Open-Set Semi-Supervised Learning for 3D Point Cloud Understanding

Xian Shi  
South China University of Technology, China

Xun Xu*  
Institute for Infocomm Research

Wanyue Zhang  
Max Planck Institute for Informatics

Xiatian Zhu  
University of Surrey

Chuan Sheng Foo  
Institute for Infocomm Research

Kui Jia  
South China University of Technology, China

Abstract—Semantic understanding of 3D point cloud relies on learning models with massively annotated data, which, in many cases, are expensive or difficult to collect. This has led to an emerging research interest in semi-supervised learning (SSL) for 3D point cloud. It is commonly assumed in SSL that the unlabeled data are drawn from the same distribution as that of the labeled ones; this assumption, however, rarely holds true in realistic environments. Blindly using out-of-distribution (OOD) unlabeled data could harm SSL performance. In this work, we propose to selectively utilize unlabeled data through sample weighting, so that only conducive unlabeled data would be prioritized. To estimate the weights, we adopt a bi-level optimization framework which iteratively optimizes a meta-objective on a held-out validation set and a task-objective on a training set. Faced with the instability of efficient bi-level optimization, we further propose three regularization techniques to enhance the training stability. Extensive experiments on 3D point cloud classification and segmentation tasks verify the effectiveness of our proposed method. We also demonstrate the feasibility of a more efficient training strategy. Our code is released on Github.

1. INTRODUCTION

Understanding 3D worlds is a fundamental and challenging problem in computer vision with a variety of critical applications for robotics, autonomous driving, etc. 3D point cloud is one of the most efficient ways to represent 3D worlds, as they can be collected from many sensors and algorithms, e.g. LiDAR and SLAM. The state-of-the-art approaches towards 3D point cloud understanding require access to a large amount of labeled data to train MLPs [20], graph convolution networks [29], or transformers [35]. Acquiring the annotations for 3D point cloud data is nevertheless expensive due to extensive interaction with 2D projections in multiple viewpoints. Alternative to developing deeper and more complicated backbone networks, recent works explore unlabeled data to improve the performance of 3D point cloud models when labeled data is scarce [33], [36], [8], [32], a.k.a. semi-supervised learning (SSL). Among these works, consistency-based SSL is proven to be particularly effective [33], [36]. In general, it adopts a self-training paradigm by maintaining two networks and use the pseudo-labels predicted on unlabeled data to supervise a trainable network.

SSL under such a setting is commonly referred to as open-set semi-supervised learning for 3D point cloud. These SSL methods often assume that unlabeled data are all relevant and conducive to the learning task. However this is often not true as unlabeled data may be collected from a different domain from the labeled data likely with different data distributions and class spaces. Moreover, unlabeled data may not be carefully curated and are heavily contaminated by noise. For instance, regions cropped out from large-scale real scans could contain both foreground object and background, along with noise introduced by reconstruction algorithms. Therefore, an increasingly important problem in semi-supervised 3D point cloud learning is to learn with potentially non-conducive unlabelled data, a.k.a. out-of-distribution data (OOD) [5], [9].

SSL can be classified into two categories. The first genre approaches from some heuristic criterion [5], [34], [25], e.g. the consistency of predictions upon multiple augmentations or training an outlier detector. A heuristic threshold is typically required to prune out outlier/out-of-distribution samples from the unlabeled data pool. Another line of research focuses on developing a principled objective to optimize the per-sample weights for unlabeled data [9], [23], dubbed as learning based approaches. These approaches formulate open-set SSL as a bi-level optimization problem. The weights on unlabeled data [23] or a parameterized weight predictor network [9] are regarded as hyperparameters. By iteratively optimizing the meta-objective, a loss defined on a separate labeled dataset, and task-objective,
a loss defined on classification or segmentation, one is able to
train a model with uneven weights applied to unlabeled data. A lower weight often indicates an OOD unlabeled sample.

Despite offering a more principled framework, there are several shortcomings with existing learning based approaches. Firstly, in the bi-level optimization formulation of [9] the meta-objective and the task objective share the same labeled training data. Consequently, a trivial solution exists with this formulation, i.e. all unlabeled data’s weights equal to 0, and the meta-gradient will be constantly 0. An alternative bi-level approach [23] treats per-sample weights as the hyper-parameters to optimize and the meta-objective is defined on a separate validation set, thus avoiding the trivial solution. However, as the weights for all unlabeled data must be updated in every training iteration, this approach restricts itself to conducting bi-level optimization on the full dataset. When more unlabeled data becomes available, the expensive bi-level optimization procedure must be carried out again. Finally, in a gradient-based bi-level optimization, the updating of hyperparameters depends on the meta-gradient. This gradient is usually computed by approximation algorithms [23], [15], [16] and is subject to approximation noise. Blindly using the hypergradients with off-the-shelf gradient optimizer, e.g. SGD or Adam [12], will lead to training instability.

To overcome the above disadvantages of existing bi-level optimization based approaches, we propose an open-set SSL framework with REgularized Bi-level Optimization (ReBO). Unlike existing alternatives, ReBO defines the meta-objective on a held-out validation set, thus getting rid of trivial solution. ReBO further employs an MLP as the weight predictor to obtain per-sample weights on unlabeled data. Importantly, this weight predictor can be pretrained on a small training subset and then transferred to the full dataset without change, resulting in more efficient model training. Besides, ReBO is complemented with three meta-gradient regularization techniques, including outlier detection, parameter moving averaging and entropy regularization. Extensive results validate the advantage of introduced regularizations.

We summarize the contributions of this work as below.
- We formulate open-set semi-supervised learning as a bi-level optimization problem. To avoid trivial solution, meta-objective is defined on a held-out validation.
- A weight predictor network is introduced to estimate per-sample weights for unlabeled data. Critically, it can be trained on a subset of dataset efficiently before applied to full-scale open-set SSL without the need for further optimization.
- To address the instability issue of bi-level optimization, we introduce three regularization terms to further stabilize meta optimization loop.

II. RELATED WORK

3D Point Cloud Deep Learning. 3D Point cloud represents 3D object surface structure via a collection of 3D points. Common tasks on 3D Point cloud learning include classification [4], [30], segmentation [18]. Since the emergence of deep learning based 3D point cloud understanding, PointNet [20] and PointNet++ [21], there has been a surge in the development of deep learning backbones, e.g. DGCNN [29], PointCNN [14], CurveNet [31], and transformer [35]. However, recent success is built upon the access to large amount of labeled 3D point cloud data. Annotating 3D data is hard and how to alleviate the dependence on labeled data is becoming increasingly important.

Semi-Supervised Learning. Semi-supervised learning aims to exploit large amount of unlabeled data to improve learning performance. Consistency-based SSL has demonstrated superior performance partially due to explicitly exploiting pseudo labels on unlabeled data [22], [27], [26]. The state-of-the-art SSL method [26] generates pseudo-labels for weakly augmented samples with confident predictions, and then match them with the prediction of the strongly augmented ones. Inspired by the success of SSL, adaptation to 3D point cloud learning was introduced by [33], [36], [6] with successful demonstration on semantic segmentation and object detection. The SSL approaches treats all unlabeled data equally while some unlabeled data could be harmful to the target task. As a result, open-set SSL emerges as a way to selectively exploit unlabeled data.

Open-Set Semi-Supervised Learning. Open-set semi-supervised learning [5], [34], [9], [19], [25] refers to the challenging case where unlabeled data may contain harmful data, sometimes referred to as out-of-distribution data (OOD). Open-set SSL can be roughly classified into two genres. The heuristic OOD detection approaches often explicitly detect unlabeled OOD samples, through training an OOD detector [34], [25] or calculating consistency of pseudo-label predictions [5], [17], [19]. Some heuristic thresholding is required to distinguish outliers from inliers. Alternative to the heuristic approaches, a learning-based paradigm defines a meta-objective, often as the loss on a separate labeled dataset [9], [23], where open-set SSL is formulated as learning per-sample weight on unlabeled data such that the model trained with weighted loss minimizes the meta-objective. The whole problem can be formulated as a bi-level optimization problem. In this work, we address the unmet challenges in existing learning-based approaches and consistently improved on open-set semi-supervised 3D point cloud classification and segmentation tasks. This work is also related to meta-learning a loss function [3] in that the weighting parameters of the training loss are learned through meta-learning.

III. METHODOLOGY

In this section, we first briefly review semi-supervised learning and then propose an open-set semi-supervised learning framework by weighting unlabeled loss. Then we introduce the bi-level optimization procedure with newly proposed regularization terms.

A. Semi-Supervised Learning

We first provide an overview of semi-supervised learning. Specifically, we denote the labeled training dataset as $D^l_{tr} = \{X_i, y_i\}_{i=1}^{N_{l_{tr}}}$, the unlabeled training dataset as
\( D_{tr} = \{ X_j \}_{j=1}^{N_{tr}} \) and the validation dataset as \( D_{val} = \{ X_k, Y_k \}_{k=1}^{N_{val}} \). Consistency based SSL often optimizes a loss function consisting of two terms, one defined on labeled data \( L_t \) and another on unlabeled data \( L_u \). For classification and segmentation tasks, labeled loss is often instantiated as a loss function determining a per-sample weight applied to unlabeled samples by assuming that harmful true under an open-set SSL scenario where OOD unlabeled data are helpful and consistency loss is optimized. This results in the Moving Averaging regularization.

\[
L_{reg} = \| w_t - w_{t-1} \|^2 = \| t(X_j; \theta) - w_t \|^2 (7)
\]

For any differential validation loss, e.g. cross-entropy loss, the first term \( \frac{\partial L_{val}(\theta^*)}{\partial \theta^*} \) is computed analytically. Calculating the second term involves differentiation over a trajectory of \( \theta \) if gradient descent style updating is adopted for task optimization. For efficient optimization, we adopt the one-step approximation proposed in [15] \( \Theta^* = \Theta - \alpha \nabla_{\Theta^*} L_{tr} \), where \( \alpha \) is set equal to the learning rate for task optimization. The meta-gradient in Eq. (3) now writes \( \nabla_{\theta} L_{val} = -\alpha \nabla_{\Theta^*} L_{tr}((\theta), \Phi)\nabla_{\Theta^*} L_{val}(\Theta^*, \Phi) \). The Hessian-vector multiplication can be efficiently approximated by finite difference approximation as below where \( \Theta^{\pm} = \Theta \pm \epsilon \nabla_{\Theta^*} L_{val}(\Theta^*, \Phi) \) and \( \epsilon = 1e^{-2}/\| \nabla_{\Theta^*} L_{val}(\Theta^*, \Phi) \|^2_2 \).

\[
\nabla_{\theta} L_{val} \approx -\alpha \nabla_{\Theta^*} L_{tr}(\Theta^+, \Phi) - \nabla_{\Theta^*} L_{tr}(\Theta^-, \Phi) (4)
\]

C. Regularizing Meta-Gradient

Learning weight predictor by minimizing validation loss is challenging as the meta-gradient, Eq. (3) is often noisy due to the multiple approximations adopted. To improve the training stability we propose to add further regularizations to help optimize meta-objective.

Regularizing by Outlier Detection. We first propose to incorporate a regularization inspired by outlier detection. As with the heuristic open-set SSL approaches [34], discovering out-of-distribution is often treated in a similar fashion to outlier detection [24]. Under this assumption, the majority of training samples are regarded as normal samples and thus the encoder network is forced to embed features in a cluster, a.k.a. one-class classification. Inspired by this design, given a clean (in-distribution) validation set, we propose to add a classification branch such that all validation set should have a uniform weight. More specifically, the following cross-entropy loss is adopted.

\[
L_{OD} = \sum_{X_k \in D_{val}} - \log \lambda(X_k; \Phi) (5)
\]

We further notice the meta-gradient computed from approximation algorithms may not always point to the correct direction that reduces the meta-objective. As a result, optimizing hyperparameters with meta-gradients alone is empirically unstable and sometimes may not converge at all. Therefore, we propose a regularization to smooth out the noisy meta-gradients.

Disappearing Tikhonov Regularization. We wish to regularize the weight predictor’s output to be smooth over iterations. Therefore, we first notice that the Disappearing Tikhonov regularization (DTR) as below helps smooth parameters temporally, where \( w_{t-1} \) is the weight predicted in the previous iteration.

\[
L_{reg} = ||\lambda(X_j; \Phi) - w_{t-1}||_2^2 (6)
\]

Moving Averaging Regularization. Despite DTR regularizes weights temporally, it only considers the weight one-step back. To enforce stronger temporally smoothness, we propose to maintain a moving average of unlabeled sample weights \( \tilde{w}_t = \beta \tilde{w}_{t-1} + (1 - \beta) w_t \) as target for regularization. This results in the Moving Averaging regularization.

\[
L_{reg} = ||w_t - \tilde{w}_{t-1}||_2^2 = ||\lambda(X_j; \Phi) - \tilde{w}_{t-1}||_2^2 (7)
\]
Entropy Regularization: Since the per-sample weights are continuous, it is not guaranteed the harmful samples will receive weights that are low enough to eliminate the negative impact. Therefore, the weight predictor is further forced to produce output close to binary distribution. We add an entropy regularization term for this purpose, defined as below.

\[
L_{\text{ent}} = - \frac{1}{|D_{\text{tl}}|} \sum_{X_j \in D_{\text{tl}}} \lambda(X_j; \Phi) \log \lambda(X_j; \Phi) \\
+ (1 - \lambda(X_j; \Phi)) \log \lambda(X_j; \Phi)
\] (8)

Eventually, we combine all regularization terms and formulate the final meta optimization objective as,

\[
\min_{\Phi} \mathcal{L}_{\text{val}} + \gamma \mathcal{L}_{\text{reg}} + \xi \mathcal{L}_{\text{ent}} + \eta \mathcal{L}_{\text{OD}}
\] (9)

All regularizers have analytic gradient w.r.t. \( \Phi \) and can be linearly combined with approximated validation gradient as the final meta-gradient.

D. Training Strategy

Bi-level optimization is expensive when unlabeled data is huge. To avoid the expensive training, we propose two efficient warm-up training strategies for open-set SSL. In specific, we opt for a less computational demanding paradigm by pre-training the backbone and predictor network on labeled data and a subset of unlabeled data. After convergence, we infer and fix the weights for all unlabeled data, and later transfer them to retrain a new backbone from scratch, termed as Transfer approach. It’s worth noting that only backbone is updated during retraining, thus it saves substantial time for bi-level optimizing weight predictor. Alternatively, we take the pre-trained parameters to initialize a backbone and a predictor, and then fine-tune on full data for just a few epochs, termed as Fine-tune approach. Although both warm-up methods can significantly save training time, they bring different enlightenment. Weights transferring implies that the predicted weights in our methods could be transferred to other backbones, as a fixed measure of data fitness. While fine-tuning suggests a general bi-level training paradigm for large-scale open-set SSL, i.e. dividing a subset data for pilot training.

IV. EXPERIMENT

A. Dataset

We evaluate our methods on three datasets including ModelNet40[30], ShapeNet Part[18] and ObjectNN[28]. ModelNet40 is a shape classification benchmark dataset consisting of 12,311 CAD shape models from 40 categories, split into 9,843 for training and 2,468 for testing. For each shape, we uniformly sample 1024 points on its CAD mesh and normalize them into a unit sphere. ShapeNet Part contains 16,881 shapes from 16 categories with 50 part categories, split into 12,137 for training, 1,870 for validating and 2,874 for testing. For each shape model, 2048 points are uniformly sampled from surface mesh. ObjectNN is a real scan object dataset, where 2,902 objects in 15 categories are scanned from real scene ScanNet[7] and SceneNN[11] and each object is uniformly sampled with 2048 points.

Data Split: Throughout the training we keep the amount of all labeled data to be fixed. At every new epoch, the validation set is randomly drawn from the whole labeled dataset and the rest are used as labeled training data with a 1:1 ratio. As a result, all labeled data could be used for updating task network throughout the whole training episode.

B. Configuration

Encoder Network. We choose DGCNN [29] as the default backbone for point cloud classification and part segmentation. In addition to DGCNN, we further evaluate with CurveNet [31] for transfer experiments.

Weight Predictor. Our weight predictor is composed of a three-layer MLP with batch normalization and relu activation following each linear layer. The output layer is normalized to between 0 and 1 by a sigmoid function. For classification network, weight predictor takes the backbone’s global feature (after maxpooling) as input. In part segmentation task, we append maxpooled feature to point-wise feature as input to weight predictor.

Warm-up. Randomly-initialized weight predictor is very unstable, resulting in great fluctuations of predicted weights in the beginning. Inspired by Multi-OS [34], we pre-train backbone and weight predictor for 30 epochs to obtain good initial weights. Specifically, we pre-train backbone with labeled loss only, and at the meantime, we guide weight predictor to output zero-weights for all unlabeled samples while one-weights for validation samples by a cross entropy loss.

Hyper-Parameters. During backbone training, we adopt cross entropy loss as labeled loss and FixMatch [26] consistency loss as unlabeled loss. In addition, loss combination in Eq. 9 is used to optimize weight predictor. Here we set the rate \( \beta \) acting in Eq. 7 to 0.5, and the constraint hyper-parameters \( \gamma, \eta \) in Eq. 9 are set to 0.1, 0.01 empirically. As a strong entropy regularization may prevent weight predictor from optimizing properly at the beginning of training, we gradually increase \( \xi \) over the training episode. We defer the details of \( \xi \) updating to the supplementary.

Competing Methods. Our baseline is conducted with DGCNN backbone trained by FixMatch semi-supervised loss with all unlabeled weights fixed to 1. We compare our method with 4 most relevant and up-to-date methods, DS3L[9], Multi-OS[34], LTWA[23] and OP-Match [25]. We migrate them to our point cloud setting and combine them with the same FixMatch SSL. Because LTWA needs to calculate meta-objective on a separate validation set, we adopt the same data split strategy to dynamically divide training and validating set in each epoch.

C. SSL with Out-of-Distribution Data

We create an open-set ssl setting by incorporating different OOD data into the unlabeled dataset. Unlike [17] and [34] which mix the unlabeled target data with an extra OOD dataset as the whole unlabeled set, we consider that there could be more than one type of OOD data present in unlabeled set. It means some OOD samples may be extremely harmful for...
backbone learning while others may be less harmful or even useful. Therefore, we define a weak and a strong OOD sets. We choose S3DIS[2] to create OOD set for classification and part segmentation experiments. S3DIS is a large indoor scene point cloud dataset, covering 271 rooms from 6 areas. We follow PointNet [20] to divide each room into 1*1 meter blocks and randomly sample 1024/2048 (classification/part segmentation) points from each blocks. 10,000 randomly selected blocks are treated as weak OOD data (W) because some blocks may just crop out objects, like chair, table, etc., that appears in the target dataset. In addition, we select another 10,000 samples and augment them by random rotating with a maximum amplitude of 90° and Gaussian jittering $N(0, 1)$, as strong OOD set (S). This augmentation will destroy even the manifold of good objects, thus be deemed strong OOD.

1) Open-set SSL Classification: We first evaluate on ModelNet40 shape classification benchmark. For training, we randomly sampled {100, 400, 1000} samples as labeled set $L$ and the rest as in-distribution unlabeled data $U$. In addition, we append S3DIS $W$ and $S$ as OOD unlabeled set. We report overall classification accuracy with different methods in Tab. I. We first show four baseline results under different combinations of data in the top section of the table. It can be seen that conducting semi-supervised learning with in-distribution unlabeled data can significantly improve backbone accuracy (from 55.1% to 62.1% @ 100 labeled). However, with weak OOD samples in unlabeled pool we observe a clear drop of performance (from 62.1% to 60.2% @ 100 labeled). When strong OOD samples are included {L, U, W, S}, a more significant loss of performance is observed (from 60.2% to 58.8% @ 100 labeled).

We then evaluate all competing methods and our proposed ReBO method on the full dataset {L, U, W, S}. We observe all open-set SSL methods improves over the baseline thanks to the ability to detect and downweight OOD samples. In particular, ReBO achieves the highest classification results under all three amounts of labeled data, even higher than the baseline with completely clean data (L + U). This suggests that ReBO is able to discover some useful examples from the weak $W$ and strong $S$ OOD samples and improve performance with these data as additional unlabeled data.

2) Open-set SSL Segmentation: We conduct part-segmentation experiments on ShapeNet with {20, 100, 400} random labeled samples $L$ and the other as in-distribution unlabeled data $U$. We also add S3DIS $W$ and $S$ in all unlabeled data. The mean IoU over all samples are showed in Tab. II. First, adding additional weak OOD dataset does not necessarily harm the baseline performance. We hypothesize the reason being segmentation focuses more on local parts and geometry and the weak OOD data are cropped out from real scanned data thus carrying more diverse local geometry that benefits the segmentation task. Moreover, compared with other methods, our method again performs the best in all three unlabeled data settings. At the higher labeling regimes (100, 400 labeled) ReBO again beats baselines without strong OOD samples.

---

### TABLE I

**ACCURACY (%) FOR DIFFERENT METHODS WITH DIFFERENT NUMBERS OF LABELED SAMPLES ON MODELNET40 CLASSIFICATION.**

| Data       | Methods | 100 labeled | 400 labeled | 1000 labeled |
|------------|---------|-------------|-------------|--------------|
| L          | Baseline | 62.1        | 80.5        | 84.7         |
| (L, U)     | Baseline | 60.2        | 79.4        | 84.0         |
| (L, U, W)  | Baseline | 65.8        | 79.9        | 83.7         |
| (L, U, W, S)| DS3L [19] | 59.6        | 79.3        | 84.4         |
|            | Multi-OS [34] | 59.2 | 79.1 | 83.6         |
|            | LTWA [23] | 60.4 | 79.5 | 84.9         |
|            | OP-Match [25] | 61.1 | 80.0 | 84.6         |
| ReBO       | 80.4     | 80.9        | 82.9        |

### TABLE II

**mIoU(%) FOR DIFFERENT METHODS WITH DIFFERENT NUMBERS OF LABELED SAMPLES ON SHAPENET PART SEGMENTATION.**

| Data       | Methods | 20 labeled | 100 labeled | 400 labeled |
|------------|---------|------------|-------------|-------------|
| L          | Baseline | 94.0       | 74.6        | 78.1        |
| (L, U)     | Baseline | 55.5       | 73.9        | 78.7        |
| (L, U, W)  | Baseline | 56.3       | 74.6        | 78.9        |
|            | DS3L [19] | 55.4       | 73.6        | 76.4        |
| (L, U, W, S)| Multi-OS [34] | 56.0 | 74.4 | 78.1         |
|            | LTWA [23] | 54.3       | 74.1        | 78.7        |
|            | OP-Match [25] | 55.8 | 74.7 | 78.4         |
| ReBO       | 58.4     | 78.6        | 82.9        |

### TABLE III

**CLASSIFICATION ACCURACY (%) ON REAL-WORLD OBJECTNN EXPERIMENTS.**

| Data       | Methods | 5% | 20% |
|------------|---------|----|-----|
| L          | Baseline | 37.5 | 66.1 |
| (L, U)     | Baseline | 40.4 | 67.5 |
| (L, U, W)  | Baseline | 38.2 | 66.4 |
|            | DS3L [19] | 38.6 | 65.0 |
| (L, U, W, S)| Multi-OS [34] | 39.3 | 65.1 |
|            | LTWA [23] | 39.0 | 67.7 |
|            | OP-Match [25] | 39.5 | 67.1 |
| ReBO       | 40.6     | 68.5        |

D. Open-set SSL on Real Scanned Data

Compared with synthetic data, real scanned data is more meaningful and challenging. We now further validate the performance of ReBO on real scanned ObjectNN dataset [28]. We assume that objects scanned from real scenes are not perfectly cropped and are subject to background noise. To simulate this scenario, we perturb the object bounding boxes provided in ScanNet and SceneNN, and then crop out OOD point cloud samples from perturbed boxes. In the experiments, we collect 2,000 noisy OOD samples denoted as O to complement the labeled data L and unlabeled data U. The final results in Tab. III demonstrate that open-set methods generalize to practical point cloud situation, and our method still performs the best among all competing methods.

E. Additional Evaluation

Comparing with SOTA SSL In this section, we compare with existing label-efficient learning methods on ShapeNet part segmentation benchmark. Similarly, we divide ShapeNet into L and U. Moreover, we could have ModelNet40 as additional unlabeled data M. For the competing methods, ScanNet $\mathcal{N}$ and ShapeNetCore $\mathcal{C}$ were respectively used as unlabeled data. In Tab. IV, we compare our method with SO-Net [13], PointCapsNet[37], JointSSL[1], Multi-task [10], PCont [32], ACD [8] under {1%, 5%} labeled samples(L). We demonstrate that with the help of unlabeled data U and additional ModelNet40 M, we can achieve much better results than the competing methods.
TABLE IV
PART SEGMENTATION ON SHAPE NET WITH LIMITED LABELED TRAINING DATA. N, C AND M INDICATE SCAN NET, SHAPE NET CORE AND MODEL NET 40, RESPECTIVELY.

| Data | Methods | 1% | 5% |
|------|---------|----|----|
| \{C\} | PointCapNet[31] | 67.0 | 70.0 |
| \{C\} | Xun[SSL][1] | 71.9 | 77.4 |
| \{C\} | Multi-task[10] | 68.2 | 77.7 |

TABLE V
ABLATION STUDY. DTR INDICATES DISAPPEARING TIKHONOV REGULARIZATION. MATR INDICATES MOVING AVERAGING TIKHONOV REGULARIZATION.

| Predictor | Entropy | OD | TmpReg | Transfer | Fine-tune | Acc(%) |
|-----------|---------|----|--------|----------|----------|--------|
| ✓         | ✓       | ✓  | ✓      | ✓        | ✓        | 59.0   |
| ✓         | ✓       | ✓  | ✓      | ✓        | ✓        | 60.4   |
| ✓         | ✓       | ✓  | ✓      | ✓        | ✓        | 61.0   |
| ✓         | ✓       | ✓  | ✓      | ✓        | ✓        | 61.7   |
| ✓         | ✓       | ✓  | ✓      | ✓        | ✓        | 62.1   |
| ✓         | ✓       | ✓  | ✓      | ✓        | ✓        | 62.7   |
| ✓         | ✓       | ✓  | ✓      | ✓        | ✓        | 63.4   |

F. Ablation Study

We conduct ablation experiments on 100 labeled samples \( \mathcal{L} \) and the rest \( \mathcal{U} \) from ModelNet40 classification with \( \mathcal{W} \) and \( \mathcal{S} \) from S3DIS.

Regularization We first validate the importance of weight predictor, entropy loss and outlier detection (OD). As is shown in Tab. V, baseline (58.8%) is the result of backbone directly trained on \( \{ \mathcal{L}, \mathcal{U}, \mathcal{W}, \mathcal{S} \} \). Though adding a pure weight predictor improves baseline by only 0.2%, the later joined Entropy in Eq. 8 and OD in Eq. 5 enhance classification performance by 1.4% and 0.6%, respectively. In addition, we compare Disappearing Tikhonov Regularization(DTR) and Moving Averaging Tikhonov Regularization(MATR) in Sect. III-C and conclude that MATR gives better performance with a larger 1.1% improvement.

Transfer and Fine-tune Estimating per-sample weight with bi-level optimization is expensive, we evaluate the two approaches, Transfer and Fine-Tune, introduced in Sect. III-D. In specific, we first train both backbone and predictor on labeled data and a subset of unlabeled data (10%), and then respectively evaluate Transfer and Fine-Tune strategies. As we observe from Tab. V, Transfer and Fine-tune(for 50 epochs) could improve the accuracy to 62.7% and 63.4%, respectively. It proves that pre-training is feasible for training bi-level optimization and it saves substantial computation cost by training on only 10% unlabeled data.

G. Analysis on Predicted Weights

Distribution of Predicted Weights In Fig. 2, we visualize the distribution histograms of final predicted weights by different methods. In detail, we divide ten intervals evenly between 0 and 1, and plot the weights as histograms according to its proportion. For example, in the top-left figure, the orange column of \( \mathcal{U} \) weights occupies nearly 5%, which means that almost 5% of \( \mathcal{U} \) weights are in the interval (0.1, 0.2]. Moreover, we present the plot of average weight, where the first point means that the average of \( \mathcal{U} \) weights is 0.479. As shown in the top-left figure, weights produced by DS3L are concentrated around 0.5, and \( \mathcal{U}, \mathcal{W} \) and \( \mathcal{S} \) have similar average values. Multi-OS and OP-Match who construct outlier detection would set all predicted outlier weights as 0 and the rest inlier as 1, thus in the top-right OP-Match weights figure, all weights are predicted as 0 or 1. Although the number of samples with a weight of 1 in \( \mathcal{U} \) is significantly more than that in \( \mathcal{W} \) and \( \mathcal{S} \), almost all weak and strong OOD samples and substantial amount of in-distribution unlabeled samples are excluded, resulting in a fewer training samples. Prediction by LTWA is shown in the bottom-left plot that it cannot distinguish \( \mathcal{U}, \mathcal{W} \) and \( \mathcal{S} \) well. In our method, as shown in the bottom-right figure, weights are concentrated at either near 0 or 1. There are more weights near 1 in \( \mathcal{U} \), while more weights are close to 0 in \( \mathcal{W} \) and \( \mathcal{S} \). Therefore, the average of \( \mathcal{U} \) is significantly higher than \( \mathcal{W} \) and \( \mathcal{S} \).

V. Conclusion

In this work, we are motivated by the negative impact of out-of-distribution unlabeled data and propose an open-set semi-supervised learning problem for 3D point cloud understanding. In light of the drawbacks of existing open-set SSL approaches, we propose a bi-level optimization based approach. The meta-objective is defined on a held-out validation set to avoid trivial solution. To further regularize the noisy meta-gradients, we propose three regularization terms, namely outlier detection, entropy regularization and moving averaging weights. Extensive experiments on 3D point cloud classification and segmentation validate the efficacy of proposed methods.

VI. Acknowledgement

This work was in part supported by the Guangdong R&D key project of China (No. 2019B010155001), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (No. 2017ZT07X183), and the National Natural Science Foundation of China (NSFC) under Grant 62106078. Xun Xu acknowledges A*STAR Career Development Award (Grant no. C210112059) for supporting this work.

Fig. 2. Distribution histograms and average lines of predicted weight by different methods. The histogram percentage scale y-axis is on the left, while the average scale is on the right.
REFERENCES

[1] Antonio Alliegro, Davide Boscaini, and Tatiana Tommasi. Joint supervised and self-supervised learning for 3D real world challenges. In 2020 25th International Conference on Pattern Recognition (ICPR), pages 6718–6725. IEEE, 2021. 5, 6

[2] I Armeni, O Sener, Ar Zamir, H Jiang, I Brilakis, M Fischer, S Savarese, and I. Armeni. A method for stochastic optimization. In International Conference on Learning Representations, 2018, 2, 3, 4, 5

[3] Sarah Bechtlé, Artem Molchanov, Yevgen Chebotar, Edward Grefenstette, Ludovic Rigettti, Gaurav Sukhatme, and Franziska Meier. Meta learning via learned loss. In International Conference on Pattern Recognition, IEEE, 2021. 2

[4] Angel X Chen, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3D model repository. arXiv preprint arXiv:1512.03012, 2015. 2

[5] Yanbei Chen, Xiaotian Zhu, Wei Li, and Shaogang Gong. Semi-supervised learning under class distribution mismatch. In Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 1, 2

[6] Mingmei Cheng, Le Hui, Jin Xie, and Jian Yang. Sophe-net: Semi-supervised semantic 3D point cloud segmentation network. In Proceedings of the AAAI Conference on Artificial Intelligence, pages 1140–1147, 2021. 2

[7] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießlten. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5828–5837, 2017. 4

[8] Mathew Gadelia, Aruni RoyChowdhury, Gopal Sharma, Evangelos Kalogerakis, Liangliang Cao, Erik Learned-Miller, Rui Wang, and Subhransu Maji. Label-efficient learning on point clouds using approximate convex decompositions. In European Conference on Computer Vision, pages 473–491, 2020. 1, 5, 6

[9] Jian-Zhe Guo, Zhi-Yu Yang, Yuan Jiang, Yu-Feng Li, and Zhi-Hua Zhou. Safe deep semi-supervised learning for unseen-class unlabeled data. In International Conference on Machine Learning, 2020. 1, 2, 4, 5

[10] K. Hassan and M. Haley. Unsupervised multi-task feature learning on point clouds. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 5, 6

[11] Binh-Son Hua, Quang-Hieu Pham, Duy Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. In 2016 Fourth International Conference on 3D Vision (3DV), pages 92–101. IEEE, 2016. 4

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In International Conference on Learning Representations, 2015.

[13] J. Li, B. M. Chen, and G. H. Lee. So-net: Self-organizing network for point cloud analysis. IEEE, 2018. 5, 6

[14] Yangyin Li, Ru Bu, Mengchao Sun, Wei Wu, Xinhan Di, and Baosuan Chen. Pointcnn: Convolution on x-transformed points. Advances in neural information processing systems, 31, 2018. 2

[15] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In International Conference on Learning Representations, 2018.

[16] Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In International Conference on Artificial Intelligence and Statistics, 2020.

[17] Hong-Ying Luo, Hao Cheng, Xiyue Guo, Ke Li, Mengdan Zhang, Fanxu Meng, Xiaoai Wei, Feiyue Huang, and Xing Sun. The consistency training for open-set semi-supervised learning. arXiv preprint arXiv:2101.08237, 2021. 2, 4

[18] Kaichun Mo, Shelin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical level-3D object understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 909–918, 2019. 2, 4

[19] Jongmin Park, Sukmin Yun, Jongheon Jeong, and Jinwoo Shin. Opencos: Contrastive semi-supervised learning for handling open-set unlabeled data. arXiv preprint arXiv:2107.08943, 2021. 2

[20] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017. 1, 2, 5

[21] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Advances in Neural Information Processing Systems, 2017. 2

[22] Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. Semi-supervised learning with ladder networks. arXiv preprint arXiv:1507.02672, 2015. 2

[23] Zhongzheng Ren, Raymond Yeh, and Alexander Schwing. Not all unlabeled data are equal: Learning to weight data in semi-supervised learning. Advances in Neural Information Processing Systems, pages 21786–21797, 2020. 1, 2, 3, 4, 5

[24] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In International conference on machine learning, 2018. 3

[25] Kuniai Saito, Donghyun Kim, and Kate Saenko. Openmatch: Open-set consistency regularization for semi-supervised learning with outliers. In Advances in Neural Information Processing Systems, 2021. 1, 2, 4, 5

[26] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. arXiv preprint arXiv:2001.07685, 2020. 2, 4

[27] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. Advances in neural information processing systems, 30, 2017.

[28] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hu, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019. 4, 5

[29] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog), 2019. 1, 2, 4

[30] Zhong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoxu Tang, and Xiaoxiao Xiao. 3D shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1912–1920, 2015. 2

[31] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. arXiv preprint arXiv:2105.01288, 2021. 2, 4

[32] Qing Yu, Daiki Ikami, Go Irie, and Kiyoharu Aizawa. Multi-task curriculum framework for open-set semi-supervised learning. In European Conference on Computer Vision, 2020. 1, 2, 3, 4, 5

[33] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In European Conference on Computer Vision, 2020. 1, 2, 3, 4, 5

[34] Qing Yu, Daiki Ikami, Go Irie, and Kiyoharu Aizawa. Multi-task curriculum framework for open-set semi-supervised learning. In European Conference on Computer Vision, 2020. 1, 2, 3, 4, 5

[35] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In European Conference on Computer Vision, 2020. 1, 2, 3, 4, 5

[36] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Sess: Self-ensembling semi-supervised 3D object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13706–13715, 2020. 1, 2

[37] Ting Yang, Zhiwei Wang, and Yisheng Li. A second-order equation for semi-supervised learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1009–1018, 2019. 5, 6