IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Adaptive Robot Traversability Estimation Based on Self-Supervised Online Continual Learning in Unstructured Environments

Hyung-Suk Yoon, *Graduate Student Member, IEEE*, Ji-Hoon Hwang, Chan Kim, E In Son, Se-Wook Yoo, *Graduate Student Member, IEEE*, and Seung-Woo Seo, *Member, IEEE*

***Abstract*—Traversability estimation is a core function for robot navigation in off-road unstructured environments and diverse research results have been published so far. One of the recent approaches is using the self-supervised learning (SSL) technique. SSL has been focused on as a breakthrough technique for situations where environments keep changing and thus traversability estimation is a challenging task. However, most of the research efforts based on SSL have several limitations: (i) they operate in an offline manner that is vulnerable to the domain distribution shift and therefore, they cannot be adaptive to the current navigation environment; and (ii) they do not take into consideration the aleatoric uncertainty of the dataset which is particularly critical in unstructured environments. In this letter, we propose an adaptive robot traversability estimation framework that considers the current navigation environment based on self-supervised online continual learning. In addition, we propose an algorithm called experience replay with uncertainty, which considers the aleatoric uncertainty of the dataset while training the traversability estimation model, thus enabling our framework to robustly estimate robot traversability. We validate our methods in various real-world environments using the Clearpath Husky robot and evaluate that our methods show better navigation performance than offline learning and rule-based methods. Moreover, we also evaluate that the proposed algorithm based on experience replay with uncertainty performs better for the benchmark dataset (ImageNet, CORe50) than the baseline algorithms.**

***Index Terms*—Field Robots, robotics and automation in agriculture and forestry, AI-enabled robotics, continual learning.**

I. Introduction

ROBOT traversability estimation (RTE) determines the area for the optimal navigation of the robot based on environment information obtained from sensors and it is especially challenging for unstructured environments because of atypicality [1]. Although rule-based methods comprising the use of nature (geometry) information such as height are one of the widely used methods [2], it is challenging to determine the optimal general rules to suit each nature condition and multiple robot figures in unstructured environments [3]. To handle this problem, learning-based methods, especially based on self-supervised learning (SSL), have been introduced [4], [5]. These methods estimate the traversability from the robot's perspective by learning the correlation between the nature information from the exteroceptive sensors and robot dynamics from the proprioceptive sensors.

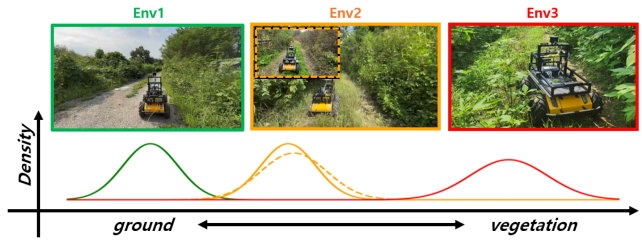


Fig. 1. Domain distributions for various unstructured environments. Env1: High ground density. Env3: High vegetation density. Env2: A medium of Env1 and Env3. In Env2, slight changes because of trivial natural factors such as rain cause a domain distribution shift within the environment.

SSL-based robot traversability estimation research has achieved various improvements [3], [4]; however, they still have several limitations. Firstly, they operate in an offline manner, that the RTE model is trained only with the offline dataset acquired in advance through human control [3] or random action [4]. This manner is vulnerable to the domain distribution shift, and therefore, it cannot be *adaptive* to the current navigation environment. This is fatal because the unstructured environments change significantly owing to trivial natural factors such as rain (Fig. 1). Secondly, previous SSL-based RTE studies did not consider the aleatoric uncertainty of the dataset [6] that occurs in unstructured environments due to unpaved terrain [7] or noise in the observation [8]. The aleatoric uncertainty inherent in the dataset could hinder the stable learning of the neural network model [9], and for navigation tasks, it degrades the navigation performance and could evoke irreversible damage. For example, if the robot incorrectly recognizes the border between traversable bushes and harsh rocks due to aleatoric uncertainty caused by similar appearance, a robot could step on the rocks and flip over.

In this letter, to address these limitations, we propose an adaptive robot traversability estimation (ARTE) framework that can be adaptive to the current navigation environment with uncertainty-aware self-supervised online continual learning. By

Manuscript received 7 December 2023; accepted 25 March 2024. Date of publication 8 April 2024; date of current version 17 April 2024. This letter was recommended for publication by Associate Editor M. Camurri and Editor J. Civera upon evaluation of the reviewers' comments. This work was supported in part by the Automation and Systems Research Institute (ASRI) and in part by the BK21 FOUR Program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2024. *(Corresponding author: Seung-Woo Seo.)*

The authors are with the Department of Electrical and Computer Engineering and ASRI, Seoul National University, Seoul 08826, South Korea (e-mail: hsyoon94@snu.ac.kr; hoons21@snu.ac.kr; chan_kim@snu.ac.kr; pingpang@snu.ac.kr; tpdnr1360@snu.ac.kr; sseo@snu.ac.kr).

Digital Object Identifier 10.1109/LRA.2024.3386451

continuously updating the RTE model using the online dataset, errors caused by domain distribution shifts can be prevented. Fine-tuning is one of the widely adopted techniques for continually learning the online dataset, but it is vulnerable to the catastrophic forgetting (CF) problem [10]. Therefore, we choose online continual learning (OCL) methods as the learning algorithm that is known to prevent the CF [11]. More specifically, we formulate our learning algorithm using the memory-based experience replay (ER) method [12] that prevents CF by storing a part of the online dataset in the memory and exploiting the memory for training. We select the ER based on the analysis that it generally demonstrates a better performance [11], and requires a lower computation power than other OCL algorithms [13].

Moreover, we propose an algorithm called experience replay with uncertainty (ERU), which considers the aleatoric uncertainty during training. The aleatoric uncertainty of the dataset can be estimated from the entropy of the RTE model output; this means that the higher the entropy, the higher the uncertainty and vice versa [9]. To prevent the RTE model from being deteriorated due to aleatoric uncertainty, we regularize the model parameter update according to the entropy of the model output. With ERU, the ARTE can robustly estimate the robot traversability. For our overall method, we design the nature cost, which is an input of the RTE model and consists of height, slope, and roughness costs using LiDAR. Next, we design the robot traversability cost, which is an output of the RTE model, using IMU. We design two costs under the assumption that the robot traversability cost would be an indicator that can comprehensively represent the three elements of the nature cost and prove its rationality through real-world robot experiments.

To verify the effectiveness of our methods, we conduct real-world navigation using the Clearpath Husky robot in various unstructured environments. By utilizing the path planner with RRTx [14] and path tracker with MPPI [15], we demonstrate better navigation performances than the offline RTE method and also the rule-based method. Moreover, to verify the effectiveness of ERU more specifically, we evaluate using the open benchmark datasets (ImageNet, Core50) [11] and show that ERU achieves better performance than the baseline OCL algorithms. In conclusion, the contributions of this letter can be summarized as follows:

1) We propose an adaptive robot traversability estimation (ARTE) framework that considers the current navigation environment and addresses the domain distribution shifts in unstructured environments with OCL.
2) We propose an algorithm called experience replay with uncertainty (ERU), which considers the aleatoric uncertainty of the dataset during training, thus enabling the ARTE to robustly estimate the robot traversability.
3) We demonstrate better performance of the ARTE and ERU compared to the baseline algorithms, and also verify that the robot traversability cost comprehensively represents the three elements of the natural cost.

II. Related Works

A. Self-Supervised Robot Traversability Estimation

The typical approach for self-supervised RTE in unstructured environments is to train a model that learns the correlation between the environment information recognized from the exteroceptive sensor, and proprioceptive sensor when passing through the recognized environments. Using this model for navigation, the SSL-based RTE interprets nature information from a robot's perspective. In [4], [16], [17], [18], a camera is used, and in [3], [19], LiDAR is used for the exteroceptive sensors. Meanwhile, the authors of [20], [21] attempted to design a self-supervised signal based on the difference between the predicted and actual dynamics. Thus far, the self-supervised RTE research in an unstructured environment has been mainly focused on sensor selection and utilization for their domain or the method of designing self-supervised signals elaborately.

It is also necessary to understand that the place at which the learned RTE model operates and is verified in this letter is an unstructured environment wherein many environmental changes occur due to weather changes and uncertain observations occur because of terrain or sensor noise. However, previous studies did not consider these aspects in depth. Firstly, in most previous studies, RTE models have been trained in an offline manner, wherein the model is trained using the offline dataset that is acquired in advance for inference or testing. In an unstructured environment wherein environmental changes occur frequently, this approach can result in a distribution shift between the training and inference datasets. To address this problem, we propose the ARTE framework that prevents distribution shifts through OCL.

Secondly, previous studies on self-supervised RTE in unstructured environments had not considered the aleatoric uncertainty while training the RTE model. However, some other studies argued that it is important to consider the aleatoric uncertainty in unstructured environments. In [7], the authors addressed the aleatoric uncertainty problem caused by the terrain by estimating the uncertainty with the Gaussian kernel smoothing (GKS) to select an optimal path with minimum uncertainty. In [8], the authors addressed the aleatoric uncertainty problem caused by sensor noise or occlusion by estimating the uncertainty using Conditional Value-at-Risk (CVaR) to generate a safety margin. While problems in these studies have been solved, they require additional procedures (GKS or CVaR) for uncertainty estimation, which are handled as processes that are separate from RTE, and which increase the computation burden of the RTE operation. To address this limitation, we propose the ERU, which is a methodology that simultaneously estimates the traversability and aleatoric uncertainty. In summary, we propose ARTE with ERU to handle two limitations of the previous SSL-based RTE research in unstructured environments.

B. Domain-Incremental Online Continual Learning

The taxonomy of continual learning (CL) can be broadly divided into two criteria. (i) Task- vs. domain- vs. class-Incremental CL. Task-incremental CL has the id of the given task that the others do not, and to maximally utilize this setting, in many studies, a multi-head neural network structure is composed [22], [23]. Domain-incremental CL takes an input from multiple domains with the fixed same class distribution [24], [25]. In class-incremental CL, both the input domain and label class distribution change [26], [27]. (ii) Batch CL vs. Online CL. While batch continual learning takes the full batch of a new dataset at once, online continual learning takes only a single instance or mini-batch of a new dataset [11]. There are three mainstream taxonomies in OCL. Memory-based methods [12], [28] exploit a small-size memory buffer to store the previous datasets. Regularization-based methods [10], [29] adjust the update of

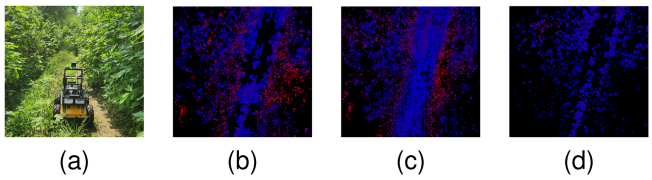


Fig. 2. Three cost components of nature cost C_n. (a) Husky robot. (b) Height costmap. (c) Slope costmap. (d) Roughness costmap.

the neural network for a given new dataset to prevent catastrophic forgetting. Parameter-isolation-based methods [30], [31] select an important or meta-parameter that dominates the neural network and adjust the update of those parameters. We design our method as domain-incremental online continual learning, with a fixed size class number, and for immediate adaptiveness. Specifically, we formulate our method using ER as it generally demonstrates a better performance [11] and faster computation speed [13] than other OCL algorithms. Based on the ER, we propose the ERU that can simultaneously measure the aleatoric uncertainty with a single pass of the RTE model [9], which reduces the computation burden.

III. Preliminaries

Memory-Based Experience Replay: As stated in Section II-B, the memory-based experience replay is one of the widely researched areas for OCL. It utilizes a small-sized memory to store the previous data for handling CF problems. There are three steps for memory-based ER. The first step is memory retrieval, which involves sampling a dataset from the memory. The second step comprises training the neural network model with a databatch composed of new data and the memory data. The third step comprises memory update, which involves updating the memory by appending specific parts of new data while discarding an equal size of appended data from the memory dataset if needed. In this letter, we follow MIR [28] for memory retrieval and ER [12] for memory update. MIR retrieves the top-k datasets in the memory with the criterion c as shown in (1):

$$c(m_i) = l(\Psi_{\theta^v}(x_i), y_i) - l(\Psi_\theta(x_i), y_i), \tag{1}$$

where $m_i = (x_i, y_i) \in \mathbb{M}$ and Ψ_{θ^v} is a virtually updated neural network of Ψ_θ with new data, and loss function l. Using this process, the MIR prevents CF by selecting the memory dataset that exhibits a large increase in the loss (maximal interference) for the neural network updated with the new data [28]. The ER updates the memory $\mathbb{M}$ using reservoir sampling [32] as shown in (2):

$$\mathbb{M}[j] \leftarrow \text{new data, if } j < \text{size}(\mathbb{M}) \tag{2}$$

With N, the number of the dataset observed until now, j is a randomly sampled integer from 0 to N. Reservoir sampling theoretically guarantees that the stored memory after the dataset streaming with the reservoir sampling becomes equal to the memory composed of random sampling from the whole dataset. Based on MIR for memory retrieval and ER for memory update, we propose an aleatoric uncertainty-aware OCL algorithm, called ERU, which targets the neural network training step.

Algorithm 1: ARTE with ERU

Initialize memory $\mathbb{M}$, traversability estimation model Π_θ
while navigation timestep t **do**
  Generate online databatch $B^{on} = \{C_n, C_r\}$
  Retrieve memory databatch B^{mem} with MIR
  $\hat{C}_r = \Pi_\theta(B^{on} \cup B^{mem})$
  Prediction loss $L_{CE} = CrossEntropy(C_r, \hat{C}_r)$
  Uncertainty $\sigma = Entropy(\hat{C}_r)$
  Total loss $L = L_{CE} + \lambda e^{\frac{\sigma}{\tau}} ||\theta||_2^2$
  Update Π_θ: $\theta_{t+1} \leftarrow \theta_t - \frac{\partial L}{\partial \theta_t}$
  Update $\mathbb{M}$ with reservoir sampling
end while

IV. Methods

A. Problem Definition and Overview

In this letter, we attempt to address two main problems of self-supervised RTE methods in unstructured environments. Firstly, previous studies on the SSL-based RTE in unstructured environments operate in an offline manner that is vulnerable to the domain distribution shift and therefore, they cannot be adaptive to the current navigation environment. Secondly, previous studies on the SSL-based RTE in unstructured environments also did not take into account the aleatoric uncertainty of the dataset which could destabilize the learning of the RTE model [9] and degrade the navigation performance. Hereafter, the aleatoric uncertainty is referred to as uncertainty for clear notation.

To address these problems, we propose an adaptive robot traversability estimation (ARTE) framework based on self-supervised online continual learning that considers uncertainty with our proposed algorithm, experience replay with uncertainty (ERU). ARTE first generates a self-supervised dataset tuple of $\{C_n, C_r\}$. C_n: The nature cost that the robot has actually contacted, and it is composed of the grid of LiDAR information of the environments. C_r: The robot traversability cost measured from the IMU sensor while traversing the C_n. This is specifically explained in Section IV-B. The ARTE then continuously updates the traversability estimation model Π_θ by ERU with C_n as an input and C_r as an output, which is specifically explained in Section IV-C.

B. Self-Supervised Dataset Generation

The self-supervised dataset comprises a nature cost C_n input and a robot traversability cost C_r output of the RTE model. First, we form a LiDAR grid world using the LiDAR point clouds (Fig. 2). Based on this, we compose C_n using three components: height, slope, and roughness. The height cost is obtained based on the average height of a point cloud in a grid. The slope cost is obtained based on the average of the surface normals of a point cloud in a grid. The roughness cost is obtained based on the standard deviation of the z values of a point cloud in a grid.

With the three height, slope, and roughness costs components, each having their own channel layer set, we define C_n as the partial patch of size $N \times N \times 3$ of the whole grid world. N is the size of the patch corresponding to the area where the robot actually traversed, and 3 indicates three channels. In particular, we extract four mini patches corresponding to each wheel with the size $\frac{N}{2} \times \frac{N}{2} \times 3$ and combine them according to the location

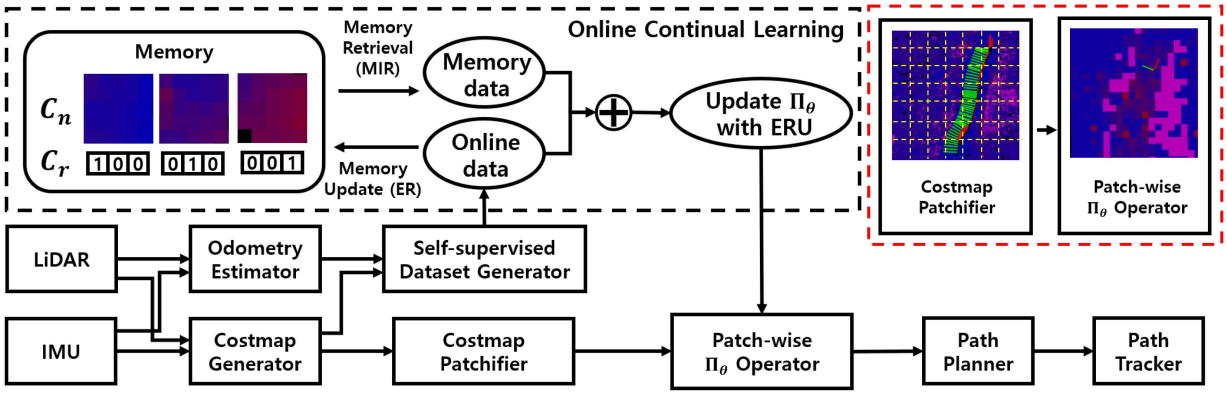


Fig. 3. Overall framework of ARTE with ERU. C_n costmap is generated from the costmap generator and the costmap patchifier slices C_n into patches of the same size as the input of Π_θ. All patches are then forwarded to Π_θ in the patch-wise Π_θ operator and C_r costmap is generated. The examples of this procedure are indicated in the red dotted box in the upper right. Next, with a path planner and path tracker that uses C_r costmap, Husky moves forward. **(Online Continual Learning)** A self-supervised online dataset $\{C_n, C_r\}$ is acquired using self-supervised dataset generator while navigation. With the online data and memory data, Π_θ is updated with the ERU. Both procedures operate continually until the end of the navigation.

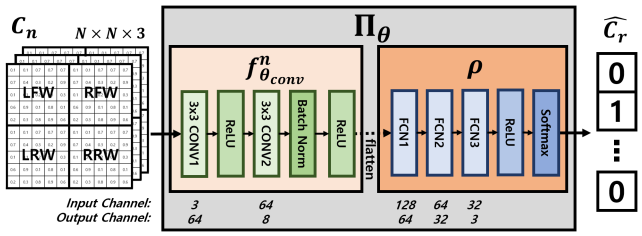


Fig. 4. Structure of the RTE model Π_θ. C_n patches for each wheel (Left Front Wheel or Right Rear Wheel) are placed in their respective positions. The output is C_r with size k.

of each wheel. We thus obtain a single C_n grid patch of the size of $N \times N \times 3$ as in Fig. 4.

Next, we define C_r as the robot traversability cost. We measure C_r from the bandpower of the IMU linear acceleration along the z-axis while traversing C_n, which is similar to the approach described in [1]:

$$C_r = \int_{\phi_{\min}}^{\phi_{\max}} PSD_z(\phi) d\phi, \tag{3}$$

where PSD_z is the power spectral density computed using Welch's method [33], and $\phi_{\min}$ and $\phi_{\max}$ indicates the frequency range for computing C_r. We then classify C_r into k classes and design C_r as a one-hot encoded vector of size k which is shown in Fig. 4. In this letter, we set $k = 3$ for mild, moderate, and harsh. The reason for designing C_r as a one-hot encoded vector is to estimate the uncertainty from the entropy of the network output for ERU, which is introduced in Section IV-C. Through this process, we generate a self-supervised online dataset $\{C_n, C_r\}$.

C. ARTE With ERU

RTE Model Π_θ: In this section, we introduce the method of using this online dataset to continually train the RTE model Π_θ, which is parameterized by the neural network θ. Π_θ takes C_n as an input and generates C_r as an output with two steps which are described in Fig. 4. First, as C_n has a 3D structure of $H \times W \times C$, it first passes through the nature feature extractor $f^n_{\theta_{conv}}$ composed of the convolutional neural networks (CNNs) with the parameter θ_{conv}. The nature feature extractor $f^n_{\theta_{conv}}$ consists of two convolutional layers. Each layer utilizes a 3×3 kernel and stride size 1. The first convolutional layer (CONV1) has an input channel size of 3, corresponding to the channel size of C_n, and an output channel size of 64. The second convolutional layer (CONV2) takes 64 channels as the input and outputs 8 channels.

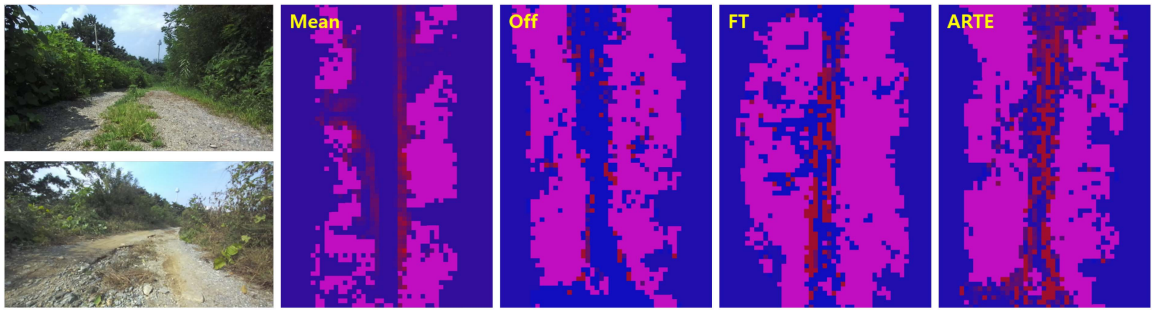
Next, the flattened output of $f^n_{\theta_{conv}}(C_n)$ is forwarded to the fully connected networks and softmax classifier (ρ) and generates $\hat{C_r}$, the prediction of C_r, as an output. In summary, our RTE model Π_θ operates as shown in (4):

$$\Pi_\theta : \hat{C_r} = \rho\{f^n_{\theta_{conv}}(C_n)\} \tag{4}$$

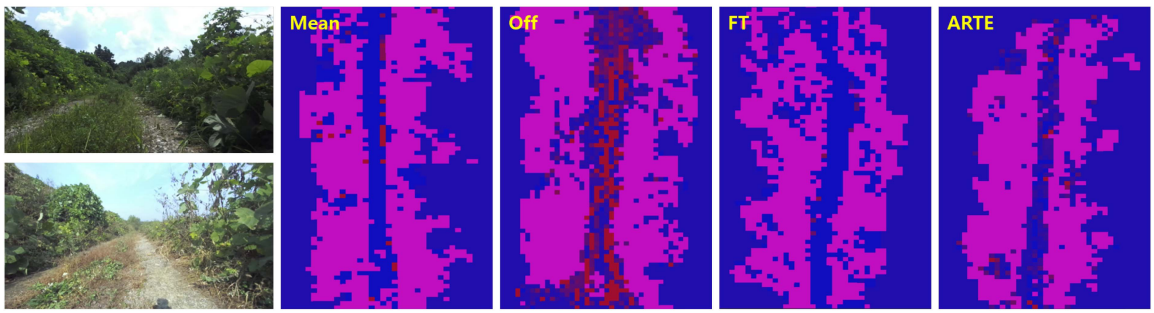
ERU: To train Π_θ while considering the dataset uncertainty, we propose our OCL algorithm, ERU. We estimate the uncertainty of the input dataset by measuring the entropy of $\hat{C_r}$ as $\sum_{i=0}^{k-1} -c_r^i \log c_r^i$, where $\hat{C_r} = \{c_r^0, c_r^1, \ldots, c_r^{k-1}\}$ [9]. When Π_θ can properly predict $\hat{C_r}$ for a given C_n, it exhibits a biased distribution indicating a discriminatively high probability to the prediction value. Therefore the entropy of the $\hat{C_r}$ is low. However, when Π_θ cannot properly predict $\hat{C_r}$ because of the uncertainty, it exhibits a less biased distribution than the previous case, and therefore the entropy of $\hat{C_r}$ is high. We measure the dataset uncertainty σ from the entropy of $\hat{C_r}$ with the property that the prediction performance of Π_θ is affected by the dataset uncertainty. Next, we use this uncertainty σ to regularize the update of Π_θ to prevent Π_θ from being interrupted by the uncertainty. We can thus design the training loss function L of Π_θ as shown in (5):

$$L = L_{CE}(C_r, \hat{C_r}) + \lambda e^{\frac{\sigma}{\tau}} ||\theta||_2^2, \tag{5}$$

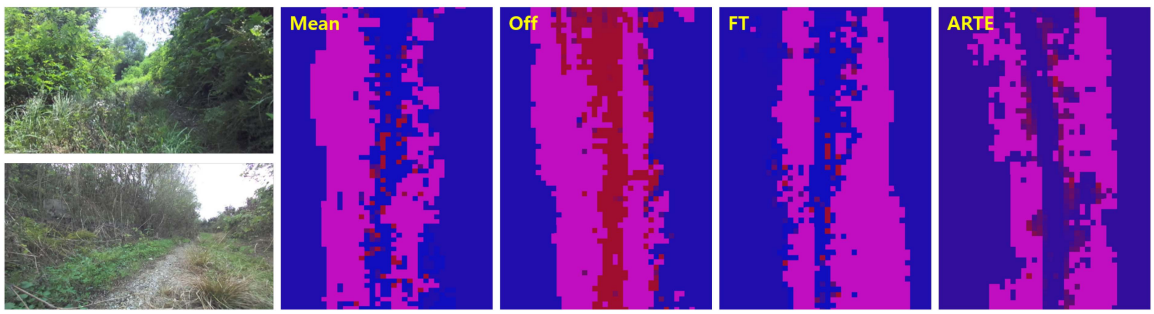
where $L_{CE}(\cdot, \cdot)$ indicates the cross-entropy loss, $||\theta||_2^2$ indicates the l_2-norm, and λ and τ are hyperparameters for the regularization term. We set λ as 1×10^{-3} and τ to 10 in our experiments. The high aleatoric uncertainty of the dataset implies that it inherently has many random, or unpredictable properties. Therefore, it is undesirable to directly combine the uncertainty σ with the prediction loss (L_{CE}). Instead, we combine σ with the regularization term (l_2-norm). Large uncertainty increases the influence of the regularization term and reduces the influence of



(a) Estimated traversability costmaps in Env1.



(b) Estimated traversability costmaps in Env2.



(c) Estimated traversability costmaps in Env3.

Fig. 5. Estimated traversability costmap for three real-world environments. For each figure, the top left image shows the previous environment (offline dataset) and the bottom left image shows the current environment (online dataset). The violet-colored grid indicates a constraint map for the path planner that is estimated from the LiDAR height information for robot safety. The blue color indicates lower cost while the red color indicates higher cost.

the input for the model parameter update. As a result, the ERU can generally and robustly learn despite the dataset uncertainty.

ARTE with ERU: We train the RTE model Π_θ with ERU and note that this procedure is conducted using a memory-based OCL method, which requires memory retrieval and memory update. We adopt the MIR [24] for memory retrieval and ER [12], or reservoir sampling, for memory update. First, a self-supervised online dataset $\{C_n, C_r\}$ is generated and a training databatch is composed by combining the online data and memory data retrieved by the MIR. Then Π_θ is trained and updated with the ERU. Finally, the memory is updated with a part of the new online data with reservoir sampling, and this is repeated during navigation.

We have thus far discussed the proposed method, the ARTE with ERU. The overall procedures are also specifically described in Fig. 3 and Algorithm 1. By continuously repeating these procedures during navigation, the ARTE with ERU can adaptively and robustly estimate the robot traversability while considering the current online navigation environment and the aleatoric uncertainties of the dataset.

D. Inference With ROS and Husky Robot

To generate the self-supervised online dataset, the odometry of the robot is required to extract the actually contacted ground patch C_n. For this purpose, we design an odometry estimator with FAST-LIO [34]. Based on this, the self-supervised dataset generator extracts the patch from C_n with a grid size of $10 \times 10 \times 3$ which comprises $0.5\,\text{m} \times 0.5\,\text{m} \times 3$ channels in the real world, considering the wheel size of the Clearpath Husky robot. Meanwhile, the costmap patchifier extracts multiple patches of C_n. Next, the piece-wise Π_θ operator passes the patches through Π_θ, and we then generate the C_r costmap that is filled with the corresponding C_r value for each grid patch (Fig. 3). Moreover, we additionally use the constraint grid map for the inference to guarantee the safety of the robot in unstructured environments. The sliced patch that has a higher height value in C_n than the threshold is set as the constraint, and the separate constraint map with the same resolution as the C_r costmap is generated (violet grids in Fig. 5). At last, the costmap with the constraint map added over the C_r costmap, which reflects the robot safety

more conservatively, is input to the path planner. We implement our methods using ROS and Python, and to guarantee real-time operation, the ARTE framework is operated at 8 Hz in an Intel-i7 CPU including the online learning time.

V. Experiments

A. Experimental Setup

We demonstrate the ARTE framework using the Clearpath Husky robot in three types of real-world unstructured environments (Fig. 1). Env1: Environment with a high proportion of ground and a low proportion of vegetation. Env3: Environment with a low proportion of ground and a high proportion of vegetation. Env2: Medium version of Env1 and Env3.

We set three baselines for the ARTE. First, in the mean setting (Mean), the costmap is measured from the weighted average of the three channel values of C_n, the height, slope, and roughness. This method is one of the widely used rule-based methods and requires meticulous tuning of weights for each environment for optimal performance. However, considering the problem definitions, we select the easier way by setting to equal weight $\frac{1}{3}$ which is a simple average. Second, in an offline setting (Off), the traversability estimation model is trained only using 500 offline datasets for each environment and does not learn the online navigation dataset. That is, for example, it estimates the robot traversability in ENV3 with the RTE model trained from the offline Env3 dataset that is collected in advance. At last, in a fine-tuning setting (FT), the traversability estimation model is first trained from the offline dataset and then learns the online dataset with a simple fine-tuning technique. We used 500 online datasets to update the FT and ARTE models for each environment. Both offline and online datasets do not reflect unnecessary elements such as slippage due to the robot starting and stopping. We set the FT as a baseline to compare it with Off and determine how the CF affects the navigation performance. Similarly, the ARTE is based on pre-trained Off models for each environment [35].

Evaluation Metrics: The final output of our methods is a costmap. However, the evaluation between the costmaps requires a ground truth cost map, which is difficult to produce. Therefore, following [4], [36], and as a better costmap induces better navigation performance, we evaluate the ARTE based on navigation performance and motion-level metrics. For this purpose, we utilize RRTx [14] as a path planner and MPPI [15] as a path tracker and select the perceived vehicle motion intensity (PMI) score [37], [38] to quantitatively evaluate our method. It is measured from the inertial motion values as shown in (6):

$$\mu_{\phi_{\max}} = w_a a_{\max} + w_j j_{\max} + w_{aj}(a_{\max} \times j_{\max}) \quad (6)$$

where $a_{\max}$ and $j_{\max}$ are the maximum acceleration and jerk value respectively, and w_a, w_j, w_{aj} indicate each weights. However, as navigation is a long-term task, we judge that an evaluation with only a single maximum value is insufficient. Moreover, the direction of the acceleration or jerk could be ignored because the direction actually does not affect the perceived comfort. Therefore we designed a variation version, the PMI-abs-top$N\%$ as shown in (7):

$$\mu_{\phi_{N\%}} = w_a\overline{|a_{N\%}|} + w_j\overline{|j_{N\%}|} + w_{aj}(\overline{|a_{N\%}|} \times \overline{|j_{N\%}|}) \quad (7)$$

where $\overline{|a_{N\%}|}$ and $\overline{|j_{N\%}|}$ are the mean of the top $N\%$ absolute values of the acceleration and jerk, respectively. We set N to 1 and 5 in our experiments and notate the original PMI score as PMI-Max in this letter. As in [37], we compute the scores for four components, z, roll, pitch, and yaw.

Benchmark Evaluation of ERU: To specifically evaluate the effectiveness of the ERU in terms of dataset uncertainty, we conduct experiments on image classification tasks. As analyzed in Section II-B, our problem can be classified as domain-incremental OCL and the evaluation is conducted based on [11]. The authors in [11] proposed the NonStationary-MiniImageNet (NS-ImageNet) and Core50-NI benchmarks for

TABLE I
Comparison Among Experiment Algorithms

Algorithms	Mean	Off	FT	ARTE
SSL-based RTE model	×	✓	✓	✓
Online dataset	×	×	✓	✓
Online continual learning	×	×	×	✓

TABLE II
PMI Score Results in Three Environments

Env		**Env1**			
Metrics		Mean	Off	FT	ARTE
PMI -Max	z	1.615 (0.15)	2.247 (0.48)	1.566 (0.13)	**1.396 (0.19)**
	roll	0.382 (0.05)	0.484 (0.23)	0.401 (0.06)	**0.362 (0.06)**
	pitch	0.166 (0.01)	0.163 (0.01)	**0.162 (0.02)**	0.166 (0.02)
	yaw	0.184 (0.02)	0.189 (0.03)	0.201 (0.06)	**0.170 (0.03)**
PMI -abs -Top1%	z	1.267 (0.06)	1.772 (0.33)	1.384 (0.20)	**1.016 (0.05)**
	roll	0.364 (0.06)	0.367 (0.11)	0.364 (0.04)	**0.333 (0.06)**
	pitch	0.158 (0.01)	0.165 (0.00)	0.163 (0.01)	**0.151 (0.01)**
	yaw	0.173 (0.03)	0.182 (0.02)	0.193 (0.02)	**0.149 (0.02)**
PMI -abs -Top5%	z	1.047 (0.03)	1.447 (0.18)	1.162 (0.02)	**0.854 (0.02)**
	roll	0.323 (0.05)	0.319 (0.07)	0.322 (0.05)	**0.298 (0.05)**
	pitch	0.145 (0.07)	0.151 (0.00)	0.151 (0.01)	**0.138 (0.01)**
	yaw	0.157 (0.03)	0.162 (0.02)	0.178 (0.02)	**0.139 (0.02)**
Env		**Env2**			
Metrics		Mean	Off	FT	ARTE
PMI -Max	z	0.847 (0.13)	0.907 (0.01)	0.687 (0.27)	**0.446 (0.09)**
	roll	0.323 (0.06)	0.307 (0.01)	0.436 (0.08)	**0.259 (0.07)**
	pitch	0.112 (0.01)	0.118 (0.00)	0.119 (0.00)	**0.102 (0.01)**
	yaw	0.160 (0.04)	0.167 (0.04)	**0.155 (0.00)**	0.168 (0.04)
PMI -abs -Top1%	z	0.716 (0.05)	0.720 (0.03)	0.629 (0.27)	**0.399 (0.09)**
	roll	0.263 (0.02)	0.289 (0.00)	0.311 (0.01)	**0.261 (0.07)**
	pitch	0.116 (0.01)	0.114 (0.01)	0.121 (0.01)	**0.096 (0.00)**
	yaw	0.156 (0.03)	0.163 (0.03)	**0.149 (0.00)**	0.155 (0.03)
PMI -abs -Top5%	z	0.559 (0.04)	0.561 (0.00)	0.519 (0.21)	**0.356 (0.07)**
	roll	**0.225 (0.01)**	0.253 (0.01)	0.265 (0.05)	0.242 (0.06)
	pitch	0.105 (0.01)	0.102 (0.01)	0.108 (0.00)	**0.090 (0.00)**
	yaw	0.143 (0.02)	0.150 (0.03)	0.133 (0.01)	**0.124 (0.03)**
Env		**Env3**			
Metrics		Mean	Off	FT	ARTE
PMI -Max	z	1.224 (0.50)	0.770 (0.31)	0.678 (0.11)	**0.519 (0.18)**
	roll	0.339 (0.02)	**0.331 (0.04)**	0.339 (0.02)	0.352 (0.08)
	pitch	0.138 (0.01)	0.145 (0.01)	0.153 (0.01)	**0.128 (0.02)**
	yaw	**0.171 (0.01)**	0.188 (0.03)	0.231 (0.01)	0.193 (0.05)
PMI -abs -Top1%	z	0.988 (0.39)	0.680 (0.24)	0.563 (0.08)	**0.469 (0.15)**
	roll	0.312 (0.02)	0.321 (0.02)	**0.303 (0.03)**	0.338 (0.07)
	pitch	0.147 (0.02)	0.133 (0.02)	0.137 (0.01)	**0.128 (0.02)**
	yaw	0.159 (0.01)	0.158 (0.01)	0.187 (0.06)	**0.148 (0.04)**
PMI -abs -Top5%	z	0.806 (0.32)	0.612 (0.23)	0.502 (0.07)	**0.424 (0.13)**
	roll	0.279 (0.01)	**0.275 (0.02)**	0.288 (0.03)	0.307 (0.07)
	pitch	0.135 (0.02)	0.122 (0.01)	0.129 (0.01)	**0.118 (0.02)**
	yaw	0.148 (0.01)	0.144 (0.01)	0.166 (0.05)	**0.129 (0.04)**

TABLE III
BENCHMARK EVALUATION RESULT OF ERU. IT INDICATES THE AVERAGE TEST ACCURACY (STANDARD DEVIATION) FOR 5 SEEDS

Dataset	NS-ImageNet-Noise		NS-ImageNet-Occlusion		NS-ImageNet-Blur		CORe50-NI	
Mem size	1k	10k	1k	10k	1k	10k	1k	10k
ER [12]	44.862 (0.25)	48.676 (0.19)	49.918 (0.41)	53.750 (0.37)	16.194 (0.15)	20.026 (0.04)	73.052 (0.66)	75.656 (0.66)
MIR [28]	42.300 (0.47)	49.834 (0.32)	50.192 (0.14)	54.058 (0.44)	16.166 (0.10)	21.266 (0.15)	74.604 (0.12)	75.716 (0.30)
ERU	**47.540 (0.18)**	**51.874 (0.15)**	**53.204 (0.12)**	**57.830 (0.32)**	**18.242 (0.26)**	**23.132 (0.20)**	**78.320 (0.37)**	**79.538 (0.51)**

domain-incremental OCL algorithms. The NS-ImageNet benchmark is based on the ImageNet dataset and three types of non-stationary components (noise, blur, and occlusion) are injected to transform the raw dataset into a domain-incremental task. Moreover, a dataset is divided into 10 tasks for each type, and therefore each task is composed of 5000 training images and 1000 test images. To better reflect the domain incrementality, the degree of non-stationary component injection increases over the task sequence. In addition, we also use the CORe50-NI dataset, which is processed in a manner similar to the NS-ImageNet. We used ResNet18 [39] for experiments and compared the performance with the ER [12] and MIR [28] which exhibit the best performance in [11]. More details may be referred to in [11].

B. Results and Discussions

Quantitative Evaluation of ARTE: The PMI score results are described in Table II. The bolded values indicate the lowest PMI score, which implies the best navigation performance and the values in parentheses represent the standard deviation. A lower score indicates a better navigation performance, and we can observe that the ARTE shows the best overall performance for the whole environment. In particular, for all the scores, we can find that a notable performance gap occurs in the z value more than other values (roll, pitch, yaw). We analyze that this result is obtained through our process of designing the C_r with a focus on the z value using (3).

Next, we analyze the results according to the environments. We can observe that in Env1 and Env2, the ARTE demonstrates a dominantly better performance while in Env3, the degree of dominance becomes slightly less than in the previous two environments. We interpret it because Env3 has a high density of vegetation, which generates severe uncertainty in both the online dataset generation and RTE model training phases. Despite these challenges, we verified that ARTE achieved competitive performance in the whole environment compared to the baselines.

At last, we analyze the performance of Off, FT, and ARTE through the effect of the dataset distribution shift and the CF in the RTE model learning. We summarize the three algorithms in Table I. Considering the result of Table II, we can conclude that the overall performance is superior in the order of the ARTE, FT, and Off. (i) Off vs. FT and ARTE: We can conclude that solving the distribution shift by training the RTE model with an online dataset enhances the model performance, which is demonstrated through the superior navigation performance. (ii) FT vs. ARTE: We can conclude that solving the CF using memory also enhances both the RTE model and navigation performance. Through the experiment result and the analysis, which demonstrate that the ARTE presents the best results, we can conclude that solving the distribution shift and the CF enhances the RTE model learning and navigation performance.

Qualitative Evaluation of ARTE: The costmap estimated after the online navigation is described in Fig. 5. Before the analysis, as the unstructured environments change due to weather conditions such as wind and repeated navigation (e.g. changing terrain shape and trampled bush), we first note that the overall shape of the costmap, especially the constraint map, appears slightly different in each trial even in the same environment. For Env1 in Fig. 5(a), we can see that the ground status becomes harsher in an online environment. This is why the road part (center vertical part) of Off shows low (blue) cost while ARTE shows high cost (red). For Env2 in Fig. 5(b), we can see that the environment becomes less harsh for an online environment. This is the reason the road part of Off shows high cost and ARTE shows low cost. At last, for Env3 in Fig. 5(c), we can first feel that the offline environment is extremely hard to navigate and this feeling is directly expressed in the Off costmap. The road part is entirely filled with the highest (reddest) cost because the robot experienced the harshest navigation when traversing an offline environment. However, the harshness of Env3 decreases significantly in an online environment and the actual feeling of navigation of the robot is described in the ARTE costmap as much milder.

At last, we confirmed that the loss stably converges to 0 without adversarial impacts, such as loss fluctuation, when training the RTE model for Off, FT, and ARTE. Based on [40], we determine that the robot traversability cost is not a noisy label for the nature cost, and finally, conclude that the robot traversability cost C_r comprehensively represents the three elements of natural cost C_n.

Benchmark Evaluation of ERU: The result of the benchmark evaluation of the ERU is presented in Table III. It shows that the ERU performs better than the other baseline algorithms for the whole benchmark. Moreover, we can generally assume that the three types of non-stationary components (noise, occlusion, and blur) for the NS-ImageNet dataset are equal to the sources of uncertainties in unstructured environments. C_n is a LiDAR-based value and various non-stationary objects in unstructured environments, such as bushes, could be recognized as similar to the noise for the LiDAR sensor (noise). In addition, there could be many fallen leaves on the ground that blind the ground information (occlusion). Besides, as the ground of unstructured environments is bumpy, it gives a vibration to the robot, which may cause the sensor configuration of the robot to become unstable (blur). Therefore, our experiment can be considered to prove sufficiently that the ERU operates robustly against various uncertainties in unstructured environments.

VI. CONCLUSION

In this letter, we propose the ARTE with ERU to address the distribution shift and aleatoric uncertainty challenges of the SSL-based RTE in unstructured environments. The ARTE continually learns the online dataset to prevent distribution shifts, and can eventually be adaptive to the current environment. In addition, with the ERU that considers the aleatoric uncertainty of the dataset while training the RTE model, the ARTE can estimate the robot traversability robustly. We evaluate that the

ARTE demonstrates a better navigation performance than the baselines for three types of real-world environments when using the Clearpath Husky robot. Based on the real-world experiment and stable learning of the RTE models, we also validate that the robot traversability cost comprehensively represents the three elements of the nature cost. In addition, we evaluate that the ERU shows better performance for domain-incremental OCL benchmark datasets than the baseline algorithms.

REFERENCES

[1] M. G. Castro et al., "How does it feel? Self-supervised costmap learning for off-road vehicle traversability," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 931–938.

[2] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter, "Navigation planning for legged robots in challenging terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 1184–1189.

[3] J. Seo, T. Kim, K. Kwak, J. Min, and I. Shim, "ScaTE: A scalable framework for self-supervised traversability estimation in unstructured environments," *IEEE Robot. Automat. Lett.*, vol. 8, no. 2, pp. 888–895, Feb. 2023.

[4] G. Kahn, P. Abbeel, and S. Levine, "BADGR: An autonomous self-supervised learning-based navigation system," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 1312–1319, Apr. 2020.

[5] J. Zürn, W. Burgard, and A. Valada, "Self-supervised visual terrain classification from unsupervised acoustic feature learning," *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 466–481, Apr. 2021.

[6] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision," in *Proc. 31st Int. Conf. Neural Inform. Process. Syst.*, 2017, pp. 5580–5590.

[7] H. Lee, J. Kwon, and C. Kwon, "Learning-based uncertainty-aware navigation in 3D off-road terrains," in *Proc. IEEE Int. Conf. Robot. Automat.*, London, United Kingdom, 2023, pp. 10061–10068.

[8] D. D. Fan, K. Otsu, Y. Kubo, A. Dixit, J. Burdick, and A.-A. Agha-Mohammadi, "Step: Stochastic traversability evaluation and planning for risk-aware off-road navigation," 2021, *arXiv:2103.02828*.

[9] Y. Gal et al., "Uncertainty in deep learning," vol. 1, Ph.D. thesis, Univ. Cambridge, 2016.

[10] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[11] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim, and S. Sanner, "Online continual learning in image classification: An empirical survey," *Neurocomputing*, vol. 469, pp. 28–51, 2022.

[12] A. Chaudhry et al., "On tiny episodic memories in continual learning," 2019.

[13] A. Prabhu et al., "Computationally budgeted continual learning: What does matter," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 3698–3707.

[14] M. W. Otte and E. Frazzoli, "Rrtx: Asymptotically optimal singlequery sampling-based motion planning with quick replanning," *Int. J. Robot. Res.*, vol. 35, no. 7, pp. 797–822, 2016. [Online]. Available: https://doi.org/10.1177/0278364915594679

[15] G. Williams, A. Aldrich, and E. Theodorou, "Model predictive path integral control using covariance variable importance sampling," vol. abs/1509.01149, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:14146342

[16] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, "Where should i walk? Predicting terrain properties from images via self-supervised learning," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1509–1516, Apr. 2019.

[17] O. Mayuku, B. W. Surgenor, and J. A. Marshall, "A self-supervised near-to-far approach for terrain-adaptive off-road autonomous driving," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 14054–14060.

[18] R. Schmid et al., "Self-supervised traversability prediction by learning to reconstruct safe terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 12419–12425.

[19] M. Stölzle, T. Miki, L. Gerdes, M. Azkarate, and M. Hutter, "Reconstructing occluded elevation information in terrain maps with self-supervised learning," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 1697–1704, Apr. 2022.

[20] M. V. Gasparino et al., "Wayfast: Navigation with predictive traversability in the field," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 10651–10658, Oct. 2022.

[21] A. Polevoy, C. Knuth, K. M. Popek, and K. D. Katyal, "Complex terrain navigation via model error prediction," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 9411–9417.

[22] R. Aljundi, P. Chakravarty, and T. Tuytelaars, "Expert gate: Lifelong learning with a network of experts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 2017, pp. 3366–3375.

[23] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Adv. Neural Inform. Process. Syst.*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paperfiles/paper/2017/file/f87522788a2be2d171666752f97ddebb-Paper.pdf

[24] M. J. Mirza, M. Masana, H. Possegger, and H. Bischof, "An efficient domain-incremental learning approach to drive in all weather conditions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3001–3011.

[25] T. Kalb, M. Roschani, M. Ruf, and J. Beyerer, "Continual learning for class- and domain-incremental semantic segmentation," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2021, pp. 1345–1351.

[26] S. Banerjee, V. K. Verma, T. Parag, M. Singh, and V. P. Namboodiri, "Class incremental online streaming learning," 2021. [Online]. Available: http://arXiv.org/abs/2110.10741

[27] M. Kang, J. Park, and B. Han, "Class-incremental learning by knowledge distillation with adaptive feature consolidation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16071–16080.

[28] R. Aljundi et al., "Online continual learning with maximal interfered retrieval," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 11872–11883.

[29] J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci.*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.

[30] M. Delange et al., "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7 pp. 3366–3385, Jul. 2022.

[31] S. Lee, J. Ha, D. Zhang, and G. Kim, "A neural Dirichlet process mixture model for task-free continual learning," in *Proc. Int. Conf. Learn. Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=SJxSOJStPr

[32] J. S. Vitter, "Random sampling with a reservoir," *ACM Trans. Math. Softw.*, vol. 11, no. 1, pp. 37–57, Mar. 1985.

[33] P. Welch, "The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Trans. Audio Electroacoust.*, vol. 15, no. 2, pp. 70–73, Jun. 1967.

[34] W. Xu and F. Zhang, "FAST-LIO: A. fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3317–3324, Apr. 2021.

[35] K.-Y. Lee, Y. Zhong, and Y.-X. Wang, "Do pre-trained models benefit equally in continual learning?," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 6485–6493.

[36] A. J. Sathyamoorthy, K. Weerakoon, T. Guan, J. Liang, and D. Manocha, "TerraPN: Unstructured terrain navigation using online self-supervised learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 7197–7204.

[37] X. Yao, J. Zhang, and J. Oh, "RCA: Ride Comfort-Aware Visual Navigation Via Self-Supervised Learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 7847–7852.

[38] H. H. B. Ksander, N. de Winkel, and F. Soyka, "The role of acceleration and jerk in perception of above-threshold surge motion," *Exp. Brain. Res.*, vol. 238,, pp. 699–711, 2020.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 770–778.

[40] B. Frenay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 845–869, May 2014.