IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 2 March 2023, accepted 16 May 2023, date of publication 24 May 2023, date of current version 1 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3279711

RESEARCH ARTICLE

Self-Supervised 3D Traversability Estimation With Proxy Bank Guidance

JIHWAN BAE[1], JUNWON SEO[1], TAEKYUNG KIM[1], HAE-GON JEON[2], KIHO KWAK[1], (MEMBER, IEEE), AND INWOOK SHIM[3], (Member, IEEE)

[1]Agency for Defense Development, Daejeon 34186, South Korea
[2]AI Graduate School and the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (GIST), Gwangju 61005, South Korea
[3]Department of Smart Mobility Engineering, Inha University, Incheon 22212, South Korea

Corresponding author: Inwook Shim (iwshim@inha.ac.kr)

This work was supported in part by the Agency for Defense Development and the Korea Institute for Advancement of Technologe (KIAT) grant funded by the Korea Government (MOTIE), (HRD Program for Industrial Innovation) under Grant P0020536.

ABSTRACT Traversability estimation for mobile robots in off-road environments requires more than conventional semantic segmentation used in constrained environments like on-road conditions. Recently, approaches to learning a traversability estimation from past driving experiences in a self-supervised manner are arising as they can significantly reduce human labeling costs and labeling errors. However, the self-supervised data only provide supervision for the actually traversed regions, resulting in epistemic uncertainty due to the lack of knowledge on non-traversable regions, also referred to as negative data. Negative data can rarely be collected as the system can be severely damaged while logging the data. To mitigate the uncertainty in the estimation, we introduce a deep metric learning-based method to incorporate unlabeled data with a few positive and negative prototypes. Our method jointly learns binary segmentation that reduces uncertainty in addition to the regression of traversability. To firmly evaluate the proposed framework, we introduce a new evaluation metric that comprehensively evaluates the segmentation and regression. Additionally, we construct a driving dataset 'Dtrail' in off-road environments with a mobile robot platform, which is composed of numerous complex and diverse representations of off-road environments. We examine our method on Dtrail as well as the publicly available SemanticKITTI dataset.

INDEX TERMS Self-supervised traversability, semantic segmentation, deep metric learning, mobile robots, autonomous driving.

I. INTRODUCTION

Estimating traversability for mobile robots is an important task for autonomous driving and machine perception. However, the majority of the relevant works focus on constrained road environments like paved roads which are all possibly observed in public datasets [1], [2], [3]. In urban scenes, road detection with semantic segmentation is enough [4], [5], but in unconstrained environments like off-road areas, the semantic segmentation is insufficient as the environment can be highly complex and rough [6] as shown in Fig. 1a. Several works from the robotics field have proposed a method to estimate the traversability cost in the unconstrained environments [7], [8], [9], [10], and to infer probabilistic traversability map with visual information such as image [11] and 3D LiDAR [6].

Actual physical state changes that a vehicle undergoes can give meaningful information on where it can traverse and how difficult it would be [12], [13], [14]. The data incorporating physical changes encountered by the vehicle itself are known as self-supervised data. Accordingly, self-supervised traversability estimation can offer more robot-oriented prediction [11], [15], [16]. Fig. 1c shows an example of the self-supervised traversability data. Previously, haptic inspection [11], [16] has been examined as traversability in the self-supervised approaches. These works demonstrate

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wei.

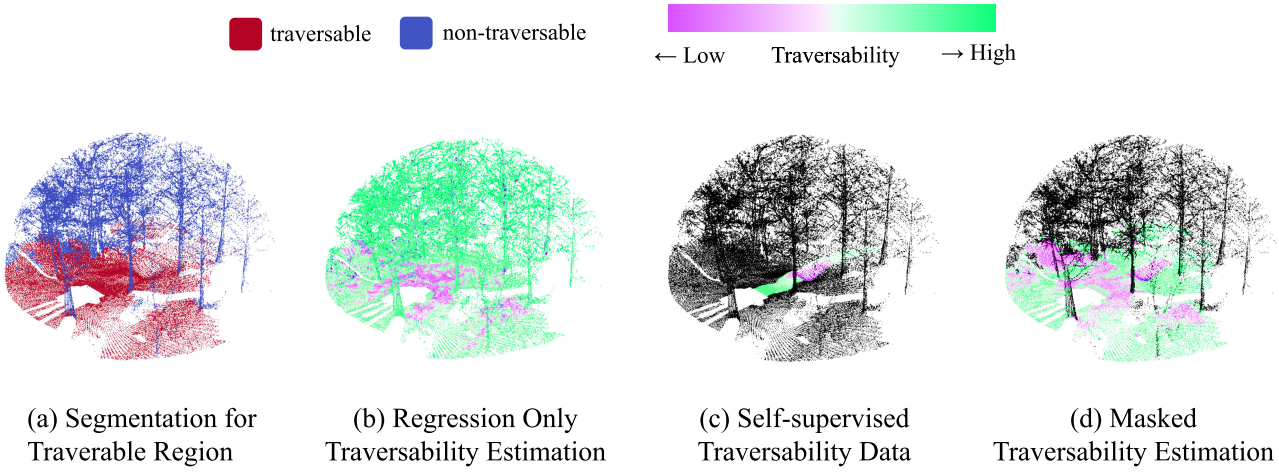


FIGURE 1. Illustration of motivation of our framework. The color map for the traversability shows that the higher (green), the traversability is, the easier to traverse, and the lower (purple), the traversability is, the harder to traverse. Self-supervised traversability estimation should be considered to minimize the epistemic uncertainty by filtering out the non-traversable regions to which no supervision is given in terms of traversability.

that learning self-supervised data is a promising approach for traversability estimation, but are only delved into the proprioceptive sensor domain or image domain. Additionally, supervision from the self-supervised data is limited to the actually traversed regions as depicted in Fig. 1c, thereby inducing an epistemic uncertainty when inferring the traversability on non-traversed regions. An example of such epistemic uncertainty is illustrated in Fig. 1b. Hazardous regions, such as trees and steep slopes, that are impossible to drive over are regressed with high traversability, which means they are easy to traverse. For safe navigation in off-road environments, such non-traversable areas with considerable uncertainty should be explicitly identified so that dependable prediction is ensured.

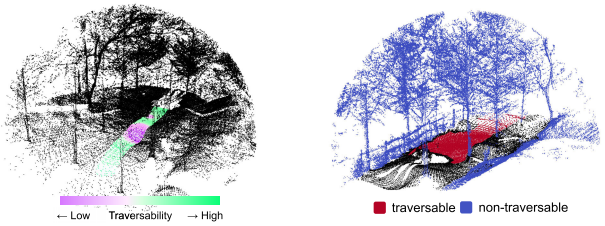
In this paper, we propose a self-supervised framework on 3D point cloud data for traversability estimation in unconstrained environments concentrated on alleviating epistemic uncertainty. Our goal is to learn a model that predicts traversability that is masked on regions with high uncertainty (as shown in Fig. 1d) leveraging self-supervised traversability data (Fig. 1c). To achieve this, we jointly learn semantic segmentation along with traversability regression via deep metric learning to filter out the non-traversable regions (see Fig. 1d.) Also, by harnessing the unlabeled data from the non-traversed area, we introduce the unsupervised loss similar to the clustering methods [17]. To better evaluate our task, we develop a new evaluation metric that can both evaluate the segmentation and the regression, while highlighting the false-positive ratio for reliable estimation. To test our method on more realistic data, we build an off-road vehicle driving dataset named '*Dtrail*.' Experimental results are both shown for Dtrail and SemanticKITTI [18] dataset. Ablations and comparisons with the other metric learning-based methods show that our method yields quantitatively and qualitatively robust results. Our contributions can be summarized in five points as follows:

- We introduce a self-supervised traversability estimation framework on 3D point clouds that mitigates the uncertainty by adopting deep metric learning and by actively utilizing self-supervised traversability data.
- We jointly learn the binary segmentation and the traversability regression to obtain reliable predictions in off-road environments.
- We also adopt the unsupervised clustering-based loss to utilize the advantage of the affluent unlabeled data in the self-supervised settings.
- We devise a new metric (Traversability Precision Error) to evaluate the self-supervised traversability estimation properly. Compared to the conventional metric (e.g., mIoU.) This metric can more appropriately measure the effect of the false positive occurrences on traversability estimation.
- We present a new 3D point cloud dataset for off-road mobile robot driving in unconstrained environments that includes IMU data synchronized with LiDAR.

II. RELATED WORKS

A. TRAVERSABILITY ESTIMATION

Traversability estimation is a crucial component in mobile robotics platforms for estimating where it should go. In the case of paved road conditions, the traversability estimation task can be regarded as a subset of road detection [5], [19] and semantic segmentation [20]. However, the human-supervised method is clearly limited in estimating traversability for unconstrained environments like off-road areas. According to the diversity of the road conditions, it is hard to determine the traversability of a mobile robot in advance by man-made predefined rules.



(a) Query (b) Support

FIGURE 2. Examples of our task data settings: query and support data. (a) is an example of query data. Unlabeled points are colored black, and non-black points indicate the robot's traversable region. Traversability is mapped only on the positive points. (b) is an example of support data. Traversable and non-traversable are manually labeled as red and blue, respectively. Only evident regions are labeled and used for the training in the support data.

Self-supervised approaches [16], [21], [22], [23] are suggested in the robotics literature to estimate the traversability using proprioceptive sensors such as inertial measurement and force-torque sensors [16]. Since these tasks only measured traversability in the proprioceptive-sensor domain, they do not affect the robot's future driving direction. To solve this problem, a study to predict terrain properties by combining image information with the robot's self-supervision has been proposed [11]. They identify the terrain properties from haptic interaction and associate them with the image to facilitate self-supervised learning. This work demonstrates promising outputs for traversability estimation, but it does not take epistemic uncertainty into account that necessarily exists in the self-supervised data. Furthermore, image data-based learning approaches are still vulnerable to illumination changes that can reduce the performance of the algorithms. Therefore, range sensors such as 3D LiDAR can be a strong alternative [24].

To overcome such limitations, we propose a self-supervised traversability estimation method based on 3D point clouds that can alleviate the uncertainty problem in unconstrained environments.

B. DEEP METRIC LEARNING

One of the biggest challenges in learning with few labeled data is epistemic uncertainty. To handle this problem, researchers proposed deep metric learning (DML) [25], which learns embedding spaces and classifies an unseen sample in the learned space. Several works adopt the sampled mini-batches called *episodes* during training, which mimics the task with few labeled data to facilitate DML [26], [27], [28], [29]. These methods with episodic training strategies epitomize labeled data of each class as a single vector, referred to as a prototype [20], [30], [31], [32], [33]. The prototypes generated by these works require non-parametric procedures and insufficiently represent unlabeled data.

Other works [34], [34], [35], [36], [37], [38], [39] develop loss functions to learn an embedding space where similar examples are attracted, and dissimilar examples are repelled. Recently, proxy-based loss [40] is proposed. Proxies are representative vectors of the training data in the learned embedding spaces, which are obtained in a parametric way [41], [42]. Using proxies leads to better convergence as they reflect the entire distribution of the training data [40]. A majority of the works [41], [42] provides a single proxy for each class, whereas SoftTriple loss [43] adopts multiple proxies for each class. We adopt the proxy-based DML loss, as traversable and non-traversable regions are represented as multiple clusters rather than a single one in the unstructured driving surfaces according to their complexity and roughness.

III. METHODS

A. OVERVIEW

Our proxy bank-guided self-supervised framework (PBG) aims to learn a mapping between point clouds to traversability. We call input data containing the traversability information as '*query*.' The traversable regions are referred to as the '*positive*' class, and the non-traversable regions are referred to as the '*negative*' class in this work. In query data, only positive data can be labeled along with their actual driving experience. The rest remains as unlabeled regions. Non-black points in Fig. 2a indicate the positive regions and the black points indicate the unlabeled regions.

However, there exists a limitation in that the query data is devoid of any supervision about negative regions. With query data only, results would be unreliable, as negative regions can be regressed as a good traversable region due to the epistemic uncertainty (Fig. 1b) Consequently, our task aims to learn binary segmentation along with traversability regression to mask out the negative regions, thereby mitigating the epistemic uncertainty. Accordingly, we utilize a very small number of hand-labeled point cloud scenes and call it '*support*' data. In support data, traversable and non-traversable regions are manually annotated as positive and negative, respectively. Manually labeling entire scenes can be biased with human intuitions. Therefore, only evident regions are labeled and used for training. Fig. 2b shows the example of labeled support data.

The overall schema of our task is illustrated in Fig. 3 and the symbols used for the description of our method are presented in Table 1. When the input point cloud data is given, a segmentation mask is applied to the initial version of the traversability regression map, producing a masked traversability map as a final output. For training, we form an episode composed of queries and randomly sampled support data. We can optimize our network over both query and relatively small support data with the episodic strategy [20]. Also, to properly evaluate the proposed framework, we introduce a new metric that comprehensively measures the segmentation and the regression, while highlighting the nature of the traversability estimation task with the epistemic uncertainty.

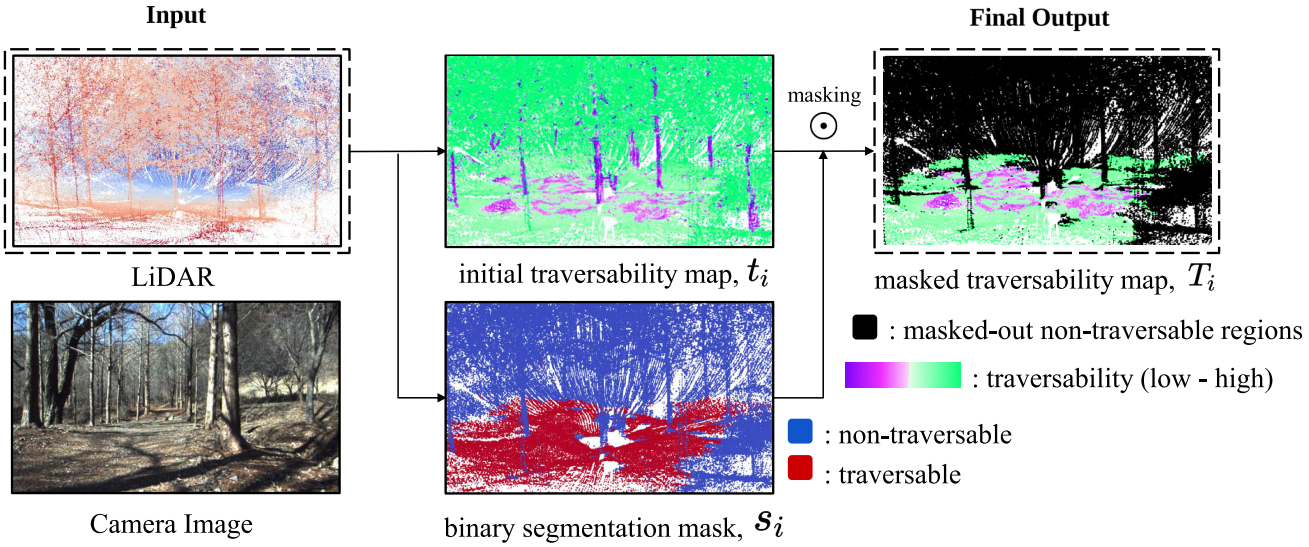


FIGURE 3. Illustration of our task definition in the inference step. Given the point cloud data, the initial traversability map and binary segmentation mask are processed. Finally, the final output is obtained by masking out the non-traversable regions of the initial traversability map.

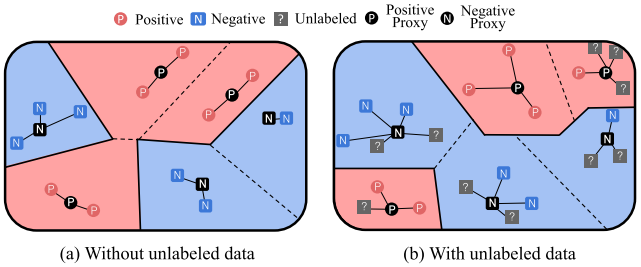


FIGURE 4. Illustration of the effect of adopting the unlabeled data. Red and blue nodes are embedding vectors of positive and negative data. Gray nodes with a question mark indicate the unlabeled data, and the black ones indicate proxies. The background color and lines indicate decision boundaries in the embedding space. The embedded vectors (non-black nodes) assigned to the proxies are connected to each other with solid lines. (a) Without unlabeled data, proxies and decision boundaries are optimized only with labeled data. (b) With unlabeled data, the optimization exploits the broader context of the training data, resulting in a more precise and discriminative decision boundary.

B. BASELINE METHOD

Let query data, consisting of positive and unlabeled data, as $\mathbb{Q} = \{\mathbb{Q}_P, \mathbb{Q}_U\}$, and support data, consisting of positive and negative data, as $\mathbb{S} = \{\mathbb{S}_P, \mathbb{S}_N\}$. Let $P_i \in \mathbb{R}^3$ denotes the 3D point, $a_i \in \mathbb{R}$ denotes the measured traversability, and $y_i \in \{0, 1\}$ denotes the class of each point. Accordingly, data from $\mathbb{Q}_P$, $\mathbb{Q}_U$, $\mathbb{S}_P$, and $\mathbb{S}_N$ are in forms of $\{P_i, a_i, y_i\}$, $\{P_i\}$, $\{P_i, y_i\}$, and $\{P_i, y_i\}$, respectively. Let f_θ denote a feature encoding backbone where θ indicates a network parameter, $x_i \in \mathbb{R}^d$ as encoded features extracted from P_i, and h_θ as the multi-layer perceptron (MLP) head for the traversability regression. g_θ denotes the MLP head for the segmentation that distinguishes the traversable and non-traversable regions.

A baseline solution learns the network with labeled data only. $\mathbb{Q}_P$ is used for the traversability regression and $\mathbb{Q}_P$ and $\mathbb{S}$ are both used for the segmentation. We obtain the traversability map $t_i = h(x_i)$, $t_i \in \mathbb{R}$, and segmentation map $s_i = g(x_i)$, $s_i \in \{0, 1\}$. The final masked traversability map T_i is represented as element-wise multiplication, $T_i = t_i \odot s_i$. The regression loss L^{reg} is computed with $\mathbb{Q}_p$ and based on a mean squared error loss as Eq. (1), where x_i is the i-th feature of point in $\mathbb{Q}_P$.

$$L^{reg}(x_i) = (h(x_i) - a_i)^2. \tag{1}$$

For the segmentation loss L^{seg}, binary cross-entropy loss is used in the supervised setting as Eq. (2), where x_i refers to the i-th element of either $\mathbb{Q}_P$ and $\mathbb{S}$. Both the positive query and the support data can be used for the segmentation loss as follows:

$$L^{seg}(x_i) = -\Big(y_i \log(g(x_i)) + (1 - y_i)\log(1 - g(x_i))\Big). \tag{2}$$

Combining the regression and the segmentation, the traversability estimation loss in the supervised setting is defined as follows:

$$L^{Baseline}(\mathbb{Q}_P, \mathbb{S}) = \frac{1}{|\mathbb{Q}_P|}\sum_{x_i \in \mathbb{Q}_P}\Big(L^{reg}(x_i) + L^{seg}(x_i)\Big) + \frac{1}{|\mathbb{S}|}\sum_{x_i \in \mathbb{S}} L^{seg}(x_i). \tag{3}$$

Nonetheless, it does not fully take advantage of data captured under various driving surfaces. Since the learning is limited to the small number of labeled data (Support $\mathbb{S}$) and the only positive supervision in Positive Query $\mathbb{Q}_P$, it can not capture the whole characteristics of the training data. This drawback hinders the capability of the traversability estimation trained in a supervised manner.

C. PROXY BANK GUIDANCE WITH METRIC LEARNING

To overcome the limitation of the supervised manner solution, we propose Proxy Bank Guidance (PBG) with a metric

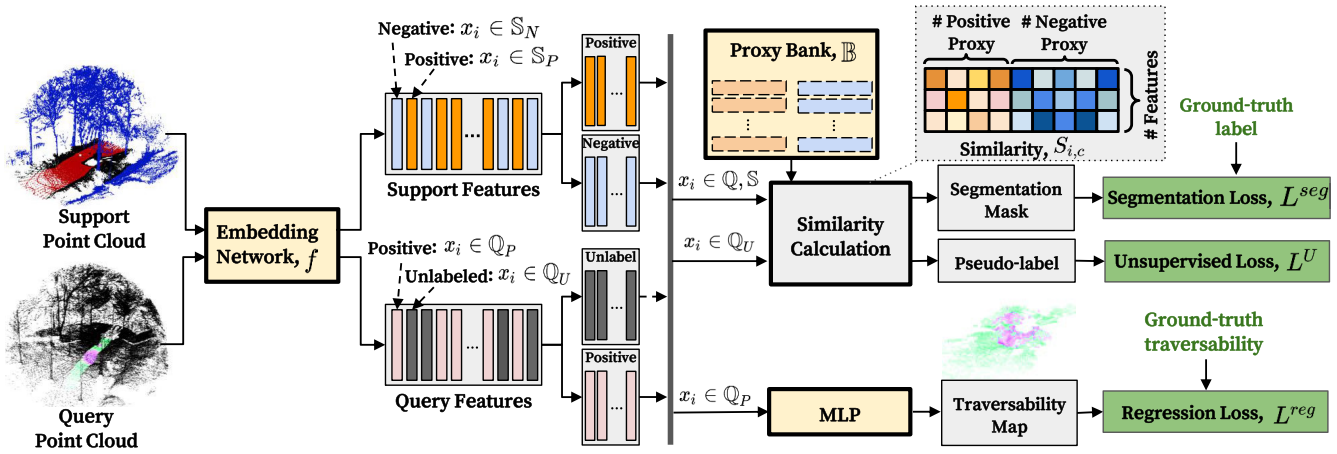


FIGURE 5. Illustration for the learning procedure of the proposed framework with deep metric learning.

learning mechanism. To this end, we adopt a proxy-based loss [42] by exploiting all available collected data including the unlabeled query data ($\mathbb{Q}_U$.) The proxy-based loss utilizes a metric learning mechanism so that we can learn embedding space and find the representation vectors that epitomize the train data in the learned embedding space. The representation vector that summarizes the data is called a proxy. The embedding network is updated based on the position of the proxies, and the proxies are adjusted by the updated embedding network, iteratively. We refer this set of proxies as '*Proxy Bank*,' denoted as $\mathbb{B} = \{\mathbb{B}_P, \mathbb{B}_N\}$, where $\mathbb{B}_P$ and $\mathbb{B}_N$ indicate the set of proxies for each class. The segmentation map is inferred based on the similarity between feature vectors and the proxies of each class, as $s_i = g(\mathbb{B}, x_i)$.

The representations of traversable and non-traversable regions exhibit large intra-class variations, where numerous sub-classes exist in each class; flat ground or gravel road for positive, and rocks, trees, or bushes for negative. For the segmentation, we use SoftTriple loss [43] that utilizes multiple proxies for each class. The similarity between x_i and class c, denoted as $S_{i,c}$, is defined by a weighted sum of cosine similarity between x_i and $\mathbb{B}_c = \{p_c^1, \ldots, p_c^K\}$, where c denotes positive or negative, K is the number of proxies per class, and p_c^k is k-th proxy in the proxy bank. The weight given to each cosine similarity is proportionate to its value. $S_{i,c}$ is defined as follows:

$$S_{i,c} = \sum_k \frac{\exp(\frac{1}{T} x_i^\top p_c^k)}{\sum_k \exp(\frac{1}{T} x_i^\top p_c^k)} x_i^\top p_c^k, \tag{4}$$

where T is a temperature parameter to control the softness of assignments. Soft assignments reduce sensitivity between multiple centers. Note that the l_2 norm has been applied to embedding vectors to sustain divergence of magnitude. Then the SoftTriple loss for the binary classification can be defined as follows:

$$L^{SoftTriple}(x_i) = -\log \frac{\exp(\lambda(S_{i,y_i} - \delta))}{\exp(\lambda(S_{i,y_i} - \delta)) + \sum_{j \neq y_i} \exp(\lambda S_{i,j})}, \tag{5}$$

where λ is a hyperparameter for smoothing effect and δ is a margin. Hence, the segmentation loss using the proxy bank is represented as Eq. (6). Here, y_i indicate a positive or negative proxy in $\mathbb{B}$. The traversability estimation loss using the proxy bank is defined as Eq. (7).

$$L^{seg}(x_i, \mathbb{B}) = -\log \frac{\exp(\lambda(S_{i,y_i} - \delta))}{\exp(\lambda(S_{i,y_i} - \delta)) + \exp(\lambda S_{i,1-y_i})}. \tag{6}$$

$$L^{Proxy}(\mathbb{Q}_P, \mathbb{S}, \mathbb{B}) = \frac{1}{|\mathbb{Q}_P|} \sum_{x_i \in \mathbb{Q}_P} \left(L^{reg}(x_i) + L^{seg}(x_i, \mathbb{B}) \right) + \frac{1}{|\mathbb{S}|} \sum_{x_i \in \mathbb{S}} L^{seg}(x_i, \mathbb{B}). \tag{7}$$

Unlabeled data, which is abundantly included in self-supervised traversability data, has not been considered in previous works. To enhance the supervision we can extract from the data, we utilize the unlabeled data in the query data in the learning process. The problem is that the segmentation loss cannot be applied to the $\mathbb{Q}_U$ because no class labels y_i exist for them. We assign an auxiliary target for each unlabeled data as clustering [44]. Pseudo class of i-th sample $\hat{y}_i$ is assigned based on the class of the nearest proxy in the embedding space as $\hat{y}_i = \text{argmax}_{c \in \{P,N\}} S_{i,c}$.

The unsupervised loss for the segmentation, denoted as L^U, is defined as Eq. (8) using the pseudo-class, where x_i is an embedding of i-th sample in $\mathbb{Q}_U$.

$$L^U(x_i, \mathbb{B}) = -\log \frac{\exp(\lambda(S_{i,\hat{y}_i} - \delta)}{\exp(\lambda(S_{i,\hat{y}_i} - \delta)) + \exp(\lambda S_{i,1-\hat{y}_i}))} \tag{8}$$

Fig. 4 illustrates the benefit of incorporating unlabeled loss. The embedding network can learn to capture the more

TABLE 1. Nomenclature.

Symbol	Description
Data	
P_i, a_i, y_i	i^{th} 3D point (x, y, z), the traversability value $(0-1)$, and binary class index (traversable:1 or non-traversable:0) corresponding to the i^{th} point, respectively.
$\mathbb{Q}_P$	Positive query data. A positive query point contains $\{P_i, a_i, y_i\}$.
$\mathbb{Q}_U$	Unlabeled query data. A negative query point only contains 3D position information $\{P_i\}$.
$\mathbb{S}_P$	Positive support data. A positive support point contains 3D position and class (traversable) corresponding to this point $\{P_i, y_i\}$.
$\mathbb{S}_N$	Negative support data. A negative support point contains 3D position and class (non-traversable) corresponding to this point $\{P_i, y_i\}$.
$\mathbb{B}$	Proxy bank. It contains the positive proxies $\mathbb{B}_P$ and negative proxies $\mathbb{B}_N$.
Deep Neural Network	
f_θ	The feature encoding network. It takes the point cloud as input. θ indicate its learnable network parameters.
h_θ	The regressor MLP head for estimating traversability value by feeding the encoded feature x_i.
g_θ	The binary classifier MLP head for distinguishing traversable and non-traversable regions by feeding the encoded feature x_i.
x_i	The encoded feature that is generated by feeding P_i to the network f.
t_i	The traversability regression value that is generated by feeding x_i to the network h.
s_i	The binary value that is generated by feeding x_i to the network g (1: traversable point, 0: non-traversable point.)
T_i	The final traversability value. It is computed by the multiplication of t_i and s_i. In other words, It indicates the traversability regression value that is classified as the traversable class.
S_{i,y_i}	The feature similarity between a point x_i and the reference class y_i.
Loss functions for Baseline	
L^{reg}	The regression loss (Eq. (1).) It uses $\mathbb{Q}_P$.
L^{seg}	The segmentation loss (Eq. (2).) It uses $\mathbb{Q}_P$, $\mathbb{S}_P$, and $\mathbb{S}_N$.
$L^{Baseline}$	The baseline loss (Eq. (3).) It is combined by L^{reg} and L^{seg}.
Loss functions for Proxy Bank Guidance (PBG)	
L^{seg} with $\mathbb{B}$	The regression loss with the Proxy Bank. It uses $\mathbb{Q}_P$, $\mathbb{S}_P$, $\mathbb{S}_N$, and $\mathbb{B}$.
L^{Proxy}	The traversability estimation loss with the Proxy Bank. It uses $\mathbb{Q}_P$, $\mathbb{S}_P$, $\mathbb{S}_N$, and $\mathbb{B}$.
L^{U}	The unsupervised loss for the segmentation. It uses $\mathbb{Q}_U$ and $\mathbb{B}$.
$L^{Traverse}$	Our final traversability loss. It uses $\mathbb{Q}_P$, $\mathbb{Q}_U$, $\mathbb{S}_P$, $\mathbb{S}_N$, and $\mathbb{B}$.

broad distribution of data, and learned proxies would represent training data better. When unlabeled data features are assigned to the proxies (Fig. 4a,) the embedding space and proxies are updated as Fig. 4b, exhibiting more precise decision boundaries.

Combining the aforementioned objectives altogether, we define our final objective as '*Traverse Loss*,' and is defined as Eq. (9). The overall high-level schema of the learning procedure is depicted in Fig. 5.

$$L^{Traverse}(\mathbb{Q}, \mathbb{S}, \mathbb{B}) = L^{Proxy}(\mathbb{Q}_P, \mathbb{S}, \mathbb{B}) + \frac{1}{|\mathbb{Q}_U|} \sum_{x_i \in \mathbb{Q}_U} L^U(x_i, \mathbb{B}) \tag{9}$$

D. RE-INITIALIZATION TO AVOID TRIVIAL SOLUTIONS

Our PBG method can suffer from sub-optimal solutions, which are induced by empty proxies. Empty proxies indicate the proxies to which none of the data are assigned. Such empty proxies should be redeployed to be a good representation of training data. Otherwise, the model might lose the discriminative power and the bank might include semantically poor representations.

Our intuitive idea to circumvent an empty proxy is to re-initialize the empty proxy with support data features. By updating the empty proxies with support data, the proxy bank can reflect training data that was not effectively captured beforehand. In order to obtain representative feature vectors without noises, M number of prototype feature vectors, denoted as $\mu^+ = \{\mu_m^+, m = 1, \ldots, M\}$ and $\mu^- = \{\mu_m^-, m = 1, \ldots, M\}$, are estimated using an Expectation-Maximization algorithm [45]. We followed the implementation of the EM algorithm from [33] that uses the vector distance function. The EM algorithm consists of iterative E-steps and M-steps. In each E-step, the expectation of each feature to the prototypes is computed, which can be regarded as cluster assignment. Then, for each M-step, the prototype vectors are updated as weighted averages of features. The prototype vectors are cluster centers of support features. We randomly choose the prototype vectors with small perturbations and use them as re-initialized proxies. Algorithm 1 summarizes the overall training procedure of our method, and Fig. 9 shows the three different states of proxies; initial, trivial, and optimal states.

E. TRAVERSABILITY PRECISION ERROR

We devise a new metric for the proposed framework, '*Traversability Precision Error*' (TPE). The new metric should be able to comprehensively evaluate the segmentation and the regression while taking the critical aspect of the traversability estimation into account. One of the most important aspects of traversability estimation is to avoid the false-positive of the traversable region, the region that is impossible to traverse but inferred as traversable.

Self-supervised traversability estimation produces overconfident predictions when confronted with out-of-distribution samples, which can result in catastrophic failure while exploring unfamiliar terrain. If such a region is estimated as traversable, a robot will likely go over that region, resulting in undesirable movements. In other words, the primary goal of the task is to eliminate false positive estimations of non-navigable locations in order to ensure safe navigation. The impact of the false-positive decreases if they are estimated as less traversable. The metrics should take this aspect into account the characteristic that non-traversable regions predicted as highly traversable are much more unfavorable than non-traversable regions estimated as less traversable.

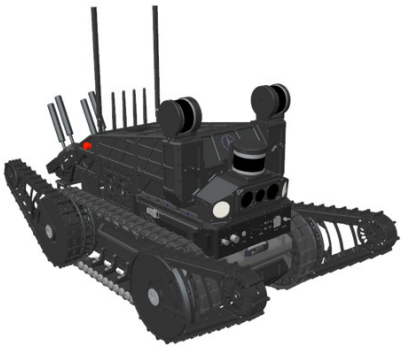
Therefore, TPE computes the degree of false-positive of the traversable region, extenuating its impact with the

Algorithm 1 Single Epoch of Traversability Estimation With Metric learning

Input: Query data $\mathbb{Q} = \{\mathbb{Q}_P, \mathbb{Q}_U\}$ and Support Data $\mathbb{S} = \{\mathbb{S}_P, \mathbb{S}_N\}$, where $|\mathbb{Q}| \gg |\mathbb{S}|$
Output: Network f with parameters θ, proxy bank $\mathbb{B} = \{\mathbb{B}_P, \mathbb{B}_N\}$
for *each query data* **do**
  Random Sample support data from $\mathbb{S}$
  Feed query and support data to f_θ, and **Get** embedding features x_i
    Calculate similarity between x_i and $\mathbb{B}$
    Estimate Pseudo-class $\hat{y}_i$ for $x_i \in \mathbb{Q}_U$
  Calculate $L^{Traverse}$
  Update θ and $\mathbb{B}$
end
Calculate the membership of each proxy
if *an empty proxy exists* **then**
  Feed $\mathbb{S}$ to f_θ, and **Get** embedding features
  Estimate M cluster centers for each class, $\mu = \{\mu^+, \mu^-\}$, by the EM algorithm
  Re-initialize empty proxy to μ with small random perturbation
end

traversability t_i. The TPE is defined as Eq. (10) where TN, FP, and FN denote the number of true negative, false positive, and false negative points of the traversable region, respectively.

$$\text{Traversability Precision Error (TPE)} = \frac{TN}{TN + FP(1 - t_i) + FN} \tag{10}$$



(a)

(b)

FIGURE 6. Dtrail dataset. (a) Our mobile robotic platform with one 32-layer and two 16-layers of LiDARs. (b) Images of the mountain trail scenes where we construct the dataset.

IV. EXPERIMENTS

In this section, our method is evaluated with *Dtrail* dataset for traversability estimation on off-road environments along with SemanticKITTI [18] dataset. Our PBG method is compared to other metric learning methods based on episodic training strategies. Furthermore, we conduct various ablation studies to show the benefits of our method. For better clarity, we first define the ablation options as Table 3.

A. DATASETS

1) DTRAIL: OFF-ROAD TERRAIN DATASET

In order to thoroughly examine the validity of our method, we build the Dtrail dataset, a real mobile robot driving dataset of high-resolution LiDAR point clouds from mountain trail scenes. We collect point clouds using one 32 layer and two 16 layers of LiDAR sensors equipped on our caterpillar-type mobile robot platform, shown in Fig. 6a. Our dataset consists of 119 point cloud scenes and each point cloud scene has approximately 4 million points. Corresponding sample camera images of point cloud scenes are shown in Fig. 6b. For the experiments, we split 98 scenes for the query set and 4 scenes for the support set, and 17 scenes for the evaluation set. For the traversability, the magnitude z-acceleration from the Inertial Measurement Unit (IMU) of the mobile robot is re-scaled from 0 to 1 and mapped to points that the robot actually explored. Also, in terms of data augmentation, a small perturbation is added along the z-axis on some positive points.

2) SemanticKITTI

We evaluate our method on the SemanticKITTI [18] dataset, which is an urban outdoor-scene dataset for point cloud segmentation. Since it does not provide any type of attributes for traversability, we conducted experiments on segmentation only. It contains 11 sequences, 00 to 10 as the training set, with 23, 210 point clouds and 28 classes. We split 5 sequences (00, 02, 05, 08, 09) with 17, 625 point clouds for training and the rest, with 5, 576 point clouds, for evaluation. We define the 'road', 'parking', 'sidewalk', 'other-ground', and 'terrain' classes as positive and the rest classes as negative. For query data, only the 'road' class is labeled as positive, and left other

TABLE 2. Comparison results on Dtrail and SemanticKITTI dataset. Our methods with different objectives are annotated as follows. **PBG (Baseline)**: Eq. (3) that is trained in supervised manner. **PBG (w.o. Unsupervised)**: Eq. (7) that does not leverage unlabeled data. **PBG (w.o. Re-init)**: Eq. (9) excluding the re-initialization step. **PBG (All)**: Eq. (9).

	Dtrail						SemanticKITTI			
	mIoU			TPE			mIoU			
$\lvert\mathbb{S}\rvert/\lvert\mathbb{Q}\rvert$	4%	2%	1%	4%	2%	1%	5%	1%	0.5%	0.1%
ProtoNet [29]	0.8033	0.7515	0.5049	0.7129	0.5624	0.3249	0.8009	0.8040	0.7993	0.7798
MPTI [20]	0.6992	0.6936	0.6390	0.6202	0.5466	0.4995	0.8586	0.8108	0.7531	0.7663
PBG (Baseline)	0.9238	0.8857	0.7779	0.8896	0.8447	0.7345	0.8405	0.8376	0.8338	0.8201
PBG (w.o. Unsupervised)	0.8864	0.8529	0.8461	0.8434	0.8121	0.8164	0.8124	0.7896	0.8049	0.7994
PBG (w.o. Re-init)	0.8970	0.8771	0.7935	0.8649	0.8163	0.7517	0.8058	0.7895	0.8058	0.7895
PBG (All)	**0.9338**	**0.9151**	**0.9005**	**0.9067**	**0.8776**	**0.8636**	**0.8652**	**0.8402**	**0.8473**	**0.8973**

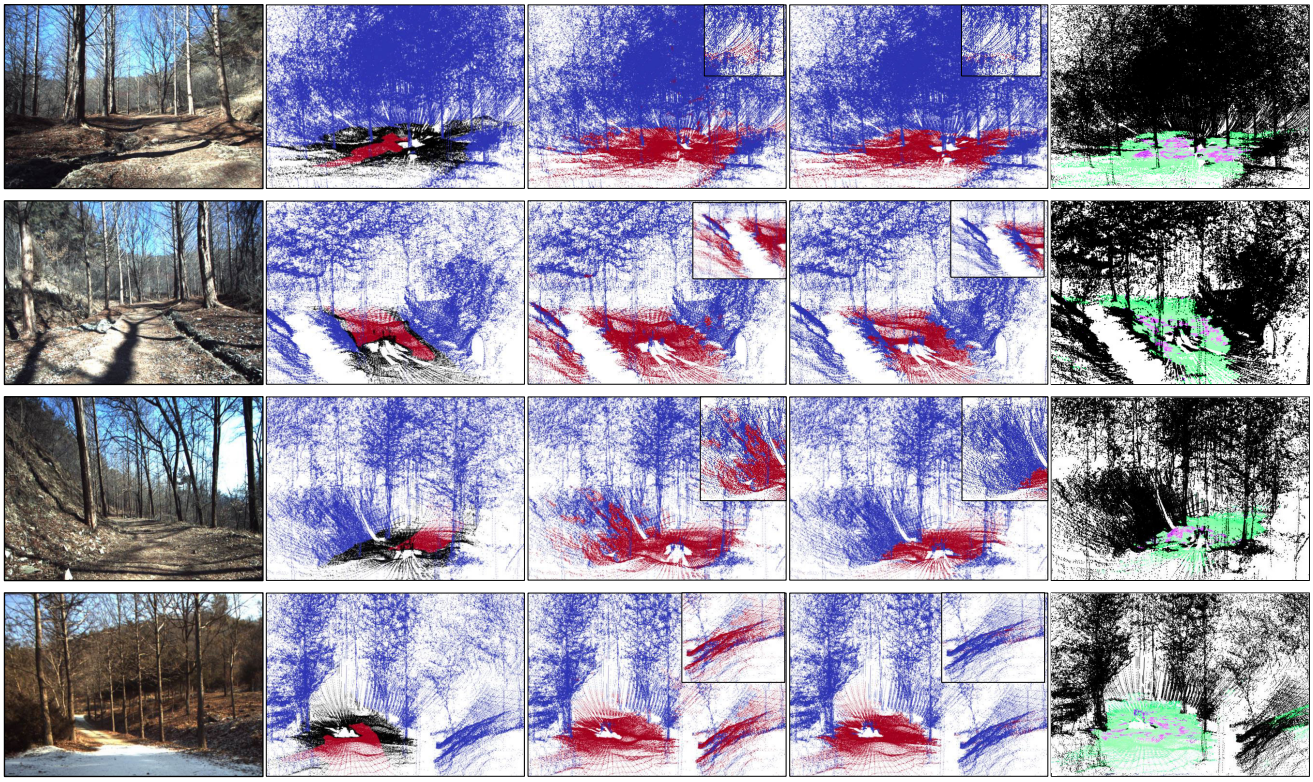


FIGURE 7. Qualitative results for Dtrail dataset. (a) Camera image of each scene. (b)-(d) Support data and inference results of segmentation. A **red** point indicates a traversable region, a **blue** one indicates a non-traversable region, and a **black** point is an unlabeled region. (e) The final output of our traversability estimation. The traversability of non-traversable regions is masked out using the segmentation result.

positive classes as unlabeled. We expect the model to learn the other positive regions using unlabeled data without direct supervision.

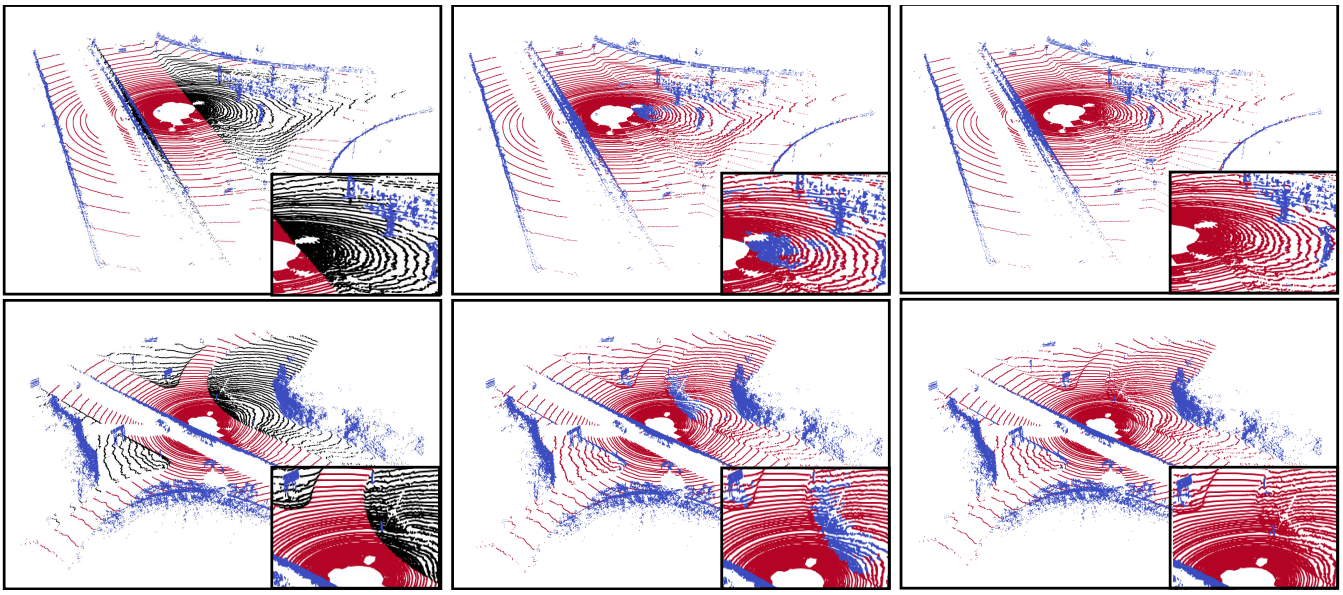
B. EVALUATION METRIC

We evaluate the performance of our method with TPE, the new criteria designed for the traversability estimation task, which evaluates segmentation and regression quality simultaneously. Additionally, we evaluate the segmentation quality with mean Interaction over Union [46] (mIoU). For each class, the IoU is calculated by $IoU = \frac{TP}{TP+FP+FN}$, where TP, FP, and FN denote the number of true negatives, false positives, and false negative points of each class, respectively.

C. IMPLEMENTATION DETAILS

1) EMBEDDING NETWORK

RandLA-Net [4] is fixed as a backbone embedding network for every method for a fair comparison. Specifically, we use 2 down-sampling layers in the backbone and excluded global (x, y, z) positions in the local spatial encoding layer, which



(a) Ground Truth (b) PBG (Baseline) (c) PBG (All)

FIGURE 8. Qualitative results for SemanticKITTI. A red-colored point indicates a traversable region, a blue-colored point indicates a non-traversable region, and a black point is an unlabeled region.

TABLE 3. Applied components for the ablation studies. (All other components not listed here are used).

Components for PBG	$L^{Baseline}$	L^{U}	Re-init
Baseline	✓		
w.o. Unsupervised	✓		✓
w.o. Re-init	✓	✓	
All	✓	✓	✓

aids the network to embed local geometric patterns explicitly. The embedding vectors are normalized with l_2 norm and are handled with cosine similarity.

2) TRAINING

We train the model and proxies with Adam optimizer with the exponential learning rate decay for 50 epochs. The initial learning rate is set as $1e^{-4}$. For query and support data, K-nearest neighbors (KNN) of a randomly picked point is sampled in training steps. We ensure that positive and negative points exist evenly in sampled points of the support data.

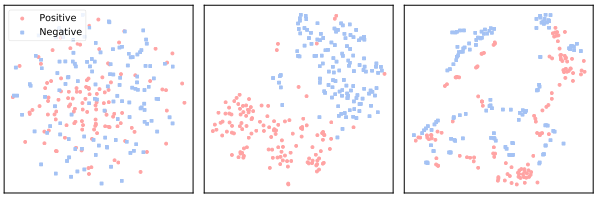
3) HYPERPARAMETER SETTING

For learning stability, proxies are updated exclusively for the initial 5 epochs. The number of proxies K is set to 128 for each class and the proxies are initialized with normal distribution. We set small margin δ as 0.01, λ as 20, and temperature parameter T as 0.05 for handling multiple proxies.

D. RESULTS

1) COMPARISON

We compare the performance to ProtoNet [29] which uses a single prototype and MPTI [20] which adopts multiple prototypes for few-shot 3D segmentation. Also, we compare the performance with our supervised manner method, denoted as 'PBG (Baseline).' Table 2 summarizes the result of experiments. Our method shows a significant margin in terms of IoU and TPE compared to the ProtoNet and MPTI. It demonstrates that generating prototypes in a non-parametric approach does not represent the whole data effectively. Moreover, it is notable that we show the performance of our proxy bank guidance with metric learning method is better than the supervised setting designed for our task. It verifies that ours can reduce epistemic uncertainty by incorporating unlabeled data by unsupervised loss. For SemanticKITTI, the observation is similar to that of the Dtrail dataset. Even though the SemanticKITTI is based on urban scenes, our method shows better performance than other few-shot learning methods by 6% and the supervised manner (PBG (Baseline)) by 2%.



(a) Initial state (b) Trivial state (c) Optimal state

FIGURE 9. t-SNE visualization of the embedding space according to the distribution of data. Red and blue colors correspond to positive and negative proxies, respectively.

2) ABLATION STUDIES

We repeat experiments with varying support-to-query ratio ($|\mathbb{S}|/|\mathbb{Q}|$) to evaluate robustness regarding the amount of support data. Table 2 shows that our metric learning method is much more robust from performance degradation than the

TABLE 4. Ablation study on Dtrail according to the number of proxies K.

K	1	2	4	8	16	32	64	128	256	512
mIoU(↑)	0.890	0.894	0.880	0.906	0.911	0.883	**0.920**	**0.934**	**0.931**	**0.924**
TPE(↑)	0.847	0.868	0.840	0.862	0.881	0.845	**0.888**	**0.906**	**0.898**	**0.895**

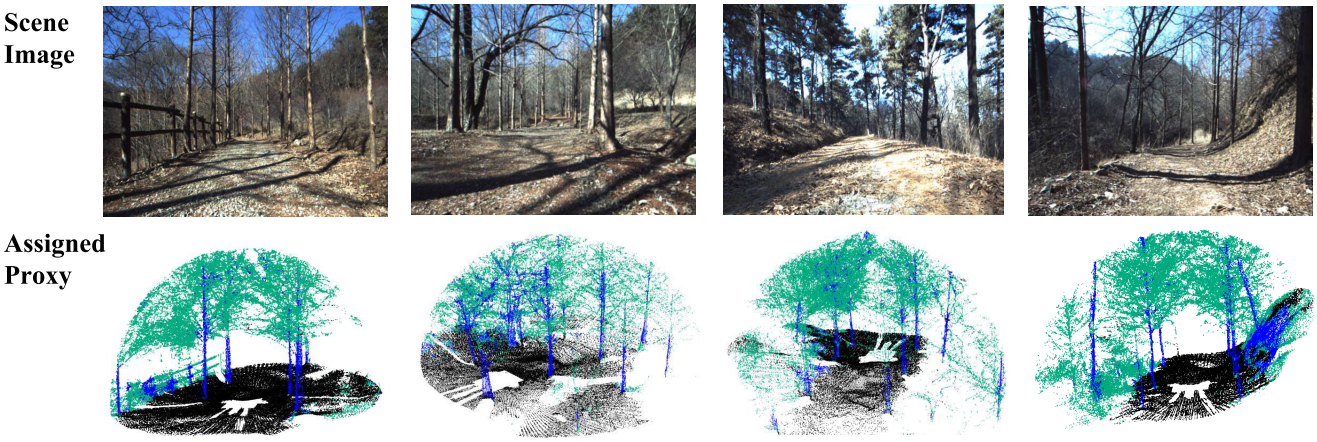


FIGURE 10. Proxy visualization of the scene in our Dtrail dataset. The color of each point represents the proxy assigned to the point. It shows that the learned proxies are well-clustered, and mapped with the semantic features on the point cloud scene. The viewpoint of the camera and LiDAR image slightly differs, according to the different sensor locations.

others when the support-to-query ratio decreases. When the ratio decreases from 4% to 1% in the Dtrail dataset, the TPE of our metric learning method only decreases about 4% while the TPE of others dropped significantly: 39% for ProtoNet, 13% for MPTI, and 16% for PBG (Baseline). It verifies that our method can robustly reduce epistemic uncertainty with small labeled data. This experiment firmly shows that PBG (Baseline) shows stable performance compared to other methods, on the strength of the unlabeled data. Note that all PBG (Baseline) can utilize the small number of support data ($\mathbb{S}$) and query data ($\mathbb{Q}$) that includes only a large amount of positive data, while other methods (ProtoNet [29] and MPTI [20]) do not.

Moreover, we observe that performance increases by 6% on average on TPE when adopting the re-initialization step. It confirms the re-initialization step can help avoid trivial solutions. Also, it is shown that adopting the unsupervised loss can boost the performance up to 6% on average. It verifies that the unlabeled loss can give affluent supervision without explicit labels. Table 3 shows the conditions for conducting ablation studies on our PBG. For the fair experiments on the effects of Unsupervised loss and Re-initialization, all methods not listed in the table are basically applied. Moreover, as shown in Table 4, an increasing number of proxies boost the performance until it converges when the number exceeds 32, demonstrating the advantages of multiple proxies.

3) QUALITATIVE RESULTS

Fig. 7 shows the traversability estimation results of our supervised-based (PBG (Baseline)) and metric learning-based method (PBG (All)) on the Dtrail dataset. We can examine that our metric learning-based method performs better than the supervised-based method. Especially, our method yields better results on regions that are not labeled on training data. We compare the example of segmentation results with the SemanticKITTI dataset in Fig. 8. The first column indicates the ground truth and the other columns indicate the segmentation results of the supervised learning-based method and our method. Evidently, our method shows better results on unlabeled regions, which confirms that our metric learning-based method reduces epistemic uncertainty.

Fig. 9 shows the t-SNE visualization of the proxy banks. The proxies are initialized with random distribution in the initial state, as shown in Fig. 9a. After learning without proxy re-initialization, empty proxies, in which no data are assigned, occur. It leads to a state where proxies have simplistic distributions, as shown in Fig. 9b. It loses the ability to represent diverse representations to distinguish positive and negative features. On the other hand, the proxies learned with re-initialization algorithms result in an optimal state of proxies, in which proxies are precisely positioned to discriminate between traversable and non-traversable regions.

Fig. 10 shows the visualization of the proxies assigned to the point cloud scenes. For better visualization, proxies are clustered into three representations. We observe that the learned proxies successfully represent the various semantic features. Leaves, grounds, and tree trunks are mostly colored green, black, and blue, respectively.

V. CONCLUSION

We propose a self-supervised traversability estimation framework on 3D point cloud data in terms of mitigating epistemic uncertainty. Self-supervised traversability estimation suffers

from the uncertainty that arises from the limited supervision given from the data. We tackle the epistemic uncertainty by concurrently learning semantic segmentation along with traversability estimation, eventually masking out the non-traversable regions. We start from the fully-supervised setting and finally developed the deep metric learning method with unsupervised loss that harnessed the unlabeled data. To properly evaluate our method, we also devise a new evaluation metric according to the task's settings and underline the important criteria of the traversability estimation. We build our own off-road terrain dataset with the mobile robotics platform in unconstrained environments for realistic testing. Various experimental results show that our method is promising.

Future work. Our proposed method presents a method for learning a deep learning network that can predict traversability by properly utilizing unlabeled data, and showed that it can have excellent performance with the help of a very small amount of support data. However, our algorithm still requires support data based on human-annotated positive and negative labels. In future work, we plan to conduct research that can overcome these limitations by utilizing a zero-shot learning method or domain adaptation method.

ACKNOWLEDGMENT

(Jihwan Bae and Junwon Seo contributed equally to this work.)

REFERENCES

[1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.
[2] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "NuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11618–11628.
[3] P. Sun et al., "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2443–2451.
[4] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11105–11114.
[5] Z. Chen, J. Zhang, and D. Tao, "Progressive LiDAR adaptation for road detection," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 3, pp. 693–702, May 2019.
[6] J. Sock, J. Kim, J. Min, and K. Kwak, "Probabilistic traversability map generation using 3D-LiDAR and camera," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 5631–5637.
[7] J. Ahtiainen, T. Stoyanov, and J. Saarinen, "Normal distributions transform traversability maps: LiDAR-only approach for traversability mapping in outdoor environments," *J. Field Robot.*, vol. 34, no. 3, pp. 600–621, May 2017.
[8] S. Matsuzaki, J. Miura, and H. Masuzawa, "Semantic-aware plant traversability estimation in plant-rich environments for agricultural mobile robots," 2021, *arXiv:2108.00759*.
[9] T. Guan, Z. He, D. Manocha, and L. Zhang, "TTM: Terrain traversability mapping for autonomous excavator navigation in unstructured environments," 2021, *arXiv:2109.06250*.
[10] H. Roncancio, M. Becker, A. Broggi, and S. Cattani, "Traversability analysis using terrain mapping and online-trained terrain type classifier," in *Proc. IEEE Intell. Vehicles Symp. Proc.*, Jun. 2014, pp. 1239–1244.
[11] 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. Autom. Lett.*, vol. 4, no. 2, pp. 1509–1516, Apr. 2019.
[12] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. R. Bradski, "Self-supervised monocular road detection in desert terrain," in *Robotics: Science and Systems*, vol. 38. Philadelphia, PA, USA, 2006.
[13] 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. (IROS)*, Oct. 2022, pp. 7197–7204.
[14] C. A. Brooks and K. D. Iagnemma, "Self-supervised classification for planetary rover terrain sensing," in *Proc. IEEE Aerosp. Conf.*, Oct. 2007, pp. 1–9.
[15] 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. (IROS)*, Oct. 2016, pp. 1184–1189.
[16] H. Kolvenbach, C. Bärtschi, L. Wellhausen, R. Grandia, and M. Hutter, "Haptic inspection of planetary soils with legged robots," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1626–1632, Apr. 2019.
[17] W. V. Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. V. Gool, "Scan: Learning to classify images without labels," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 1–12.
[18] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9296–9306.
[19] A. Valada, J. Vertens, A. Dhall, and W. Burgard, "AdapNet: Adaptive semantic segmentation in adverse environmental conditions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 4644–4651.
[20] N. Zhao, T. Chua, and G. H. Lee, "Few-shot 3D point cloud semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8869–8878.
[21] P. Dallaire, K. Walas, P. Giguère, and B. Chaib-draa, "Learning terrain types with the Pitman–Yor process mixtures of Gaussians for a legged robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 3457–3463.
[22] L. Ding, H. Gao, Z. Deng, J. Song, Y. Liu, G. Liu, and K. Iagnemma, "Foot–terrain interaction mechanics for legged robots: Modeling and experimental validation," *Int. J. Robot. Res.*, vol. 32, no. 13, pp. 1585–1606, Nov. 2013.
[23] W. Bosworth, J. Whitney, S. Kim, and N. Hogan, "Robot locomotion on hard and soft ground: Measuring stability and ground properties in-situ," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 3582–3589.
[24] G. G. Waibel, T. Löw, M. Nass, D. Howard, T. Bandyopadhyay, and P. V. K. Borges, "How rough is the path? Terrain traversability estimation for local and global path planning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16462–16473, Sep. 2022, doi: 10.1109/TITS.2022.3150328.
[25] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML Deep Learn. Workshop*, vol. 2, 2015, p. 10.
[26] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," 2017, *arXiv:1711.04043*.
[27] O. Vinyals, C. Blundell, T. P. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. NIPS*, 2016, pp. 1–22.
[28] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.
[29] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," 2017, *arXiv:1703.05175*.
[30] C. Chen, O. Li, C. Tao, A. Jade Barnett, J. Su, and C. Rudin, "This looks like that: Deep learning for interpretable image recognition," 2018, *arXiv:1806.10574*.
[31] J. Deuschel, D. Firmbach, C. I. Geppert, M. Eckstein, A. Hartmann, V. Bruns, P. Kuritcyn, J. Dexl, D. Hartmann, D. Perrin, T. Wittenberg, and M. Benz, "Multi-prototype few-shot learning in histopathology," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 620–628.
[32] K. R. Allen, E. Shelhamer, H. Shin, and J. B. Tenenbaum, "Infinite mixture prototypes for few-shot learning," 2019, *arXiv:1902.04552*.
[33] B. Yang, C. Liu, B. Li, J. Jiao, and Q. Ye, "Prototype mixture models for few-shot semantic segmentation," 2020, *arXiv:2008.03898*.
[34] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[35] M. Wang and W. Deng, "Deep face recognition: A survey," *Neurocomputing*, vol. 429, pp. 215–244, Mar. 2021.
[36] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jul. 2005, pp. 539–546.
[37] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 1735–1742.
[38] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1386–1393.
[39] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4004–4012.
[40] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 360–368.
[41] E. Wern Teh, T. DeVries, and G. W. Taylor, "ProxyNCA++: Revisiting and revitalizing proxy neighborhood component analysis," 2020, *arXiv:2004.01113*.
[42] S. Kim, D. Kim, M. Cho, and S. Kwak, "Proxy anchor loss for deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3235–3244.
[43] Q. Qian, L. Shang, B. Sun, J. Hu, T. Tacoma, H. Li, and R. Jin, "SoftTriple loss: Deep metric learning without triplet sampling," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6449–6457.
[44] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 1–6.
[45] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, Nov. 1997.
[46] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

JIHWAN BAE received the B.S. degree in electrical engineering and computer science (EECS) from the Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea, in 2020. Since 2020, he has been a Researcher with the Ground Technology Research Institute, Agency for Defense Development, Daejeon, South Korea. His research interests include theoretical machine learning and deep learning, high-level computer vision, and studies of perception for autonomous vehicles. His awards and honors include the Caltech Summer Undergraduate Research Fellow (SURF) Award, in 2019 and the Best Paper Award from the GIST EECS Department, in 2019.

JUNWON SEO received the B.S. degree from the Department of Computer Science Engineering, College of Business Administration, Seoul National University, South Korea, in 2021. He is currently a Research Officer with Agency for Defense Development(ADD). His research interests include robot vision and 3D perceptions for autonomous systems and deep learning.

TAEKYUNG KIM was born in Daejeon, South Korea, in 1997. He received the B.S. degree from the College of Transdisciplinary Studies, Daegu Gyeongbuk Institute of Science and Technology (DGIST), in 2020. Since 2020, he has been a Researcher with the Ground Technology Research Institute, Agency for Defense Development, Daejeon. His research interests include the development of mobile robot navigation systems, robot planning using deep learning and studies of path planning, control, perception, and integration for autonomous vehicles. His awards and honors include the Talent Award of Korea (Deputy Prime Minister of the Republic of Korea); the First prize from the SOSCON 2019 Robot Open Source Lab (Competition at Samsung Open Source Conference); the First prize from the SOSCON 2018 Robot Cleaner Autonomous Path Planning Algorithm Hackathon (Competition at the Samsung Open Source Conference); and the First prizes on the Autonomous Vehicle Technical Report, International Student Green Car Competitions, in 2018 and 2019.

HAE-GON JEON received the B.S. degree from the School of Electrical and Electronic Engineering, Yonsei University, in 2011, and the M.S. and Ph.D. degrees from the School of Electrical Engineering, KAIST, in 2013 and 2018, respectively. He was a Postdoctoral Researcher with the Robotics Institute, Carnegie Mellon University. He is currently affiliated with both the AI Graduate School and the School of Electrical Engineering and Computer Science, GIST, as an Associate Professor. His research interests include computational imaging, 3D reconstruction, and AI for social good. He was the winner of the Best Ph.D. Thesis Award from KAIST, in 2018.

KIHO KWAK, (Member, IEEE) received the B.S. and M.S. degrees from Korea University, in 1999 and 2001, respectively, and the Ph.D. degree in ECE from Carnegie Mellon University (CMU), in 2012. He is a Principal Researcher in Agency for Defense Development, South Korea. His research interests include sensor fusion, online object modeling and perception and navigation for autonomous vehicles in outdoor environment.

INWOOK SHIM (Member, IEEE) received the B.S. degree in computer science from Hanyang University, in 2009, the M.S. degree in robotics program, E.E. from KAIST, in 2011, where he received the Ph.D. degree from the Division of Future Vehicles, E.E., in 2017. From 2017 to 2022, he was a Research Scientist with the Agency for Defense Development, South Korea. He is currently an Assistant Professor with the Department of Smart Mobility Engineering, Inha University, Incheon, South Korea. His research interests include a 3D vision for autonomous systems and deep learning. He was a member of "Team KAIST", which took first place at the DARPA Robotics Challenge Finals, in 2015. He received the KAIST Achievement Award of Robotics and the Creativity and Challenge Award from KAIST.