionally, during the process of model updates, all samples are treated equally, without distinguishing between short-term quick memories and long-term foundational memories in the algorithmic design. Inspired by the complex neural connections in the human brain²¹, numerous DNN architectures have been designed to tackle various tasks¹⁸. For CL, the recent trend in neuroscience has demonstrated that the human brain's memory maintenance and knowledge accumulation align with the Complementary Learning System (CLS) theory^{22,23}. According to the CLS theory²³, effective learning requires two complementary systems: one, located in the neocortex, serves as the basis for the gradual acquisition of structured knowledge about the environment, while the other, centered on the hippocampus, allows rapid learning of the specifics of individual items and experiences. Inspired by the biological mechanism of CLS theory, we propose a novel algorithm with **Dual** memory buffer and **Long-Short term learning strategy** under the OTFCL paradigm, which is termed Dual-LS.

- **Dual memory buffer:** In the replay-based CL, a new memory buffer with two kinds of samples from reservoir sampling and gradient-based diversity sampling²⁴, which serve for equal probability in the data stream and are immune to the imbalanced data stream, respectively.
- **Long-short term learning strategy:** Inspired by the CLS theory in the human brain²², a new model update strategy is developed to trade-off the long-term consolidate and short-term rapid knowledge, which mimics the interplay mechanism between hippocampus and neocortex.

To demonstrate the effectiveness of the designed paradigm and the proposed algorithm for CL, the vehicle motion forecasting⁸ is selected as the learning objective, which is a typical, challenging and safe-related task for SV. It is to predict the multi-modal behaviours of the target vehicle in the environment with multiple interactive vehicles. Accurate prediction of the target vehicle's future behavior can support the SV's reasonable decision-making and safe navigation. For the OTFCL paradigm, the data stream is created based on the worldwide naturalistic driving data with over 772,000 vehicles and cumulative

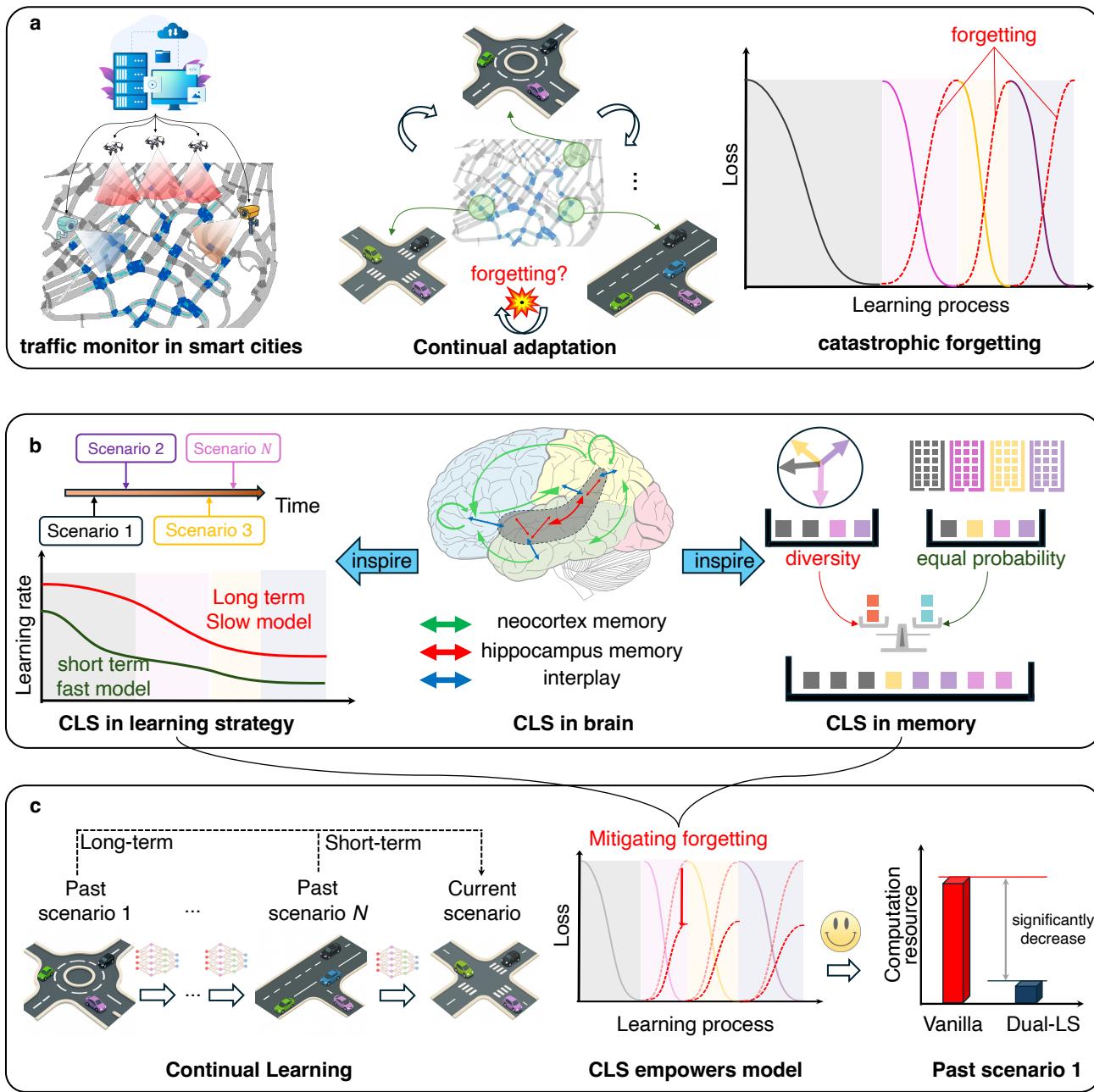


Figure 1. The proposed CLS-inspired Dual-LS algorithm. **a**, The DNN model is sequentially trained by the samples from each scenario and trying to adapt to the newly encounter one. It may suffer from the catastrophic forgetting when meets the previously learnt scenario. Left: the traffic monitor system in smart cities; Middle: the DNN-based system continual adaptation to various traffic scenarios; Right: the catastrophic forgetting occurs in the learning process. **b**, The CLS in brain learning strategy and replay memory buffer. Left: The long-short term learning strategy with long-term memory slow model and short-term memory fast model. The cooperation of these two models mimics the interplay mechanism between hippocampus and neocortex. Middle: the CLS between neocortex and hippocampus in the human brain. Right: the dual memory buffer with two kinds of samples from reservoir sampling and gradient-based diversity sampling is designed, which serves for a better trade-off between equal probability-based and diversity enhanced-based sampling strategies. **c**, The synthesis of dual memory and the long-short term learning strategy empowers online continual learning. The catastrophic forgetting is mitigated and the demand of computational resource is reduced.

testing mileage of 11,187 km. The dataset consists of multiple interactive traffic scenarios, such as the highway on-ramp,

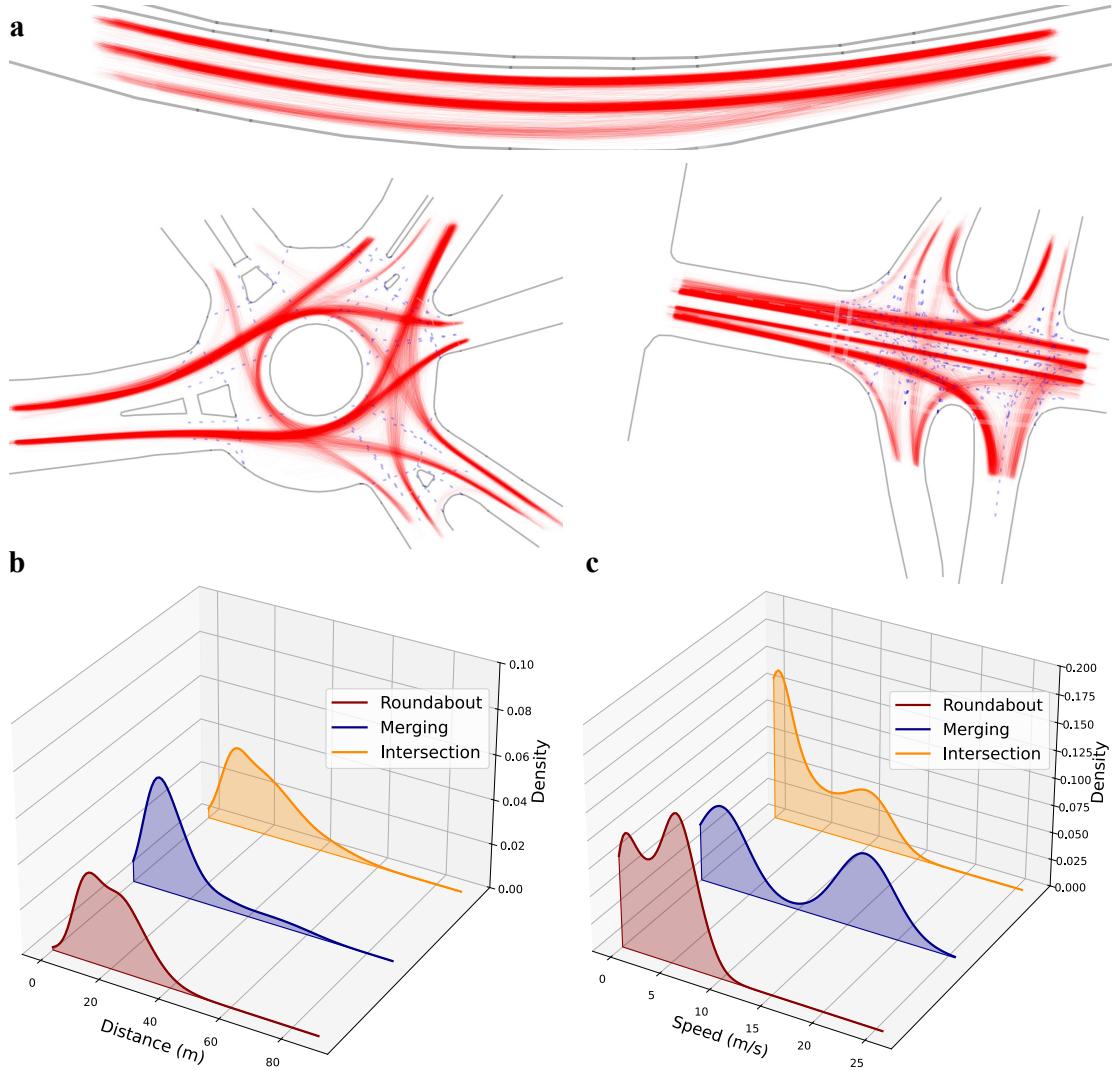


Figure 2. The illustration of the used dataset³⁰. Three scenarios (roundabout, merging and intersection) are selected for comparison. **a**, The visualization of map and all trajectories in the training set. **b**, The distributions of distance between the target vehicle and surrounding vehicles. **c**, The distributions of distance between the vehicle speed. The difference in trajectories, relative distance and speed indicates that these three scenarios are distinguished with each other.

intersections, roundabouts, etc. Several metrics are developed to quantify CL performance in terms of adaptability to new tasks and the ability to mitigate catastrophic forgetting. Comparative experiments with multiple baseline methods demonstrated that the proposed method exhibits excellent performance in vehicle motion forecasting. From a CL perspective, Dual-LS can maintain memory stability and learning efficiency, simultaneously. To the best of our knowledge, this is the first systematic CL paradigm for vehicle motion forecasting in smart cities. Additionally, the proposed Dual-LS provides an outlook for biological intelligence-inspired CL algorithms. In this article, the paradigm, algorithm, and validation pipeline are versatile, and not only applicable to DNN-based modules in SVs but also to other data-driven systems^{25–27}. For example, they could be applied to the self-evolving system^{9,28} or to evaluate the CL capability of smart cities²⁹.

Results

Dataset description

In this paper, the DNN of smart cities is specified as the motion forecasting of SVs. To effectively validate the performance of continual learning, the INTERACTION dataset is employed³⁰, which clearly categorizes data collected by unmanned aerial vehicles across multiple distinct scenarios. It covers several countries and includes diverse scenarios such as intersections,

roundabouts, and highways. Compared to datasets that do not differentiate between scenarios (e.g., Waymo³¹), this is particularly beneficial for designing experiments related to DIL¹⁶. Although the proposed Dual-LS is task-free, meaning that it does not require the information of task boundary, the clear segmentation of tasks aids in standardizing tests and ensuring fair comparisons. According to the order in which the data of different scenarios appear in the data stream, they are outlined as: intersection#1 (MA), roundabout#1 (FT), roundabout#2 (LN), Merging#1 (ZS2), roundabout#3 (OF), intersection#2 (EP0), intersection#3 (GL), Merging#2 (ZS0). The number of samples in each scenarios are provided in the Supplementary Material (Section 1). Fig. 2a illustrates the differences in feature distribution across different scenarios, using scenes ZS2, FT, and GL as examples. As shown in Fig. 2b and Fig. 2c, the distributions of relative distance and vehicle speed vary significantly across different scenarios. The statistical results of remaining scenarios are listed in the Supplementary Material (Section 1).

Mitigating memory and computational resource demand

Continual learning algorithms are valuable because they maintain motion forecasting performance while restraining memory and compute demands. Fig. 3 depicts prediction error alongside relative consumption resource. The joint training (JT) is trained on all data at once, whereas Vanilla proceeds training sequentially from task 1 to 8. For task 1 (Fig. 3a and b), Dual-LS cuts the peak memory requirement by 94.02% compared with JT, without materially affecting prediction error. Fig. 3a plots the cumulative volume of data processed over the entire training procedure, a quantity that is broadly proportional to the peak memory footprint required. Compared to Vanilla, Dual-LS lowers both prediction error and computational cost by 96.54% and 34.08%, respectively, demonstrating that it guards past task 1 against catastrophic forgetting even under a markedly slimmer computational budget. Similar trends hold for task 3 (Fig. 3c and d). Collectively, these results show that Dual-LS simultaneously mitigates catastrophic forgetting and curbs memory and compute overhead.

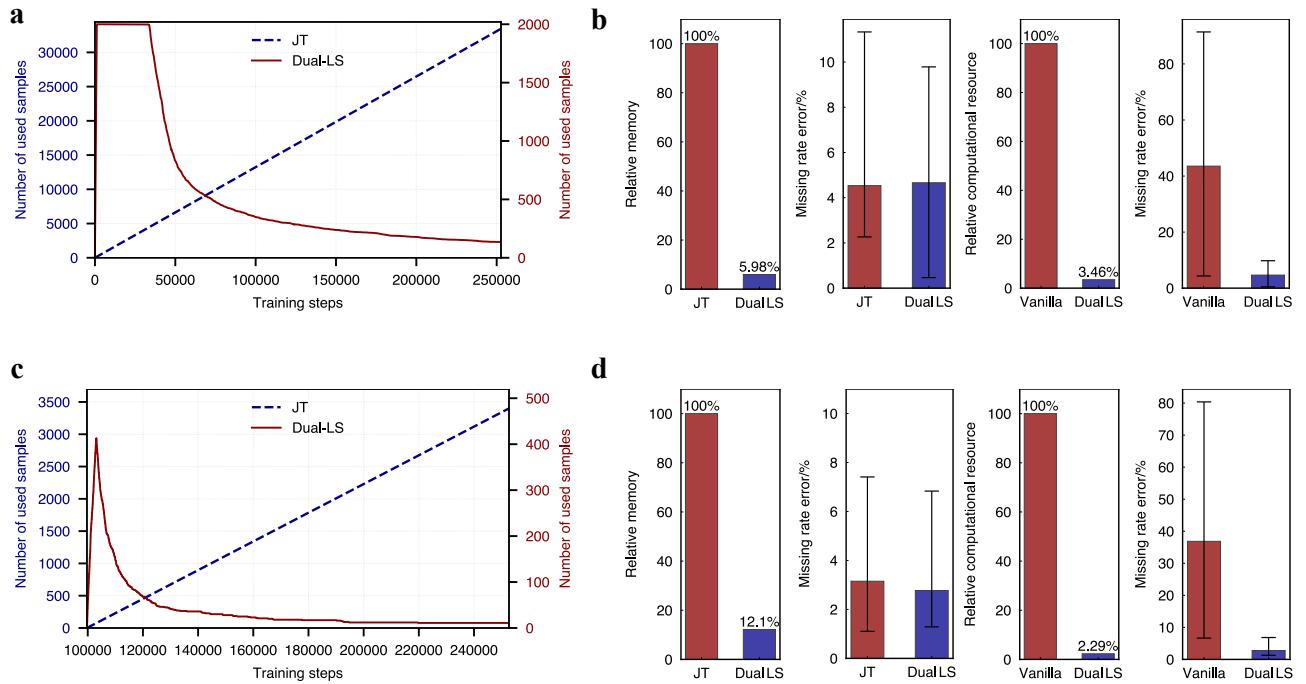


Figure 3. The overall performance of Dual-LS, Vanilla and joint training in mitigating catastrophic forgetting, memory and computational resource demand. **a** and **c**, the cumulative volume of data processed over the entire training procedure for Task 1 Task 3, respectively. **b** and **d** (from left to right), the relative memory used between JT and Dual-LS, the missing rate error (%) between JT and Dual-LS, the relative computational resource used between Vanilla and Dual-LS, the missing rate error (%) between Vanilla and Dual-LS. The error bars represent the standard error of 10 runs.

Mitigating catastrophic forgetting

Fig. 4 presents the overall results of the Dual-LS algorithm compared to all baseline methods. It is important to note that, while Fig. 4 explicitly delineates different tasks, the task index and boundary information were not disclosed to algorithms. This division is solely for facilitating standardized comparative studies. As shown in Fig. 4b, for the four metrics (MR, FDE, MR-BWT, and FDE-BWT), CL-based algorithms outperform non-CL algorithm (Vanilla). This confirms the existence of

catastrophic forgetting and demonstrates that CL algorithms can mitigate it effectively. Joint training represents the theoretical error lower bound, and the experimental results indicate that all CL methods achieve performance closer to joint training than Vanilla. After learning all 8 tasks, both non-CL and CL algorithms exhibit higher prediction errors on Task 4 and Task 5. This may be due to the significant differences between these two tasks and the others. Among all CL methods, the Dual-LS algorithm achieves lower prediction errors compared to the other three CL algorithms (A-GEM³², GSS²⁴, and DER²⁰), with performance closer to joint training, the theoretical lower bound. This highlights the potential of the CLS theory in biological intelligence to enhance the predictive performance of CL models. In Fig. 4b, after all algorithms complete Task 8, the performance on this recently learned task is comparable, with low errors across the board. This indicates that all algorithms adapt well to the current task. However, CL algorithms uniquely maintain adaptability to the current task while alleviating catastrophic forgetting of past tasks.

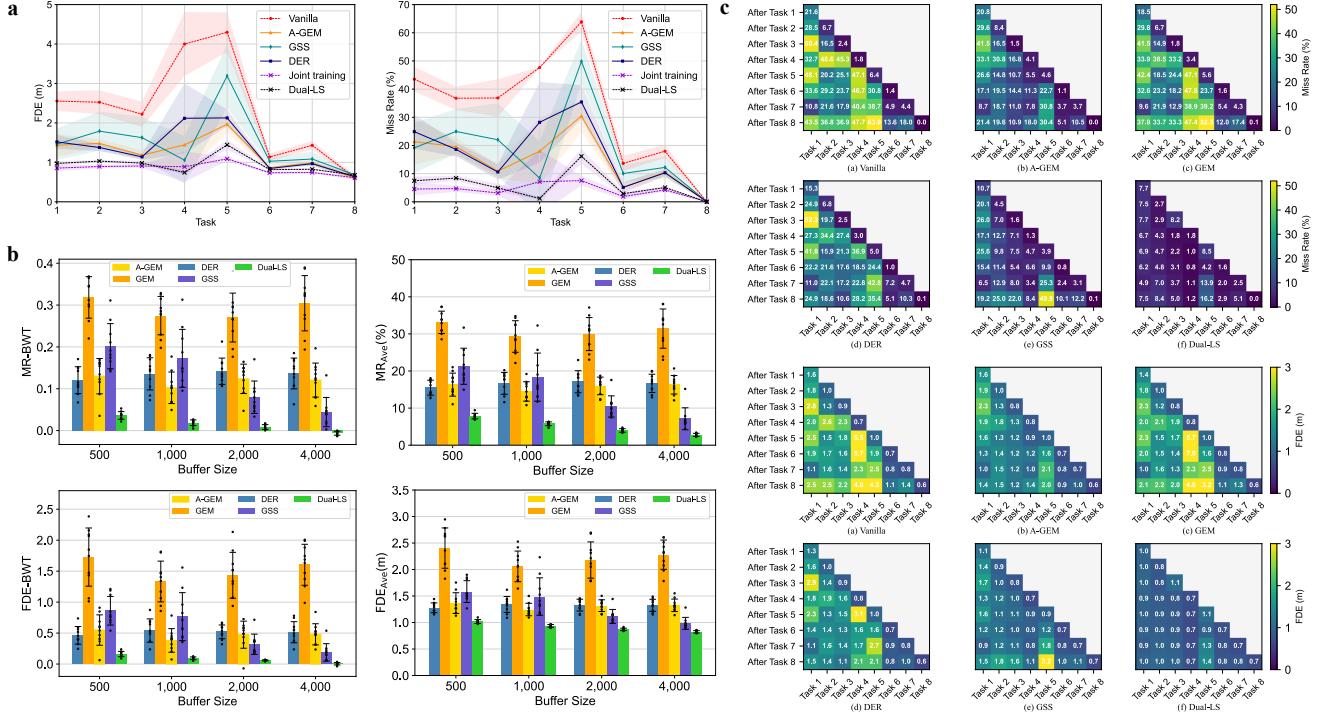


Figure 4. The overall results of Dual-LS and other baseline methods in mitigating the catastrophic forgetting. **a**, The quantification of catastrophic forgetting in FDE and MR (Total memory buffer size: 1000). The x-axis represents that each model is trained from Task 1 to Task 8. The y-axis is the metrics. The transparent area is the standard deviation of 10 repeated experiments. **b**, The comparison of all methods in different total memory buffer size (500, 1000, 2000 and 4000) with several metrics (MR-BWT, FDE-BWT, MR-AVE, FDE- AVE). The error bars denote the standard deviation of 10 repeated experiments, while scatters represent each experiment's results. **c**, The matrix of errors (FDE and MR) in the process of CL (Total memory buffer size: 1000). The $(i, j)^{\text{th}}$ in the matrix represents the error when model is trained after Task i and then tested in Task j . The error matrix reflects the detailed variation in memory stability.

Similar to the memory replay mechanism in human learning, buffer memory replay serves as the foundation of the proposed OTFCL paradigm, and its effectiveness is influenced by the buffer size. Fig. 4a illustrates the average performance of all CL algorithms under four different buffer sizes. As shown in the Fig. 4b, Dual-LS outperforms all other CL methods and achieves optimal results across different experimental settings. Additionally, the results for GSS and Dual-LS demonstrate that increasing the buffer size enhances model performance. This phenomenon aligns with observations in human learning and memory processes: having a larger memory capacity reduces the forgetting of previously acquired knowledge. This indicates that, compared to A-GEM and DER, both GSS and Dual-LS can effectively utilize the memory buffer. Each bar in Fig. 4b not only presents the average results but also includes the standard deviation from 10 repeated experiments. Fig. 4a shows that the Dual-LS algorithm exhibits more stable predictive performance, achieving both lower average error and smaller standard deviation.

To better capture the dynamics of CL algorithms across sequential tasks, Fig. 4c illustrates the performance of models on all past tasks' test sets after completing each new task. Overall, it can be observed that with the completion of each new task, the

performance on past tasks deteriorates for all algorithms, which reflects the phenomenon of catastrophic forgetting. However, compared to the vanilla algorithm, all five CL algorithms mitigate this issue to some extent. From the stepwise plots in Fig. 4c, it is evident that after learning each new task, Dual-LS consistently achieves better and more stable performance on past tasks compared to A-GEM, GEM, GSS, and DER in most cases. More results of comparative results are shown in Supplementary Material (Section 5).

The effect of complementary learning system

The above analysis primarily focuses on the overall performance improvements brought by the CLS-inspired algorithm to CL. This section delves into the mechanisms behind how Dual-LS embodies principles of biological intelligence.

Fig. 5a illustrates the distribution of samples from different past tasks in two memory replay buffers when the buffer size is set to 1,000. The reservoir sampling algorithm (Fig. 5a bottom) ensures that the selection probability of all samples is unaffected by the sequence of data in the stream. Consequently, the proportion of samples from each past task in the memory buffer is relatively balanced and aligns with the proportion of training samples in each past scenario. The detailed process of reservoir sampling strategy for \mathcal{M}_R is depicted in Algorithm 1 in **Extended Data**. Initially, the buffer is empty. As new samples are observed, they are stored in the buffer. Once the buffer is full, newly encountered samples may replace one of the existing samples. Notably, the probability of a sample being stored is intended to be uniform across all samples in the data stream. However, the total length of the data stream, $|\mathcal{D}|$, is unknown, as the samples arrive incrementally. To realize this equiprobable selection from the data stream, we applied reservoir sampling. As depicted in Fig. 6a, for every $n > |\mathcal{M}_R|$, the n^{th} observed sample is selected as a candidate with a probability of $|\mathcal{M}_R|/n$. Once the n^{th} sample is chosen as the candidate, one of the previously stored samples is randomly replaced with this candidate, with the replacement probability being $1/|\mathcal{M}_R|$. Consequently, each sample in \mathcal{D} has an equal probability of $|\mathcal{M}_R|/|\mathcal{D}|$ of being stored in the buffer.

In contrast, the diversity-aware memory buffer aims to enhance the diversity of samples stored in the buffer. As described in Algorithm 2 in **Extended Data**, the gradient-based diversity sampling aims at maximizing the diversity among the selected samples. The diversity is quantified by the cosine similarity between the loss gradients of samples²⁴, referred to as the similarity score q . A sample in the buffer with a larger value of q is more likely to be replaced by a newly encountered sample. When the buffer is not yet full, each incoming sample \mathbf{d}_n is stored along with its associated similarity score q_n . Once the buffer reaches its capacity, a sample is randomly selected from the buffer to serve as the candidate for replacement. Let the similarity score of the new sample be denoted as q_n , and the similarity score of the candidate sample be q_i . The replacement process follows a Bernoulli distribution, where the probability of replacement is given by $q_i/(q_n + q_i)$. Fig. 6b also shows the mechanism of the gradient-based diversity sampling. As shown in the top of Fig. 5a, as the model progresses from Task 1 to Task 8, the proportion of samples from each past task in the buffer does not strictly match the proportion of training samples in the dataset. Specifically, more training samples from Task 7 and Task 8 are retained in the buffer, whereas fewer samples from Tasks 1 through 6 are preserved by the end of the learning process. This comparison indicates that training samples from Task 7 and Task 8 exhibit higher diversity than those from Tasks 1 to 6. This characteristic facilitates the selection of more representative samples to construct the memory buffer. Fig. 5b shows the sampling results for each past task when the buffer sampling size is set to 8, further validating the conclusions drawn from Fig. 5a. The complementary strengths of the two memory buffers can be combined to enhance model performance. This mechanism resembles the interplay between the hippocampus and neocortex in the human brain, which improves learning efficiency and outcomes.

In addition to the memory buffer design being inspired by biological intelligence, the learning mechanism of Dual-LS also embodies the principles of the CLS theory. Fig. 5c illustrates the interplay between short-term and long-term models. For samples retrieved from the memory buffer, comparing the fast model's error with the slow model's error enables effective selection of the current model's representation in prediction. The detailed mechanism of the interplay between the fast and slow models is provided in Methods. Fig. 5d shows the training loss, indicating that during the training process under the CL paradigm, the model's predictive performance declines after task transitions.

In Fig. 5e, we present a case study to demonstrate the impact of CL on the vehicle motion forecasting. The target vehicle is expected to continue moving along the inner lane of the roundabout and reach the location marked by the green triangle after 3 seconds. This sample is part of the test set from task 5. After the model completes training on all samples from the fifth task, the proposed Dual-LS algorithm achieves a prediction similar to that of the vanilla model. However, as the model continues learning up to Task 8, Dual-LS maintains its predicted target position close to the ground truth. In contrast, the vanilla model produces a completely incorrect prediction, estimating that the target vehicle will leave the roundabout at the next exit. Such a misjudgment could significantly impact the dynamic risk assessment of SV. More case studies are presented in Supplementary Material (Section 4).

In the proposed Dual-LS algorithm, the hyperparameters influence the model's performance. the impact of the number of epochs, the update probability of the long-short term models, and the replay memory loss coefficient on prediction performance are detailed in Supplementary Material (Section 3). Experimental results indicate that increasing the number of epochs does not

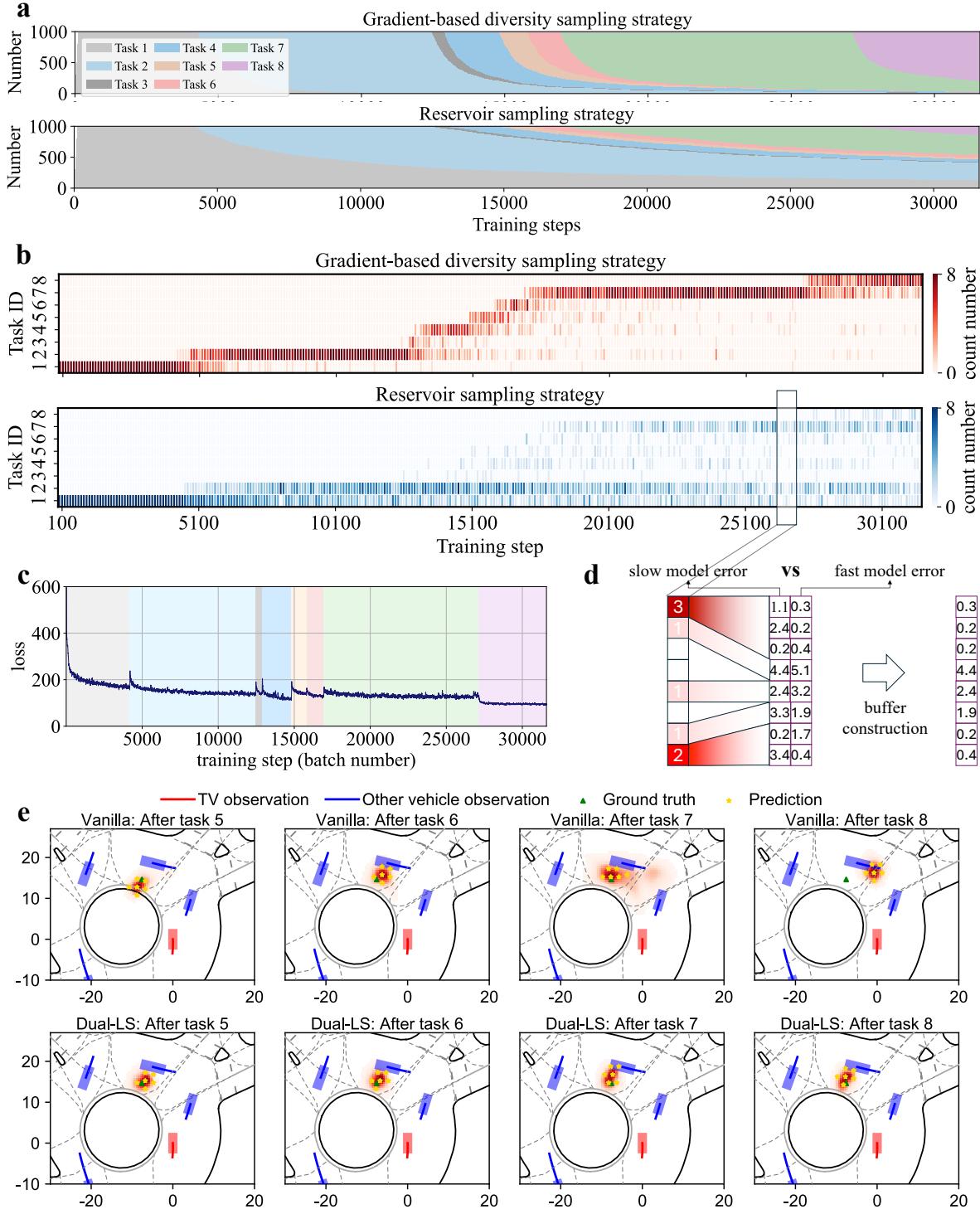


Figure 5. The CLS-inspired learning mechanism of Dual-LS in the CL setting. **a**, The proportion of samples from each past tasks in the memory buffer. It changes after each update of memory buffer. The comparison of two kinds of sampling strategies highlights the significant of CLS in memory buffer design. **b**, The constitution of two memory buffers from each past task. The color represents the number of samples from corresponding past task. **c**, The variation of training loss. Each background color corresponds to a task. **d**, The illustration of long-short term learning strategy. Each sample in the memory buffer has two errors from slow and fast models, respectively. **e**, Case study in the test set from scenario FT (task 2). Top: the vanilla model trained after 5-8 tasks. Down: the Dual-LS trained after 5-8 tasks. In each case, the predictions are represented by six yellow stars in red heatmaps, where the specific predicted postions are sampled based on the predicted heatmap.

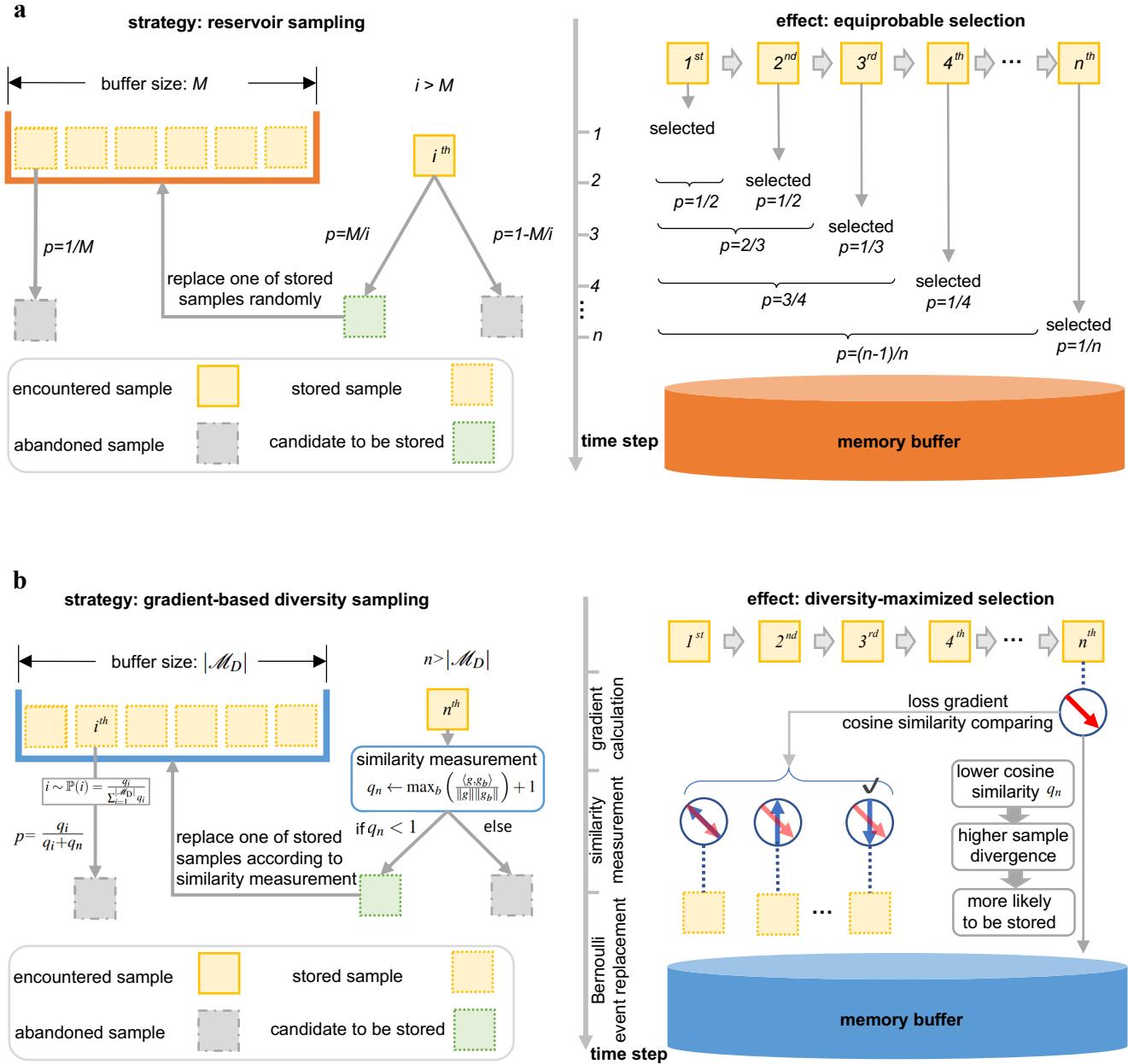


Figure 6. Two memory sampling strategies in Dual-LS. **a**, Reservoir sampling strategy selects each sample from an unlimited online stream with equal probability: First, the buffer is initially empty. As new samples are observed, they are stored in the buffer until its capacity is reached. Once full, newly encountered samples may replace one of the existing samples. The probability of a sample being stored is designed to be uniform across all samples in the data stream. For each $n > |M_R|$, the n^{th} observed sample is selected as a candidate with probability $|M_R|/n$. Once selected, one of the previously stored samples is randomly replaced with this candidate, with the replacement probability being $1/|M_R|$. **b**, Gradient-based diversity sampling strategy aims to maximize the diversity of samples in the memory buffer: The diversity is quantified by the cosine similarity between the loss gradients of samples, referred to as the similarity score, q . The smaller q for a sample represents the relatively higher diversity of this sample to others. Thus, in order to maximize the diversity of stored samples, a sample in the buffer with a larger q value is more likely to be replaced by a newly encountered sample. Let the similarity score of the new sample be denoted as q_n , and the similarity score of the candidate sample be q_i . The replacement follows a Bernoulli distribution, with the probability of replacement given by $q_i/(q_n + q_i)$.

significantly enhance the CL performance of the DNN. Furthermore, no clear patterns emerge between other hyperparameter combinations and evaluation metrics. This suggests that for specific DNN, the selection of relevant CL hyperparameters requires tailored analysis.

Discussion

For both humans and intelligent systems in smart cities, it is crucial to preserve past learning experiences while adapting to dynamic and ever-changing environments. However, the DNN is prone to forget the learned knowledge if CL is not considered. Inspired by the memory replay mechanism in the human brain, this article proposes the OTFCL paradigm to address the issue of catastrophic forgetting during the learning process from continuous data stream. Experimental results indicate that in the motion forecasting for intelligent vehicles, incorporating CL into the pretrain-and-deploy pipeline improves the prediction performance. This enhancement benefits prediction-based risk assessment, contributing to the safety of SVs and other vulnerable road users³³. Furthermore, it promotes the harmony in human-machine interactions³⁴. The proposed paradigm is generalizable and applicable not only to SVs but also to data-driven industrial robots³⁵, service robots³⁶, and other intelligent systems³⁷. Additionally, the OTFCL paradigm imposes strict constraints on the storage capacity of the memory buffer, effectively preventing unlimited data storage from the data stream. This facilitates the development of AI-based smart cities with controllable data storage costs, addressing energy efficiency and environmental sustainability challenges^{38,39}.

Experiments demonstrate that the CLS of the human brain can inspire the design of CL strategies⁴⁰. The integration of dual memory systems can mitigate the issue of sample imbalance across different past tasks in the OTFCL paradigm while ensuring that previously encountered samples are replayed with equal probability. Moreover, the coordination of long-term and short-term learning strategies enables a balance between long-term memory consolidation and the rapid acquisition of short-term knowledge. This design emerges the interplay mechanism between the hippocampus and the neocortex in the human brain. Furthermore, the partitioned architecture of the human brain may offer valuable insights for designing DNN-based CL frameworks.

AI and biological intelligence share common goals in the CL of new knowledge^{21,41,42}, the rapid consolidation of experiences, and the efficient coordination of memory systems. Emerging theories in human brain science, neuroscience, and memory science will continue to inspire the architectural design of DNN^{43–46}. As described in this article, the theory of CLS between the hippocampus and the neocortex reflected in the Dual-LS algorithm, which effectively mitigates catastrophic forgetting. Subsequent work could further explore the integration of CL and collective learning to investigate how individual biological intelligence is transferred, shared, and expanded within society^{47,48}.

Methods

Problem definition

In CL, the DNN-based learner f encounters T tasks $\{\mathcal{T}_1, \dots, \mathcal{T}_T\}$ sequentially. Each task \mathcal{T}_t ($1 \leq t \leq T$) has its learning objective \mathcal{O}_t and contains N_t training samples $\{\mathbf{X}_i, \mathbf{Y}_i\}_{i=1}^{N_t}$, which are drawn from an i.i.d. distribution \mathcal{D}_t . The learning goal within each task can be expressed as:

$$\theta_t^* = \arg \min_{\theta_t} \mathcal{L}_t \quad \text{with} \quad \mathcal{L}_t \triangleq \mathbb{E}_{(\mathbf{X}, \mathbf{Y}) \sim \mathcal{D}_t} (\ell(f_{\theta_t}(\mathbf{X}), \mathbf{Y})) \quad (1)$$

where \mathcal{L}_t is the loss function of task \mathcal{T}_t and \mathbb{E} is the expectation in the calculation of loss. ℓ is each sample's loss between prediction $f_{\theta_t}(\mathbf{X})$ and ground truth \mathbf{Y} . (\mathbf{X}, \mathbf{Y}) is the sample following the distribution \mathcal{D}_t . f_{θ_t} is the learning function of task \mathcal{T}_t equipped with parameters θ_t . Usually, different tasks correspond to distinct distributions,

$$\forall t_1 \neq t_2, \quad \mathcal{D}_{t_1} \neq \mathcal{D}_{t_2} \quad (2)$$

According to the availability of the task identifier in testing and whether the model needs to infer it, the CL can be categorized into three types¹⁶: task-incremental learning (TIL), domain-incremental learning (DIL), and class-incremental learning (Class-IL). In this article, the output representation is fixed for all tasks and the model has no access to the task identifier in testing, so the vehicle motion forecasting is a variant of DIL from image classification. Based on Eq. (1), the sequential learning process from first task \mathcal{T}_1 to the current task \mathcal{T}_{t_c} is formulated as:

$$\theta_{1:t_c}^* = \arg \min_{\theta_{1:t_c}} \sum_{t=1}^{t_c} \mathcal{L}_t \quad \text{with} \quad \mathcal{L}_t \triangleq \mathbb{E}_{(\mathbf{X}, \mathbf{Y}) \sim \mathcal{D}_t} (\ell(f_{\theta}(\mathbf{X}), \mathbf{Y})) \quad (3)$$

where $\theta_{1:t_c}^*$ is the optimized weights of DNN until the current task \mathcal{T}_{t_c} and $t_c \in \{1, \dots, T\}$ is the index of current task. In Eq. (3), obtaining the optimal weights requires access to the dataset corresponding to each task that conforms to distribution

$\{\mathcal{D}_1, \dots, \mathcal{D}_{t_c-1}\}$. However, in CL, it is not possible to access all the original data from previously experienced tasks, meaning that not all samples following distribution $\{\mathcal{D}_1, \dots, \mathcal{D}_{t_c-1}\}$ can be obtained. Ideally, the optimization process in Eq. (3) aims to mimic the prediction performance on past samples while fitting the samples in the current task \mathcal{T}_{t_c} . In this way, the model is designed to approximate the best and original prediction for past samples. Therefore, the learning process is converted into:

$$\theta_{1:t_c}^* = \arg \min_{\theta_{1:t_c}} \mathcal{L}_{t_c} + \alpha \sum_{t=1}^{t_c-1} \mathbb{E}_{(\mathbf{X}, \mathbf{Y}) \sim \mathcal{D}_t} \left[D_{\text{KL}} \left(f_{\theta_{1:t}^*}(\mathbf{X}) \| f_{\theta_{1:t_c}}(\mathbf{X}) \right) \right] \quad (4)$$

where α is the hyper-parameter to trade off the current task and all previous tasks. D_{KL} is the Kullback–Leibler divergence between the best response of sample input \mathbf{X} with $\theta_{1:t_c}$ and optimal parameters $\theta_{1:t}^*$. In the vehicle motion forecasting of SVs, the input feature combines dynamic information from traffic participants and static information from the road map:

$$\mathbf{X} = \{\mathbf{O}, \mathbf{E}\} \quad (5)$$

where $\mathbf{O} \in \mathbb{R}^{d_v \times d_s}$ and $\mathbf{E} \in \mathbb{R}^{d_e}$ denote features for dynamic and static inputs, respectively. d_v is the number of vehicles in the scenario and d_s is the length of feature space for each vehicle. The output of DNN $\hat{\mathbf{Y}} \in \mathbb{R}^{l \times w}$ is represented by the fixed-size heatmap, which is the non-parametric distribution of the two-dimensional goal position. l and w are the length and width of the rectangular heatmap, respectively. The prediction of goal distribution can be formulated as:

$$\hat{\mathbf{Y}} = f_{\theta}(\mathbf{X}) = f_{\theta}(\mathbf{O}, \mathbf{E}) \quad (6)$$

The structure of the DNN for vehicle motion forecasting is detailed in the Supplementary Material (Section 2).

Replay-based continual learning

For most data-driven learning processes of DNN, the costs associated with data storage and maintenance are substantial, especially when effective they need access to all previously encountered samples. Furthermore, as the number of tasks increases, methods that indiscriminately revisit a large volume of past task samples will degrade the training efficiency. This is unacceptable for models that require online learning and rapid updates. In CL, a common strategy is to maintain only a small subset of past samples to represent historical tasks. This collection of limited past samples is referred to as the memory buffer \mathcal{M} . Thus, the Eq. (4) can be transformed into:

$$\theta_{1:t_c}^* = \arg \min_{\theta_{1:t_c}} \mathcal{L}_{t_c} + \alpha \sum_{t=1}^{t_c-1} \mathbb{E}_{(\mathbf{X}, \mathbf{Y}) \sim \mathcal{M}_t} \left[D_{\text{KL}} \left(f_{\theta_{1:t}^*}(\mathbf{X}) \| f_{\theta_{1:t_c}}(\mathbf{X}) \right) \right] \quad (7)$$

where \mathcal{M}_t is the memory buffer of t^{th} task. it indicates that the learner can select samples from \mathcal{M}_t . To guarantee the efficiency of CL, the size of \mathcal{M} is strictly limited:

$$|\mathcal{M}| \ll \sum_{t=1}^T N_t \quad (8)$$

where $|\cdot|$ is the size of memory buffer. The maintenance of the memory buffer and the sampling process from it are collectively termed replay-based CL. In Eq. (7), the ground truth label (goal position) is not considered in the approximation to past samples, which is important information in measuring the performance. Following the operation in the last term of Eq. (7), an improved expression is as follow²⁰:

$$\theta_{1:t_c}^* = \arg \min_{\theta_{1:t_c}} \mathcal{L}_{t_c} + \alpha \sum_{t=1}^{t_c-1} \mathbb{E}_{(\mathbf{X}, \mathbf{Y}) \sim \mathcal{M}_t} \left[D_{\text{KL}} \left(f_{\theta_{1:t}^*}(\mathbf{X}) \| f_{\theta_{1:t_c}}(\mathbf{X}) \right) \right] + \beta \sum_{t=1}^{t_c-1} \mathbb{E}_{(\mathbf{X}, \mathbf{Y}) \sim \mathcal{M}_t} [\ell(f_{\theta_{1:t_c}}(\mathbf{X}), \mathbf{Y})] \quad (9)$$

where β is the hyper-parameter to trade off the influence of ground truth-based loss.

Online task-free continual learning paradigm (OTFCL)

Eq. (9) represents the standard form of CL with task constraints. However, in the fields of vehicle motion forecasting, not all scenarios have task identifiers, which can be manually specified as strong constraints in CL. In many cases, obtaining task identifiers is impractical. According to Eq. (9), if the task guidance is absent, it can be expressed in a more general task-free form:

$$\theta_{\text{TF}}^* = \arg \min_{\theta_{\text{TF}}} \mathcal{L}_{t_c} + \alpha \mathbb{E}_{(\mathbf{X}, \mathbf{Y}) \sim \mathcal{M}} [D_{\text{KL}}(f_{\theta^*}(\mathbf{X}) \| f_{\theta_{\text{TF}}}(\mathbf{X}))] + \beta \mathbb{E}_{(\mathbf{X}, \mathbf{Y}) \sim \mathcal{M}} [\ell(f_{\theta_{\text{TF}}}(\mathbf{X}), \mathbf{Y})] \quad (10)$$

where θ_{TF} are the weights of the model with the task-free CL setting. The task-based memory buffer \mathcal{M}_t is relaxed as \mathcal{M} , which stores selected samples from the stream data. It is the form of general continual learning (GCL)²⁰ with the following characteristics: no task label, no test time oracle, and limited memory buffer size.

To realize OTFCL based on GCL, four challenges need to be addressed: First, for a learner with online learning and updating capabilities, the samples selected from the memory buffer and those from the data stream can only be used for one backpropagation iteration, preventing multiple epochs in the training process. Second, from the perspective of efficient training in CL, the model cannot save the best model in real-time after training on samples obtained from the data stream. Therefore, the key information corresponding to the samples needs to be retained during the training process. Third, it is essential to select effective and representative samples from the stream data to build the memory buffer. Fourth, after constructing the memory buffer, an effective learning strategy is required under the setting of OTFCL. The first challenge is the basic setting in OTFCL, which puts a higher demand on one-shot learning. The second one is solved by the heatmap-based output representation. The last two issues are considered in the proposed Dual-LS algorithm.

Complementary Learning System theory in continual learning

The core of CLS theory is that the human brain has two complementary subsystems for learning and remembering knowledge: the hippocampus is responsible for short-term rapid memory, while the neocortex works for long-term structured memory. The interplay between the hippocampus and the neocortex effectively enhances the learning process. Inspired by this biological mechanism, we developed a new algorithm, Dual-LS, based on replay-based CL methods, which integrates CLS theory into the design of the memory buffer and model update strategy. In the following sections, we will introduce the design of the dual memory replay buffer and the strategy for the long-short term.

Dual memory replay buffer

In the replay-based task-free CL¹⁸, the most commonly used memory buffer selection method is reservoir sampling, which ensures that all samples from a dynamically distributed data stream have an equal probability of being selected. The detailed algorithm is shown in Extended Data Algorithm 1, and the illustration of equal probability is provided in Fig. 6a. The memory buffer obtained from reservoir sampling is denoted as \mathcal{M}_R . However, in the absence of task labels, if there is a data imbalance, the constraint of equal probability may lead to more frequent sampling from certain distributions. In other words, it will result in an imbalanced sample selection in the memory buffer. To trade off the samples across different distributions, the diversity is introduced into the memory buffer selection process. The goal is to address the imbalance in the data stream while ensuring sample diversity during buffer construction. Generally, for a data stream with N samples, to simplify the problem, we assume only one sample can be observed at a time. When receiving the n^{th} sample $n \in \{1, 2, \dots, N\}$, the following objective needs to be achieved to maintain the performance of the DNN:

$$\begin{aligned} \theta_{\text{TF}}^n &= \arg \min_{\theta_{\text{TF}}^n} \ell \left(f_{\theta_{\text{TF}}^n}(\mathbf{X}_n), \mathbf{Y}_n \right) \\ \text{s.t. } \ell \left(f_{\theta_{\text{TF}}^n}(\mathbf{X}_i), \mathbf{Y}_i \right) &\leq \ell \left(f_{\theta_{\text{TF}}^{n-1}}(\mathbf{X}_i), \mathbf{Y}_i \right) \quad \forall i \in \{1, 2, \dots, n-1\} \end{aligned} \quad (11)$$

where $f_{\theta_{\text{TF}}^n}$ is the DNN model for OTFCL parametrized by θ_{TF}^n . n and i are the indexes of current and past samples from the data stream, respectively. As suggested in GEM, the constraints in Eq. (11) are equivalent to the constraints of gradients:

$$\langle \mathbf{g}_n, \mathbf{g}_i \rangle = \left\langle \frac{\partial \ell \left(f_{\theta_{\text{TF}}^n}(\mathbf{X}_n), \mathbf{Y}_n \right)}{\partial \theta_{\text{TF}}^n}, \frac{\partial \ell \left(f_{\theta_{\text{TF}}^n}(\mathbf{X}_i), \mathbf{Y}_i \right)}{\partial \theta_{\text{TF}}^n} \right\rangle \geq 0 \quad \forall i \in \{1, 2, \dots, n-1\} \quad (12)$$

where \mathbf{g}_n and \mathbf{g}_i are gradients for current and past samples. $\langle \cdot, \cdot \rangle$ is the high-dimensional angle between two vectors. To realize the objective in Eq. (11) by satisfying constraints in Eq. (12), the optimization process can be converted into maximizing the variance of gradient directions:

$$\begin{aligned} \mathcal{M}_D^* &= \arg \max_{\mathcal{M}_D} \text{Var}_{\mathcal{M}_D} \left[\frac{\mathbf{g}}{\|\mathbf{g}\|} \right] \\ &= \arg \max_{\mathcal{M}_D} \left(\frac{1}{|\mathcal{M}_D|} \sum_{m \in \mathcal{M}_D} \left\| \frac{\mathbf{g}_m}{\|\mathbf{g}_m\|} \right\|^2 - \left\| \frac{1}{|\mathcal{M}_D|} \sum_{m \in \mathcal{M}_D} \frac{\mathbf{g}_m}{\|\mathbf{g}_m\|} \right\|^2 \right) \\ &= \arg \max_{\mathcal{M}_D} \left(1 - \frac{1}{|\mathcal{M}_D|^2} \sum_{i,j \in \mathcal{M}_D} \frac{\langle \mathbf{g}_i, \mathbf{g}_j \rangle}{\|\mathbf{g}_i\| \|\mathbf{g}_j\|} \right) \end{aligned} \quad (13)$$

where \mathcal{M}_D is the sampling diversity-based memory buffer and $|\mathcal{M}_D|$ is the number of samples in \mathcal{M}_D . $\|\cdot\|$ is the 2-norm for gradients. m, i and j are indexes of samples in \mathcal{M}_D . To improve the efficiency in maintaining the memory buffer \mathcal{M}_D , a greedy-based algorithm is applied, which is presented in Extended Data Algorithm 2. The illustration of gradient-based diversity sampling strategy are detailed in Fig. 6b. With samples from two kinds of memory \mathcal{M}_R and \mathcal{M}_D , the learning process in Eq. (10) is further updated as:

$$\theta_{TF}^* = \arg \min_{\theta_{TF}^n} \mathcal{L}_{tc} + \alpha_1 \mathbb{E}_{(\mathbf{X}, \mathbf{Y}) \sim \mathcal{M}_R} [D_{KL}(f_{\theta^*}(\mathbf{X}) \| f_{\theta_{TF}}(\mathbf{X}))] + \beta_1 \mathbb{E}_{(\mathbf{X}, \mathbf{Y}) \sim \mathcal{M}_R} [\ell(f_{\theta_{TF}}(\mathbf{X}), \mathbf{Y})] \\ + \alpha_2 \mathbb{E}_{(\mathbf{X}, \mathbf{Y}) \sim \mathcal{M}_D} [D_{KL}(f_{\theta^*}(\mathbf{X}) \| f_{\theta_{TF}}(\mathbf{X}))] + \beta_2 \mathbb{E}_{(\mathbf{X}, \mathbf{Y}) \sim \mathcal{M}_D} [\ell(f_{\theta_{TF}}(\mathbf{X}), \mathbf{Y})] \quad (14)$$

where $\alpha_1, \alpha_2, \beta_1$ and β_2 are parameters trading off different replay loss.

Long-short term model update strategy

In addition to incorporating CLS theory into the design of the dual memory buffer, a fast-slow integrated model update mechanism is designed in Dual-LS to reflect the interplay between rapid and consolidated memory in the human brain when acquiring new knowledge. Specifically, the model $f_{\theta_{TF}^*}$ trained directly using the data stream under the OTFCL paradigm is the working model $f_{\theta_{TF}}^W$. Furthermore, based on the working model, we simultaneously maintain a fast learning model $f_{\theta_{TF}^*}^F$ and a slow learning model $f_{\theta_{TF}^*}^S$, corresponding to rapid and consolidated knowledge, respectively. The update of the working model is based on the Eq. (14). Meanwhile, $f_{\theta_{TF}^*}^F$ and $f_{\theta_{TF}^*}^S$ are stochastically updated with fixed manually set probabilities p_F and p_S . We make $p_F < p_S$ to guarantee that the fast model changes frequently. The specific strategy is taking an exponential moving average of the working model's weights:

$$\theta_{TF}^F = \alpha_F \theta_{TF}^F + (1 - \alpha_F) \theta_{TF}^W \quad (15)$$

$$\theta_{TF}^S = \alpha_S \theta_{TF}^S + (1 - \alpha_S) \theta_{TF}^W \quad (16)$$

where α_F and α_S are decaying rates. Additionally, to reflect the enhancement effect of the fast model and slow model on the working model, the memory buffer samples used to update the working model will select the heatmap with the smallest prediction error for the calculation of KL divergence in Eq. (14):

$$f_{\theta^*}(\mathbf{X}_M) = \begin{cases} f_{\theta^*}^F(\mathbf{X}_M) & \text{if } \ell(f_{\theta^*}^F(\mathbf{X}_M), \mathbf{Y}_M) < \ell(f_{\theta^*}^S(\mathbf{X}_M), \mathbf{Y}_M) \\ f_{\theta^*}^S(\mathbf{X}_M) & \text{other} \end{cases} \quad (17)$$

where \mathbf{X}_M and \mathbf{Y}_M are samples selected from the memory buffer. The complete diagram of the long short term model update is detailed in Extended Data Algorithm 3.

Implementation

In this paper, we mainly consider the DIL setting in OTFCL paradigm. To verify the proposed Dual-LS algorithm, the interaction dataset is selected, which covers intersections, roundabouts, and highway merging. In experiments, each traffic scenario corresponds to a task. The duration of the original recorded video, the number of samples, and the corresponding order in data streaming are detailed in the Supplementary Material (Section 1). The proposed OTFCL paradigm and dual-LS algorithm both require a basic DNN model in specific prediction tasks. Our previous work uncertainty quantification network (UQnet) is selected as the base network for vehicle motion forecasting⁴⁹, which is the 1st SOTA algorithm for the interaction prediction challenge in ICCV 2021¹. As shown in Eq. (5), the input of UQnet includes the historical trajectory information of surrounding vehicles in 1 second and the static map information, and the output is the non-parametric probability distribution of the target position after 3 seconds, which sums to 1. The details of the UQnet are introduced in the Supplementary Material (Section 2). In the training process, UQnet uses focal loss to quantify the differences between non-parametric distribution and sparse target positions.

Baseline algorithms

To verify the effectiveness of the proposed Dual-LS algorithm, the following algorithms are selected as baselines.

- Vanilla⁴⁹: Vanilla means that the UQnet trained sequentially from 1st to 8th scenarios. No CL strategy is applied to mitigate the catastrophic forgetting in DNN.

¹INTERPRET: Interaction-dataset-based prediction challenge, single-agent track, organized by ICCV 2021. Available at: <https://challenge.interaction-dataset.com/leader-board/>. Last accessed on May 3rd, 2025.

- GEM⁵⁰: GEM is a typical replay-based CL algorithm, which needs the guidance of task label. GEM equally allocates memory resource to each past task and the gradient update directions from each task's replay buffer are integrated together with an inequality constraint.
- A-GEM³²: A-GEM is the modified version of GEM, which relaxes the setting of task-based to task-free.
- DER²⁰: A replay-based general CL method, which uses reservoir sampling to maintain the memory buffer.
- GSS²⁴: A replay-based task-free CL method, which applies the gradient-based sample selection in the memory buffer.

Evaluation metrics

To evaluate the prediction accuracy of the proposed approach, this work adopts two metrics⁴⁹: the minimum final displacement error (FDE) and the miss rate (MR). The UQnet generates K predicted goals, where the k^{th} position of i^{th} sample is denoted as $\hat{\mathbf{p}}_k^i$. FDE measures the minimum Euclidean distance between the K predicted positions and the ground truth \mathbf{p} over each sample. For the i^{th} sample, FDE is formulated as:

$$\text{FDE}_K^i = \min_{k \in \{1, \dots, K\}} \|\hat{\mathbf{p}}_k^i - \mathbf{p}^i\|_2 \quad (18)$$

where $\|\cdot\|_2$ is the measurement of Euclidean distance. For each task \mathcal{T}_t ($1 \leq t \leq T$), the FDE is mean of all samples in the test set,

$$\text{FDE}_t = \frac{1}{N'_t} \sum_{i=1}^{N'_t} \text{FDE}_K^i \quad (19)$$

where N'_t is the number of samples in the test set of task \mathcal{T}_t . The average FDE over all tasks is as follow:

$$\text{FDE}_{\text{Ave}} = \frac{1}{T} \sum_{t=1}^T \text{FDE}_t \quad (20)$$

The MR measures the percentage of predicted goals that are out of the rectangular area around the ground truth. The lateral threshold of the given area is 1 m, and the longitudinal threshold is determined by the piece-wise function depending on the velocity v of the predicted vehicle:

$$th_{\text{MR}}(v) = \begin{cases} 1, v < 1.4 \\ 1 + \frac{v-1.4}{11-1.4}, 1.4 \leq v \leq 11 \\ 2, v > 11 \end{cases} \quad (21)$$

where the unit of the longitudinal threshold is m, and the unit of the velocity v is ms^{-1} . Let N_t^{miss} represent the total number of predicted goals that are out of the given area, the MR in the testing set of task \mathcal{T}_t is $\frac{N_t^{\text{miss}}}{N'_t \times W}$. The average MR over T tasks can be represented as:

$$\text{MR}_{\text{Ave}} = \frac{1}{T} \sum_{t=1}^T \frac{N_t^{\text{miss}}}{N'_t \times W} \quad (22)$$

Since FDE and MR both measure the prediction errors, the smaller values of FDE and MR represent the better prediction performance.

For the evaluation of the capability of CL, FDE-based backward transfer (FDE-BWT) and MR-based backward transfer (MR-BWT) are proposed to quantify catastrophic forgetting in CL tasks. If the prediction errors at the j^{th} task after training on the i^{th} task is denoted as $R_{i,j}$, FDE-BWT and MR-BWT can be computed uniformly by:

$$\text{FDE-BWT} = \frac{1}{t-1} \sum_{i=1}^{t-1} (\text{FDE}_{c,i} - \text{FDE}_{i,i}) \quad \forall t \in \{1, \dots, T\} \quad (23)$$

$$\text{MR-BWT} = \frac{1}{t-1} \sum_{i=1}^{t-1} (\text{MR}_{c,i} - \text{MR}_{i,i}) \quad \forall t \in \{1, \dots, T\} \quad (24)$$

Similar to FDE and MR, the smaller FDE-BWT and MR-BWT represent better CL performance to avoid catastrophic forgetting.

References

1. Mnih, V. *et al.* Human-level control through deep reinforcement learning. *nature* **518**, 529–533 (2015).
2. Silver, D. *et al.* A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science* **362**, 1140–1144 (2018).
3. Jumper, J. *et al.* Highly accurate protein structure prediction with alphafold. *nature* **596**, 583–589 (2021).
4. Feng, S. *et al.* Dense reinforcement learning for safety validation of autonomous vehicles. *Nature* **615**, 620–627 (2023).
5. Kaufmann, E. *et al.* Champion-level drone racing using deep reinforcement learning. *Nature* **620**, 982–987 (2023).
6. Song, Y., Romero, A., Müller, M., Koltun, V. & Scaramuzza, D. Reaching the limit in autonomous racing: Optimal control versus reinforcement learning. *Sci. Robotics* **8**, eadg1462 (2023).
7. Gehrig, D. & Scaramuzza, D. Low-latency automotive vision with event cameras. *Nature* **629**, 1034–1040 (2024).
8. Mozaffari, S., Al-Jarrah, O. Y., Dianati, M., Jennings, P. & Mouzakitis, A. Deep learning-based vehicle behavior prediction for autonomous driving applications: A review. *IEEE Transactions on Intell. Transp. Syst.* **23**, 33–47 (2020).
9. Wang, X. *et al.* Traffic light optimization with low penetration rate vehicle trajectory data. *Nat. communications* **15**, 1306 (2024).
10. Jafari, A. & Liu, Y.-C. Pedestrians' safety using projected time-to-collision to electric scooters. *Nat. communications* **15**, 5701 (2024).
11. Chen, Z. *et al.* Predicting driving comfort in autonomous vehicles using road information and multi-head attention models. *Nat. Commun.* **16**, 2709 (2025).
12. Cao, Z. *et al.* Continuous improvement of self-driving cars using dynamic confidence-aware reinforcement learning. *Nat. Mach. Intell.* **5**, 145–158 (2023).
13. Lesort, T. *et al.* Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Inf. fusion* **58**, 52–68 (2020).
14. Tajbakhsh, N. *et al.* Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging* **35**, 1299–1312 (2016).
15. Perkonigg, M. *et al.* Dynamic memory to alleviate catastrophic forgetting in continual learning with medical imaging. *Nat. communications* **12**, 5678 (2021).
16. Van de Ven, G. M., Tuytelaars, T. & Tolias, A. S. Three types of incremental learning. *Nat. Mach. Intell.* **4**, 1185–1197 (2022).
17. Kirkpatrick, J. *et al.* Overcoming catastrophic forgetting in neural networks. *Proc. national academy sciences* **114**, 3521–3526 (2017).
18. Wang, L., Zhang, X., Su, H. & Zhu, J. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis Mach. Intell.* **46**, 5362–5383 (2024).
19. De Lange, M. *et al.* A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis machine intelligence* **44**, 3366–3385 (2021).
20. Buzzega, P., Boschini, M., Porrello, A., Abati, D. & Calderara, S. Dark experience for general continual learning: a strong, simple baseline. *Adv. neural information processing systems* **33**, 15920–15930 (2020).
21. Kudithipudi, D. *et al.* Biological underpinnings for lifelong learning machines. *Nat. Mach. Intell.* **4**, 196–210 (2022).
22. Kumaran, D., Hassabis, D. & McClelland, J. L. What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends cognitive sciences* **20**, 512–534 (2016).
23. Sun, W., Advani, M., Spruston, N., Saxe, A. & Fitzgerald, J. E. Organizing memories for generalization in complementary learning systems. *Nat. neuroscience* **26**, 1438–1448 (2023).
24. Aljundi, R., Lin, M., Goujaud, B. & Bengio, Y. Gradient based sample selection for online continual learning. *Adv. neural information processing systems* **32** (2019).
25. Yan, X. *et al.* Learning naturalistic driving environment with statistical realism. *Nat. communications* **14**, 2037 (2023).
26. Feng, S., Yan, X., Sun, H., Feng, Y. & Liu, H. X. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nat. communications* **12**, 748 (2021).
27. Yu, W. *et al.* Online legal driving behavior monitoring for self-driving vehicles. *Nat. communications* **15**, 408 (2024).

28. Duan, J. *et al.* Spatiotemporal dynamics of traffic bottlenecks yields an early signal of heavy congestions. *Nat. communications* **14**, 8002 (2023).
29. Doshi, K. & Yilmaz, Y. Continual learning for anomaly detection in surveillance videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 254–255 (2020).
30. Zhan, W. *et al.* Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. *arXiv preprint arXiv:1910.03088* (2019).
31. Ettinger, S. *et al.* Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9710–9719 (2021).
32. Chaudhry, A., Ranzato, M., Rohrbach, M. & Elhoseiny, M. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations* (2018).
33. Xing, J. *et al.* A comprehensive study on self-learning methods and implications to autonomous driving. *IEEE Transactions on Neural Networks Learn. Syst.* **36**, 7786–7805 (2025).
34. Wang, W. *et al.* Social interactions for autonomous driving: A review and perspectives. *Foundations Trends Robotics* **10**, 198–376 (2022).
35. Dai, T. *et al.* Autonomous mobile robots for exploratory synthetic chemistry. *Nature* **635**, 890–897 (2024).
36. Zhou, C. *et al.* Human-behaviour-based social locomotion model improves the humanization of social robots. *Nat. Mach. Intell.* **4**, 1040–1052 (2022).
37. Meng, Y. *et al.* Preserving and combining knowledge in robotic lifelong reinforcement learning. *Nat. Mach. Intell.* **7**, 256–269 (2025).
38. Stanojevic, A. *et al.* High-performance deep spiking neural networks with 0.3 spikes per neuron. *Nat. Commun.* **15**, 6793 (2024).
39. Yin, B., Corradi, F. & Bohté, S. M. Accurate online training of dynamical spiking neural networks through forward propagation through time. *Nat. Mach. Intell.* **5**, 518–527 (2023).
40. Jiao, L. *et al.* Brain-inspired learning, perception, and cognition: A comprehensive review. *IEEE Transactions on Neural Networks Learn. Syst.* **36**, 5921–5941 (2025).
41. Wu, Y. *et al.* Brain-inspired global-local learning incorporated with neuromorphic computing. *Nat. Commun.* **13**, 65 (2022).
42. Van de Ven, G. M., Siegelmann, H. T. & Tolias, A. S. Brain-inspired replay for continual learning with artificial neural networks. *Nat. communications* **11**, 4069 (2020).
43. Williamson, M. R. *et al.* Learning-associated astrocyte ensembles regulate memory recall. *Nature* **637**, 478–486 (2024).
44. Gava, G. P. *et al.* Organizing the coactivity structure of the hippocampus from robust to flexible memory. *Science* **385**, 1120–1127 (2024).
45. Zaki, Y. *et al.* Offline ensemble co-reactivation links memories across days. *Nature* **637**, 145–155 (2024).
46. Shuai, Y. *et al.* Forgetting is regulated through rac activity in drosophila. *Cell* **140**, 579–589 (2010).
47. Soltoggio, A. *et al.* A collective ai via lifelong learning and sharing at the edge. *Nat. Mach. Intell.* **6**, 251–264 (2024).
48. Hassabis, D., Kumaran, D., Summerfield, C. & Botvinick, M. Neuroscience-inspired artificial intelligence. *Neuron* **95**, 245–258 (2017).
49. Li, G., Li, Z., Knoop, V. L. & van Lint, H. Unravelling uncertainty in trajectory prediction using a non-parametric approach. *Transp. Res. Part C: Emerg. Technol.* **163**, 104659 (2024).
50. Lopez-Paz, D. & Ranzato, M. Gradient episodic memory for continual learning. *Adv. neural information processing systems* **30** (2017).

Acknowledgements

This research is supported by the National Natural Science Foundation of China under Grant 52372405.

Data availability

The dataset used in this study is publicly available on <https://interaction-dataset.com/>.

Code availability

The code used in this study is publicly available on <https://github.com/lzrbit/Dual-LS>.

Author contributions statement

Z.L., C.L., and J.G. led the research project, and wrote the paper. Z.L., Y.L., and C.G developed the framework and algorithm. G.D., X.Z., C.L., and C.G. wrote the paper. All authors provided feedback during the manuscript writing and results discussions. C.L., and J.G. approved the submission and accepted responsibility for the overall integrity of the paper.

Additional information

Supplementary information The online version contains supplementary material available at the website.

Correspondence and requests for materials should be addressed to Chao Lu and Jianwei Gong.

Extended data

Extended Data Algorithm 1: Reservoir sampling strategy

Input: The data stream \mathcal{D} , where the n^{th} sample is denoted as \mathbf{d}_n ; the rapid memory buffer \mathcal{M}_R with maximum size $|\mathcal{M}_R| \in \mathbb{Z}^+$.

Output: The updated \mathcal{M}_R , where the i^{th} sample stored in \mathcal{M}_R is denoted as $\mathcal{M}_R(i)$.

```

1  $\mathcal{M}_R \leftarrow \emptyset;$  ▷Initialization.
2 for  $n$  in  $\text{range}(1, |\mathcal{D}|)$  do
3   if  $n \leq |\mathcal{M}_R|$  then
4      $\mathcal{M}_R(n) \leftarrow \mathbf{d}_n$   $\mathcal{M}_R \leftarrow \mathcal{M}_R \cup \mathcal{M}_R(n)$ 
5   else
6      $r \sim \text{Uniform}(1, n);$ 
7     if  $r \leq |\mathcal{M}_R|$  then
8        $\mathcal{M}_R(r) \leftarrow \mathbf{d}_n;$  ▷Sample replacement.
9     end
10   end
11 end

```

Extended Data Algorithm 2: Gradient-based diversity sampling strategy

Input: The data stream \mathcal{D} , where the n^{th} sample is denoted as \mathbf{d}_n with the similarity score q_n ; the distinctive memory buffer \mathcal{M}_D with the maximum size $|\mathcal{M}_D|$; the number of samples B for score computation; the loss function ℓ ; the vehicle motion forecasting model $f_{\theta_{\text{TF}}}$.

Output: The updated \mathcal{M}_D , where the i^{th} stored sample is denoted as $\mathcal{M}_D(i)$.

```

1 for  $n$  in  $\text{range}(1, |\mathcal{D}|)$  do
2   if  $n == 1$  then
3      $q_n \leftarrow 0.1;$  ▷Initialization.
4   else
5      $\mathbf{g} \leftarrow \nabla \ell(f_{\theta_{\text{TF}}}(\mathbf{d}_n)), \Omega \leftarrow \emptyset;$ 
6     for  $k$  in  $\text{range}(1, B)$  do
7        $b \sim \text{Uniform}(1, |\mathcal{M}_D|), \Omega \leftarrow \Omega \cup \mathcal{M}_D(b);$  ▷Randomly select  $B$  stored samples for the score computation.
8     end
9     for all samples in  $\Omega$  do
10     $\mathbf{g}_b \leftarrow \nabla \ell(f_{\theta_{\text{TF}}}(\mathcal{M}_D(b)));$ 
11   end
12    $q_n \leftarrow \max_b \left( \frac{\langle \mathbf{g}, \mathbf{g}_b \rangle}{\|\mathbf{g}\| \|\mathbf{g}_b\|} \right) + 1;$  ▷Similarity score.
13   if  $n \leq |\mathcal{M}_D|$  then
14      $\mathcal{M}_D \leftarrow \mathcal{M}_D \cup \mathbf{d}_n;$ 
15   else
16     if  $q_n < 1$  then
17        $i \sim \mathbb{P}(i) = \frac{q_i}{\sum_{i=1}^{|\mathcal{M}_D|} q_i}, r \sim \text{Uniform}(0, 1);$ 
18       if  $r < \frac{q_i}{q_i + q_n}$  then
19          $\mathcal{M}_D(i) \leftarrow \mathbf{d}_n, q_i \leftarrow q_n;$  ▷Sample replacement.
20       end
21     end
22   end
23 end
24 end

```

Extended Data Algorithm 3: Long-short term model update strategy

Input: The data stream \mathcal{D} , memory buffer \mathcal{M}_R and \mathcal{M}_D Learning rate α_W , decaying rates α_F and α_S , Update rates p_F and p_S

Output: θ_{TF}^F , θ_{TF}^S and θ_{TF}^W .

1 $\theta_{TF}^F = \theta_{TF}^S = \theta_{TF}^W$; ▷Initialization.

2 **while** Training **do**

3 $(X, Y) \sim \mathcal{D}$;

4 $(X_M, Y_M) \sim \mathcal{M}_R, \mathcal{M}_D$;

5 $\hat{Y}_M^F = f_{\theta^*}^F(X_M)$, $\hat{Y}_M^S = f_{\theta^*}^S(X_M)$;

6 $\hat{Y}_M \leftarrow \hat{Y}_M^F$ **if** $\ell(f_{\theta^*}^F(X_M), Y_M) < \ell(f_{\theta^*}^S(X_M), Y_M)$ **else** \hat{Y}_M^S as in (17);

7 Calculate total \mathcal{L} as in (14);

8 $\theta_{TF}^W \leftarrow \theta_{TF}^W - \alpha_W \nabla_{\theta_{TF}^W} \mathcal{L}$;

9 $a, b \sim \text{Uniform}(0, 1)$;

10 $\theta_{TF}^F = \alpha_F \theta_{TF}^F + (1 - \alpha_F) \theta_{TF}^W$ **if** $a < p_F$ **else** θ_{TF}^F ;

11 $\theta_{TF}^S = \alpha_S \theta_{TF}^S + (1 - \alpha_S) \theta_{TF}^W$ **if** $b < p_S$ **else** θ_{TF}^S ;

12 Update \mathcal{M}_R as in (15);

13 Update \mathcal{M}_D as in (16);

14 **end**
