2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Preventing Catastrophic Forgetting in Continuous Online Learning for Autonomous Driving

Rui Yang[1], Tao Yang[2], Zhi Yan[1*], Tomas Krajnik[3], Yassine Ruichek[1]

***Abstract*— Autonomous vehicles require online learning capabilities to enable long-term, unattended operation. However, long-term online learning is accompanied by the problem of forgetting previously learned knowledge. This paper introduces an online learning framework that includes a catastrophic forgetting prevention mechanism, named Long-Short-Term Online Learning (LSTOL). The framework consists of a set of short-term learners and a long-term controller, where the former is based on the concept of ensemble learning and aims to achieve rapid learning iterations, while the latter contains a simple yet efficient probabilistic decision-making mechanism combined with four control primitives to achieve effective knowledge maintenance. A novel feature of the proposed LSTOL is that it avoids forgetting while learning autonomously. In addition, LSTOL makes no assumptions about the model type of short-term learners and the continuity of the data. The effectiveness of the proposed framework is demonstrated through experiments across well-known datasets in autonomous driving, including KITTI and Waymo. The source code for the method implementation is publicly available at `https://github.com/epan-utbm/lstol`.**

I. Introduction

As the field of mobile robotics continues to advance rapidly, unmanned autonomous vehicles have emerged as a promising solution within the transportation industry. The intelligence functioning of these vehicles heavily depend on their ability to effectively sense and learn objects, allowing them to swiftly identify and understand various entities in real-time, including cars, pedestrians, cyclists and other participants in the complex road situation. Over the past decade, machine learning has made remarkable progress in object detection, especially with the deep learning models that outperform humans [1], [2]. However, deploying machine learning to autonomous vehicles faces challenges including expensive training, deployment and maintenance costs, domain shift, long tail problem, and so forth [3].

Compared to offline training, online learning is considered an effective solution [4], [5]. An open problem in the latter is how autonomous vehicles can prevent catastrophic forgetting while continuously absorbing new knowledge. While many upstream methods have been proposed to address this problem in deep neural networks [6], [7], [8], they are inherently rooted in offline or batch training and thus, are incompatible with online learning by design.

This work was supported by the China Scholarship Council (CSC-UT-INSA program), the Contrat de Plan État-Région (CPER Mobilitech), the National Natural Science Foundation of China (No. 62103332), the CZ MSMT (No. 8J23FR023) and PHC Barrande (No. 49275QM) project 3L4AR, and the project ROBOPROX no. CZ.02.01.01/00/22_008/0004590.

[1]UTBM, CIAD UMR 7533, F-90010 Belfort, France. `firstname.lastname@utbm.fr`

[2]Unmanned System Research Institute, Northwestern Polytechnical University, China. `yangtao@nwpu.edu.cn`

[3]Artificial Intelligence Center, Czech Technical University in Prague, Czechia. `krajnt1@fel.cvut.cz`

*Corresponding Author: `zhi.yan@utbm.fr`

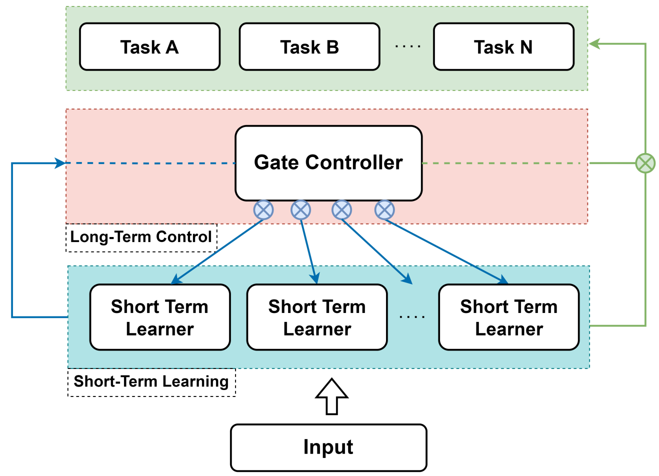

Fig. 1. Illustration of the Long-Short-Term Online Learning (LSTOL) framework which consists of two modules: short-term learning and long-term control. Input samples are first processed by a set of short-term learners for pre-prediction. The long-term control module collects these pre-prediction to calculate indicators for online prediction, then passes them to the Gate Controller to determine actions for each learner. During the learning phase (blue line), the module calculates the online loss to update the learner weights, which are used in the prediction phase (green line) to determine the object category.

Our previous work has shown that vehicles can autonomously and rapidly learn road participant detection capabilities in a deployed environment, on-the-fly and without human intervention [4], and with good online adaptability across environments [5]. This paper builds on these foundations to further investigate how to avoid the catastrophic forgetting problem in the process of in-situ learning, by extending online learning to online continual learning, and still uses the detection of road participants including cars, pedestrians, and cyclists as a downstream task.

In continual learning, catastrophic forgetting usually happens in two scenarios: an increase in classes or tasks that need to be learned, and a shift of knowledge domains. The method proposed in this paper addresses the challenges posed by the latter, aiming to adapt to new data distributions without forgetting knowledge from previous learned domains. Specifically, we propose an ensemble learning framework, named Long-Short-Term Online Learning (LSTOL), which consists of a set of short-term learners and a long-term control mechanism, as shown in Fig. 1. The former can be any model but needs to be subject to the requirements of online learning, such as fast iteration without saving learning samples. The latter contains a gate controller that controls whether each existing short-term learner should be updated, kept or removed, or a new short-term learner should be created. The design of the controller is based on primitives rather than complex reasoning, fully considering the real-time requirements of physical interaction of vehicles in the real world. It is worth mentioning that, unlike the well-known long short-term memory (LSTM) [9], LSTOL emphasizes the learning strategy rather than the network structure, makes no assumptions about the continuity of learning data, and allows any short-term model in design.

The contributions of this paper are twofold.

- A novel framework to prevent catastrophic forgetting in online learning is proposed, without assumptions on the structure and complexity of short-term model, using a simple yet efficient long-term control strategy.
- Taking road participant detection in autonomous driving as a downstream task, two very different datasets including KITTI [10] and Waymo [11] are used to conduct online continual learning experiments, and results show that the proposed framework enables the vehicle to learn new knowledge while avoiding catastrophic forgetting.

II. Related Work

In general, catastrophic forgetting occurs when learning new tasks degrades performance on previously learned tasks, posing challenges for long-term deployment [12]. Research on catastrophic forgetting can be traced back to the 1990s [13], [14], with approaches like memory buffer [15] that store past data or gradients to constrain updates.

In situations where retaining information from previous tasks is impractical due to privacy or resource constraints, regularization-based methods [7], [16], [17] offer a solution by employing cleverly designed regularization losses to constrain forgetting old knowledge while learning new data. Another intuitive solution is to build a sufficiently large model and create a subset of the model for each task. This can be achieved by fixing the shared trunk and adding new branches for each new task, allowing old and new knowledge to be separated. However, this will lead to another problem of scale explosion [8]. Moreover, replay-based approaches are grounded on the concept of retaining or compressing the underlying data of past tasks [18], [19], [20]. These methods combat forgetting by reintroducing stored samples during training when learning a new task, while the samples play a crucial role in joint training or loss optimization, protecting knowledge from previous tasks.

Autonomous vehicles have an essential need for agents to be able to learn on their own and continuously [3], [21], as they will be deployed into our daily lives for a long time [22], [23], [24]. In response to this need, our previous work [4], [5] proposed an online learning framework that allows vehicles to learn the detection of road participants in-situ and on-the-fly in the environment in which they operate. However, continual learning across different driving scenarios brings about catastrophic forgetting problems. This motivates us to explore related prevention mechanisms within the framework of online learning. A work with a similar concept to the learning framework proposed in this paper – to avoid forgetting when learning online – is the Lifelong Learning for Navigation (LLfN) method introduced in [25]. This method allows the robot to not forget the navigation experience of the old environment when exploring a new one. However, the essential differences between LLfN and our LSTOL is that the former aims to learn an auxiliary planner to only help the classical planner navigate in difficult situations, while the latter treats the model to be learned and the model to be used as identical, and makes no assumptions about the usage situations.

Another work related to our LSTOL is the Expert Gate method introduced in [26], which relies on different expert networks to handle the differences in data distribution between various tasks, and also utilizes gate control to intelligently select appropriate experts to achieve effective processing of different tasks. However, unlike LSTOL, which supports parallel online learning for multiple tasks, Expert Gate is an offline training method and learns different tasks in a sequential manner. In addition, the gate control strategy of Expert Gate is not like LSTOL, which can achieve differentiated learning for short-term learners. Moreover, as with [26], there has been much other work on avoiding catastrophic forgetting in computer vision [6], [7], [8], [27], [28] which has emerged alongside the boom in deep learning methods. Unfortunately, these methods cannot be straightforwardly applied to online robot learning due to their requirements on annotated data, computational resources, and training time. Therefore, our research goal is to establish an online learning framework that incorporates an autonomous forgetting prevention mechanism to enable vehicles to maintain stable performance on downstream tasks during long-term operation across environments.

III. Long-Short-Term Online Learning

As shown in Fig. 1, the LSTOL framework combines short-term learning and long-term control modules to prevent catastrophic forgetting during learning. The short-term learning module consist of multiple short-term learners. Each learner can be embodied as a model such as SVM, random forest, neural network, etc., leaning from streaming data of various modalities such as images or point clouds.

The long-term control module supervises short-term learners via a gate controller with three sub-functions.

- *Information Collection* gathers learners' confidence, accuracy and activity, forming the basis for decisions on knowledge retention and new learning.
- *Gate Controller* decides actions like retaining, updating, creating or deleting learners based on a joint evaluation of collected information and the predicted probabilities.
- *Weight Estimation* dynamically adjusts the learner weights based on the past performance. More accurate learners gain more weight, and high-confidence learners act as "experts" in final predictions.

It is worth emphasizing that the proposed LSTOL framework is learn-as-you-go, i.e., the output of the long-term control module can also be used for downstream tasks such as road participant detection.

IV. IMPLEMENTATION

The implementation of the LSTOL framework for point cloud-based road participant detection is shown in Fig. 2.

A. Learning Sample

The learning samples are extracted from point clouds generated by 3D lidar, defined as:

$$S = track(\{x, c, t\}, \bar{c}) \tag{1}$$

where $\{x, c, t\}$ represents a set of tracked instances (in the form of clusters) of object x at different times t. c and $\bar{c}$ respectively represent the confidence that each instance and the entire trajectory belong to an object class. Unlike the usual assumption of a fixed dataset for offline training, LSTOL processes streaming data. In practice, our previously proposed Efficient Online Transfer Learning (EOTL) [5] method is used for sample generation.

Specifically, as shown in Fig. 3, EOTL collects object detection proposals (i.e. labels and probabilities) from two different sensors and associates them spatiotemporally to form the trajectory. Then, the label associated probabilities of all samples on a single trajectory are fused to determine the object category. Finally, the determined trajectory label is assigned to all samples on the trajectory to form labeled learning samples. This sample generation method has the advantages of high efficiency, good reliability, and the ability to avoid individual bias. For example, a certain sample on a trajectory is considered more like a cyclist while other samples on the same trajectory are considered more like a car, the final misclassified sample can be corrected through global awareness.

B. Short-term Learning

In the short-term learning module (denoted by stl), each learner adopts an online random forest (ORF) [29], which facilitates rapid multi-class model training and real-time deployment. Boosting adjusts learner weights dynamically, focusing on challenging samples, while bagging aggregates multiple trees for robustness. The overall framework is formalized as:

$$stl(x) = h \sum_{i=1}^{I} w_i \mathrm{ORF}_i(x) \tag{2}$$

where w_i represents the weight of learner i, and h denotes the fusion strategy determined by the long-term control module. ORF's node-splitting mechanism is driven by the need to improve predictive accuracy and reduce uncertainty. The motivation behind splitting a leaf node in ORF lies in maximizing the information gain, thus enabling the model to better distinguish between classes. The splitting criterion is defined as:

$$|R_j| > \alpha \quad \wedge \quad \exists s \in S : \triangle L(R_j, s) > \beta \tag{3}$$

where α is the minimum number of samples a node must observe before splitting, β is the minimum gain required for a split, R_j is a decision node, and $\triangle L(R_j, s)$ represents the gain from test s, calculated as:

$$\triangle L(R_j, s) = L(R_j) - \frac{|R_{jls}|}{|R_j|} L(R_{jls}) - \frac{|R_{jrs}|}{|R_j|} L(R_{jrs}) \tag{4}$$

with R_{jls} and R_{jrs} being the left and right partitions based on s. The ORF model is updated by storing the tree structure and node information, thus retaining the learned knowledge without needing previous training samples. This approach ensures that the model continuously adapts while preserving previously acquired knowledge, effectively mitigating catastrophic forgetting.

The number of short-term learners is closely related to the distribution of data. In scenarios where data distributions shift significantly, a number of learners may be necessary to adapt to new environments and retain previously learned knowledge. Conversely, as the data distribution stabilizes, the contribution of adding more learners tends to diminish. This indicates that while performance generally improves with an increasing number of learners, the effect plateaus once the model has sufficiently adapted to the new distribution. Different tasks may also demand varying numbers of short-term learners. More complex tasks with greater class diversity or data variations, such as 3D object detection involving cars, pedestrians, and cyclists, benefit from a higher number of learners to capture diverse features. Simpler tasks may not require as many learners for optimal performance.

C. Long-term Control

1) Information Gatherer (IG): In online learning, traditional metrics like precision and recall are hard to achieve real-time performance evaluation without a complete test set or ground truth. Instead, we evaluate each learner using novel metrics for real-time requirement including confidence, accuracy, and activity. Confidence reflects the certainty of a learner's predictions. Accuracy measures learner's performance on past data, ensuring that learners with low accuracy are prioritized for updates. Activity tracks how frequently a learner has been updated, preventing learners from becoming too stagnant or overly active.

Specifically, faced with samples that constitute an object trajectory, a learner's confidence is formulated as:

$$Confidence_i = \max(p_i^j) \quad j = 1, \ldots, J \tag{5}$$

where p_i^j represents the predicted probability of learner i that each sample belongs to category j. The confidence of learner i takes the maximum value among all predicted probabilities. However, overconfidence is common in data-driven classifiers [30]. Therefore, one cannot rely solely on this metric to determine primitive operations for the short-term learners. Additionally, if a learner's predictions differ from what it predicts later on the entire object trajectory, it will be penalized during weight updates (See IV-C.3).

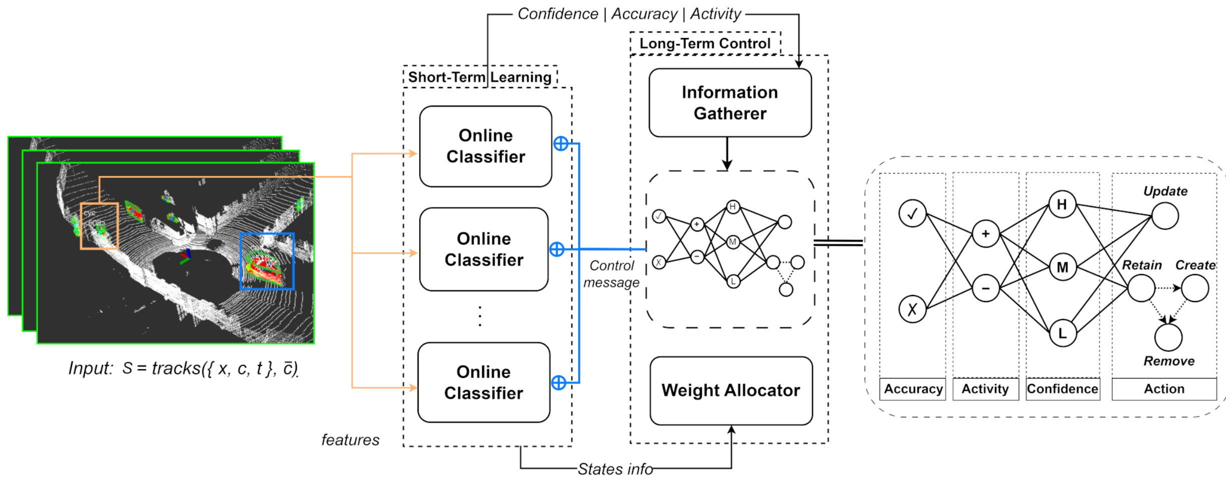

Fig. 2. Implementation overview of the LSTOL framework. Viewed from left to right: Samples in different point cloud frames are correlated by a multi-target tracker [5] and fed into each online classifier of the short-term learning module. The Information Gatherer (IG) in the long-term control module collects classification data and evaluates learners based on confidence, accuracy, and activity. This evaluation informs the Dynamic Gate Controller (DGC), which decides whether to update, retain, create, or remove classifiers. The Weight Allocator (WA) uses loss information to adjust classifier weights, which are then used for final object classification.

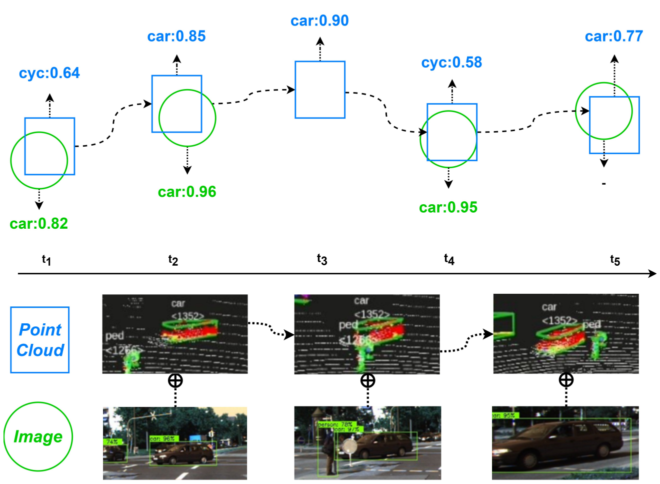

Fig. 3. Illustration of sample generation based on our previously proposed EOTL method. The rectangles represent the detection results of the point cloud detector that requires online learning, while the circles represent the detection results of the off-the-shelf image detector with guaranteed performance.

Accuracy is used to evaluate the learner's prediction accuracy for the entire trajectory and is formulated as:

$$Accuracy_i = \frac{p_i^{correct}}{p_i^{total}} \tag{6}$$

where $p_i^{correct}$ represents the number of correct predictions produced by learner i, and p_i^{total} represents the total number of predictions performed by learner i.

Activity is a measure of how often a learner updates, and is formulated as:

$$Activity_i = \sum_{t=1}^{T} update(i) \tag{7}$$

where T represents a time window.

2) Dynamic Gate Controller (DGC): The DGC mechanism is designed based on the stability-plasticity tradeoff theory in continual learning, to dynamically adjust the short-term learners based on their current performance in real-time. By integrating confidence, accuracy, and activity metrics, the system determines whether a learner should be updated, retained, or removed. This ensures that the model remains efficient, preventing unnecessary expansion or updates while maintaining high accuracy and avoiding catastrophic forgetting. A probabilistic decision-making process is designed in this module, summarized as Algorithm 1, which implements an appropriate operation based on the three metrics provided by the IG module. Specifically, In line 3, the P_{gate} function combines the learner's confidence, accuracy, and activity to compute the probability of updating the learner. The odds formula is defined as:

$$P_{gate}(Y|X,\mathscr{D}) = \frac{odds_X}{1+odds_X} \tag{8}$$

where

$$odds_X = \prod_{i=1}^{t}\prod_{j=1}^{K} odds_{x_i}^{j} \tag{9}$$

and

$$odds_{x_i}^{j} = \frac{P(y_i|x_i,d_j)}{1-P(y_i|x_i,d_j)} \tag{10}$$

The inversion of confidence and activity is crucial in ensuring that learners with low confidence or low activity are prioritized for updates. In contrast, high accuracy is used as a signal that the learner's performance is satisfactory, thus reducing the chances of unnecessary updates.

The "retain" operation represented in line-8 corresponds to two typical situations. The first is when the learner's accuracy is low, indicating that the learner's predictions for new samples are beyond its knowledge. The second is when the learner has both high accuracy and high confidence, which indicates that the learner is already very familiar with the input data and does not need to learn it anymore.

The idea behind line-16 is to remove learners, which is a risky operation and therefore only occurs when the maximum number of learners is reached and new learners need to be created. Removal will inevitably lead to the forgetting of some old knowledge, but we must find a compromise between the former and the unlimited number of learners. Line-26 means that if none of the existing learners have been updated, a new one will be created.

Algorithm 1 Dynamic Gate Control

Require: N: maximum number of learners to be created $Confidence_i$ (Con_i), $Accuracy_i$ (Acc_i), $Activity_i$ (Act_i)
Ensure: learner i, learner j, a new learner
1: $updated \leftarrow 0$
2: **for** each learner $i \in I$ **do**
3: $\quad p \leftarrow P_{gate}(1-Con_i,\ Acc_i,\ 1-Act_i)$
4: $\quad$ **if** $p > 0.5$ **then**
5: $\quad\quad$ **update** (learner i)
6: $\quad\quad updated \leftarrow 1$
7: $\quad$ **else**
8: $\quad\quad$ **retain** (learner i) // do nothing
9: $\quad$ **end if**
10: **end for**
11: **if** $updated = 0$ **then**
12: $\quad$ **if** $I = N$ **then**
13: $\quad\quad p_{max} \leftarrow 0$
14: $\quad\quad j \leftarrow \emptyset$
15: $\quad\quad$ **for** each learner $i \in I$ **do**
16: $\quad\quad\quad p \leftarrow P_{gate}(1-Con_i,\ 1-Acc_i,\ 1-Act_i)$
17: $\quad\quad\quad$ **if** $p > 0.5$ **and** $p > p_{max}$ **then**
18: $\quad\quad\quad\quad j \leftarrow i$ // mark learner i for removal
19: $\quad\quad\quad\quad p_{max} \leftarrow p$
20: $\quad\quad\quad$ **end if**
21: $\quad\quad$ **end for**
22: $\quad\quad$ **if** $j \neq \emptyset$ **then**
23: $\quad\quad\quad$ **remove** (learner j)
24: $\quad\quad$ **end if**
25: $\quad$ **else**
26: $\quad\quad$ **create** (learner)
27: $\quad\quad I \leftarrow I+1$
28: $\quad$ **end if**
29: **end if**

3) Weight Allocator (WA): The Weight Allocator (WA) is inspired by weighted voting strategies in ensemble learning. For each learner, a dynamic expert weights (DEW) table is constructed, by dynamically adjusting the weight of each learner based on their classification performance, the system amplifies the influence of more accurate learners while diminishing the impact of less reliable ones. This prevents the model from being overly biased towards newly learned data, which is crucial for avoiding forgetting. Specifically, assume that a new set of samples S is input at time step $t+1$, where the confidence that $s \in S$ belongs to a certain class $k \in K$ is very high, Kronecker delta is used to measure the sample's predicted class and actual class k, denoted as y_s. Learner i's predicted probability that sample s belongs to class k is denoted as $p_{s,k}$. Then the loss of sample s is calculated by log-loss:

$$L_{s,k} = y_s log(p_{s,k}) + (1-y_s) log(1-p_{s,k}) \tag{11}$$

Next, the current weights $w(t)$ are updated using an exponentially weighted moving average (EWMA), designed to reward accurate predictions and penalize incorrect ones:

$$w_k(t+1) = \lambda w_k(t) + (1-\lambda) L_{total} \tag{12}$$

where

$$L_{total} = -\frac{1}{S}\left(\sum_{s=1}^{S} L_{s,k}\right) \tag{13}$$

where $L_{s,k}$ represents the loss of sample s of class k. λ is determined according to the update speed of weights, used to balance the learner's past and present accuracy judgements.

The final prediction stage uses an intuitive voting strategy known as the hand-raised as expert (HRE):

$$p_k = \sum_{i,(p_{s,k} > \theta_c)}^{I} (p_{s,k} \cdot w_{i,k}) \tag{14}$$

where $w_{i,k}$ is the weight of learner i for class k, θ_c represents the minimum weight required by the learner to predict, which is set to 0.5 in our experiments. Essentially, this strategy prioritizes learners' predictions based on their degree of influence in the final prediction. This allows the long-term control module to pay more attention to the predictions of modules with higher weights.

V. Experimental Evaluation

A. Experimental Setups

Our experiments aim to evaluate whether the proposed LSTOL can effectively prevent catastrophic forgetting when learning across environments. To this end, two very different datasets in autonomous driving including KITTI [10] and Waymo [11] are used. Theoretically, the more different the data learned before and after, the greater the challenge in preventing catastrophic forgetting. In practice, an instance is designed to simulate an autonomous vehicle transitioning between two different environments.

Specifically, the system first performs online learning on the KITTI dataset, then switches to Waymo for continual learning, causing a domain shift. Our previous research [5] demonstrated that performance drops when models trained on KITTI are deployed on Waymo, but by online continual learning on the latter, the model performance will rebound. Finally, the system returns to KITTI to assess if learning on Waymo caused catastrophic forgetting.

1) Datasets: Learning is initiated using randomly sampled segments from the raw data (without any annotations) of the "City", "Residential", and "Campus" scenes in KITTI. These three scene categories were selected because they contain a relatively large number of road participants, while the other two scenes, "Road" and "Person", are relatively monotonous thus unsuitable for online learning of our downstream task. The system iterates the model every time it learns 100 samples and evaluates the model performance on the test set. The latter is built from randomly selected samples from the annotated training set of KITTI's 3D object detection task, containing 5347 cars, 668 pedestrians and 271 cyclists.

In the second step, we deploy the system trained on KITTI to the Waymo dataset. We randomly selected 15 segments from the latter, for a total of 2970 images and 2970 lidar scans. These segments are dominated by daytime and clear weather conditions, and the driving scenes include cars, pedestrians and cyclists, corresponding to the three classes learned by the system on the KITTI dataset. We deliberately avoid adverse weather conditions (e.g. foggy days [31], [32]) and scenes with poor lighting conditions (e.g. evenings [33], [34]) to ensure a fair comparison of model performance on the two datasets. The system continues learning online on these segments. It iterates the model every 100 learned samples and evaluates its performance on the KITTI test set.

2) Comparison Models: We compare our proposed LSTOL with the following methods:

- PointNet-STD: Each dataset is independently trained based on the baseline PointNet [35]. After training on the KITTI dataset, the model parameters are further trained on the Waymo dataset to update new parameters.
- PointNet-MIX: The baseline PointNet is jointly trained on the mixed data from KITTI and Waymo. The resulting model is evaluated on the KITTI validation set.
- Expert Gate [26]: To better suit our downstream tasks while staying true to the original algorithm, we switched the neural network-based Expert Gate to an Online Expert Gate (OEG) using ORF. Still, the core idea remains unchanged: selecting the most relevant expert to handle new data based on task relevance through comparison.
- Dynamic Expandable Network (DEN, the latest implementation is 3D-DEN) [28] can perform point cloud object classification tasks end-to-end and has the capability to dynamically expand the network, thus having online adaptability.

Moreover, we simplified the Autoencoder Gate within the Expert Gate. Instead of comparing task relevance based on validation sets of incoming data, as done in the original algorithm, we directly test the predictions of new data against the ground truth to determine task relevance. Once we find the expert with the highest task relevance, if the relevance exceeds a threshold (set to 0.85), we conduct online training on the existing expert (corresponding to Learning without Forgetting (LwF) [8]). Otherwise, we build a new expert to train the new data (corresponding to fine-tuning).

B. Evaluation across Datasets

The procedure for cross-dataset evaluation is as follows. First, the model is trained on the KITTI training set and then tested on the KITTI validation set. Second, the model is further trained on the Waymo dataset and the updated model is tested on the same KITTI validation set used in the first step. Table I illustrates the performance difference between ours and the aforementioned four methods. It is worth pointing out that, due to the sequential requirements of the tasks to be learned by the compared methods (i.e., learning the classification tasks of cars, pedestrians, and cyclists one by one), – different from LSTOL, which supports multi-task parallel learning – it forces us to manually sort the input samples. However, in real-world deployments of autonomous vehicles, it would be more beneficial for the vehicle to learn multiple tasks in parallel.

As can be seen from Table I, PointNet-STD shows superior performance under the data from the same distribution. However, its shortcomings are also obvious, as its classification performance dropped significantly for all three road participant categories once trained on the Waymo dataset. On the other hand, although PointNet-MIX maintains consistent performance on mixed datasets, as tasks and data increase, this advantage will not persist, and the training cost will also increase significantly. Expert Gate and 3D-DEN show competitive results in avoiding forgetting, but our proposed LSTOL maintains the overall performance advantage (i.e. the classification performance of three categories of road participants) after training on the Waymo dataset. Moreover, compared to the other two methods, LSTOL also shows the most balanced performance degradation and the least forgetfulness of pedestrians and cyclists.

C. LSTOL vs. EOTL

After comparing with other methods, we performed experiments to compare with our own EOTL method to demonstrate the advancement of LSTOL. We first evaluate the learning results of both. The upper part of Fig. 4 shows the classification performance results of EOTL and LSTOL after ten learning iterations on the KITTI dataset respectively, while its lower part shows the performance after ten more iterations on the Waymo dataset. It can be seen that compared with our previous EOTL method that does not include a catastrophic forgetting prevention mechanism, the effect of LSTOL is obvious. Specifically, the degradation in classification performance for cars, pedestrians, and cyclists dropped from 0.05, 0.09, and 0.15 to 0.03, 0.05, and 0.07, respectively. Furthermore, it can be seen that LSTOL surpasses EOTL in its ability to distinguish cars and cyclists from pedestrians.

Secondly we are interested in the details of the learning process of the two methods. Fig. 5 shows the changes in classification performance of different online learning models after each iteration. There are a total of 20 iterations, of which the first ten occur on KITTI and the last ten occur on Waymo. The evaluation metric used is “recall”. In multi-class classification tasks, recall is a metric that evaluates a model’s ability to correctly identify all examples of each class. It is calculated as the ratio of the number of true positive predictions to the total number of actual positive examples in the class. It can be seen that the anti-forgetting mechanism of LSTOL has a clear role in mitigating the performance degradation of the model caused by environmental changes, especially for pedestrians and cyclists, two types of road participants that are more difficult to detect than cars. The reason is attributed to the fact that LSTOL uses ensemble learning (a set of learners), which allows examples of each class to be better preserved independently. This is different from the single powerful learner we designed previously

TABLE I
EVALUATION OF THE CLASSIFICATION PERFORMANCE OF DIFFERENT METHODS ON THE KITTI DATASET BEFORE AND AFTER TRAINING ON THE WAYMO DATASET

Method	Only KITTI			+ Waymo					
	Cars(%)	Pedestrians(%)	Cyclists(%)	Cars(%)		Pedestrians(%)		Cyclists(%)	
PointNet-STD [35]	**99.17**	**83.51**	**77.25**	93.45	-5.72	73.79	-9.72	61.80	-15.45
PointNet-MIX [35]	95.63	78.28	67.38	**95.63**	-	**78.28**	-	67.38	-
Expert Gate [26]	96.60	78.14	74.54	93.40	**-2.97**	71.26	-6.88	63.10	-11.44
3D-DEN [28]	95.43	75.24	69.45	91.90	-3.53	69.47	-5.77	59.33	-10.12
Ours	97.31	78.74	75.28	93.90	-3.41	73.50	**-5.24**	**68.27**	**-7.01**

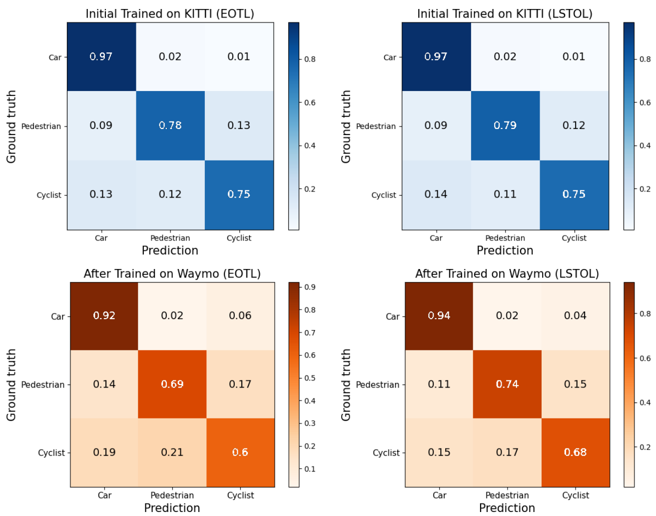

Fig. 4. Confusion matrices for performance comparison of EOTL and LSTOL. For each matrix, the ordinate represents the true label while the abscissa represents the predicted result. The darker the colour, the higher the proportion of correct classifications.

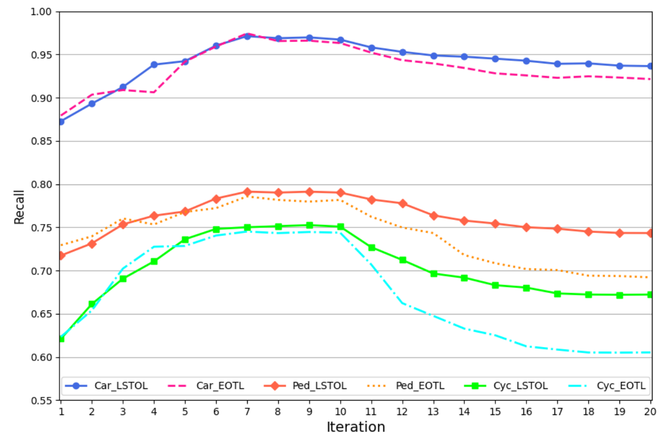

Fig. 5. Recall curves for cars, pedestrians, and cyclists during the system transfer from KITTI to Waymo.

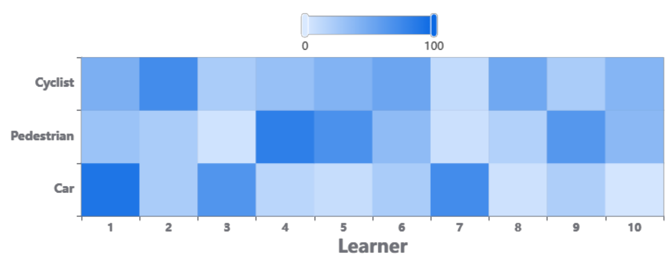

Fig. 6. Heatmap of "expert" opinions of each learner. Each column represents the distribution of high-confidence predictions for one learner across the test samples, while each row represents high-confidence results produced by different learners for the same class.

in EOTL, whose performance is often dominated by object classes (such as cars) that are easy to detect and have a larger number of learning examples than other classes. It can also be seen that during the learning process on the KITTI dataset, LSTOL shows better stability than EOTL. This reveals that the former can be used not only for cross-environment deployment of autonomous vehicles, but may also be suitable for long-term deployment in changing environments [12].

D. Short-term Learner Assessment

At the micro level, we focus on the performance and knowledge acquisition of each short-term learner. The LSTOL model in our experiments contains ten short-term learners, chosen to balance real-time performance and accuracy. Our experiment revealed a generally positive correlation between the number of short-term learners and performance. However, as the distribution of data stabilizes, the performance gains from adding more learners become less significant. Each learner uses ORF with consistent parameters, i.e., $trees = 100, depth = 50, epochs = 20, split\_threshold = 50$. After 20 iterations, each learner's predictions on the KITTI test set were evaluated, shown in Fig. 6. After filtering by the HRE mechanism, some learners' predictions are considered as expert opinions in the final determination. The darker the color of the square in the heatmap, the higher the frequency of participating in the final prediction of that class, and vice versa. For example, Learner 1 was more involved in cars, but also contributed to the pedestrians and cyclists.

It can be seen from Fig. 6 that each learner has a different focus on the knowledge they learn. The predictions of the first few learners show a dispersed distribution. As more new samples, such as previously unseen pedestrians and cyclists, appear in the scene, the learning goals of the learners begin to differentiate, e.g., the 2nd and 4th learners begin to focus more on pedestrians and cyclists. Starting with the 7th learner, we can see that the dataset changes, as the learner starts learning more from new examples of cars and subsequently new pedestrians and cyclists. It can also be seen that the learner trained in Waymo provides lower confidence predictions for samples in the KITTI test set than the one trained in KITTI.

VI. Conclusion

This paper introduced the Long-Short-Term Online Learning (LSTOL) framework, designed to prevent catastrophic forgetting and maintain stable performance for autonomous vehicles across environments during long-term operation. The framework's effectiveness is demonstrated through experiments on the KITTI and Waymo datasets for road participant classification.

Despite the promising results, the framework still has several limitations. One potential issue is that the need for priors (e.g., the number of short-term learners) to adapt to data distribution changes. In cases with new object categories or drastic changes in sensor data (e.g., weather or sensor failures), the framework may struggle to adapt without forgetting prior knowledge. Furthermore, the framework's adaptability can be improved, as it may not fully capture subtle differences in data distributions and could be optimized for dynamic environments.

Future work will involve isolating or replacing various components of the LSTOL framework to gain a clearer understanding of their sensitivity in preventing catastrophic forgetting. Additionally, we will explore mechanisms to dynamically control the number of short-term learners and test the framework on other downstream tasks, such as human-aware navigation [22], [36], to demonstrate its generality.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of CVPR*, 2016, pp. 770–778.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[3] Z. Yan, L. Sun, T. Krajnik, T. Duckett, and N. Bellotto, "Towards long-term autonomy: A perspective from robot learning," in *Proceedings of the AAAI-23 Bridge Program on AI & Robotics*, 2023.

[4] R. Yang, Z. Yan, Y. Ruichek, and T. Yang, "Efficient online transfer learning for 3d object classification in autonomous driving," in *Proceedings of ITSC*, 2021, pp. 2950–2957.

[5] R. Yang, Z. Yan, T. Yang, Y. Wang, and Y. Ruichek, "Efficient online transfer learning for road participants detection in autonomous driving," *IEEE Sensors Journal*, 2023.

[6] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgetting in gradient-based neural networks," *arXiv preprint arXiv:1312.6211*, 2013.

[7] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[8] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[9] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[10] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of CVPR*, 2012.

[11] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of CVPR*, 2020, pp. 2446–2454.

[12] Z. Yan, L. Sun, T. Krajnik, and Y. Ruichek, "EU long-term dataset with multiple sensors for autonomous driving," in *Proceedings of IROS*, 2020, pp. 10 697–10 704.

[13] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*, 1989, vol. 24, pp. 109–165.

[14] R. Ratcliff, "Connectionist models of recognition memory: constraints imposed by learning and forgetting functions." *Psychological review*, vol. 97, no. 2, p. 285, 1990.

[15] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[16] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proceedings of ICML*, 2017, pp. 3987–3995.

[17] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell, "Progress & compress: A scalable framework for continual learning," in *Proceedings of ICML*, 2018, pp. 4528–4537.

[18] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in neural information processing systems*, vol. 30, 2017.

[19] H. Wang, W. Xiong, M. Yu, X. Guo, S. Chang, and W. Y. Wang, "Sentence embedding alignment for lifelong relation extraction," *arXiv preprint arXiv:1903.02588*, 2019.

[20] J. Ramapuram, M. Gregorova, and A. Kalousis, "Lifelong generative modeling," *Neurocomputing*, vol. 404, pp. 381–400, 2020.

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

[22] T. Vintr, J. Blaha, M. Rektoris, J. Ulrich, T. Roucek, G. Broughton, Z. Yan, and T. Krajnik, "Toward benchmarking of long-term spatio-temporal maps of pedestrian flows for human-aware navigation," *Frontiers in Robotics and AI*, vol. 9, p. 890013, 2022.

[23] T. Vintr, Z. Yan, T. Duckett, and T. Krajnik, "Spatio-temporal representation for long-term anticipation of human presence in service robotics," in *Proceedings of ICRA*, 2019, pp. 2620–2626.

[24] L. Sun, Z. Yan, S. M. Mellado, M. Hanheide, and T. Duckett, "3DOF pedestrian trajectory prediction learned from long-term autonomous mobile robot deployment data," in *Proceedings of ICRA*, 2018, pp. 5942–5948.

[25] B. Liu, X. Xiao, and P. Stone, "A lifelong learning approach to mobile robot navigation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1090–1096, 2021.

[26] R. Aljundi, P. Chakravarty, and T. Tuytelaars, "Expert Gate: Lifelong learning with a network of experts," in *Proceedings of CVPR*, 2017, pp. 3366–3375.

[27] K. Shmelkov, C. Schmid, and K. Alahari, "Incremental learning of object detectors without catastrophic forgetting," in *Proceedings of ICCV*, 2017, pp. 3400–3409.

[28] S. Jain and H. Kasaei, "3d_den: Open-ended 3d object recognition using dynamically expandable networks," *IEEE Transactions on Cognitive and Developmental Systems*, 2021.

[29] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *ICCV Workshops*, 2009, pp. 1393–1400.

[30] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of ICML*. PMLR, 2017, pp. 1321–1330.

[31] T. Yang, Y. Li, Y. Ruichek, and Z. Yan, "LaNoising: A data-driven approach for 903nm ToF LiDAR performance modeling under fog," in *Proceedings of IROS*, 2020, pp. 10 084–10 091.

[32] ——, "Performance modeling a near-infrared tof lidar under fog: A data-driven approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 227–11 236, 2021.

[33] L. Sun, M. Taher, C. Wild, C. Zhao, F. Majer, Z. Yan, T. Krajnik, T. J. Prescott, and T. Duckett, "Robust and long-term monocular teach-and-repeat navigation using a single-experience map," in *Proceedings of IROS*, 2021, pp. 2635–2642.

[34] L. Liu, C. Cappelle, and Y. Ruichek, "Day and night place recognition based on low-quality night-time images," in *Proceedings of ITSC*, 2021, pp. 1–6.

[35] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of CVPR*, 2017, pp. 652–660.

[36] I. Okunevich, V. Hilaire, S. Galland, O. Lamotte, L. Shilova, Y. Ruichek, and Z. Yan, "Human-centered benchmarking for socially-compliant robot navigation," in *Proceedings of ECMR*, 2023, pp. 1–7.