You Never Get a Second Chance To Make a Good First Impression: Seeding Active Learning for 3D Semantic Segmentation

Nermin Samet\textsuperscript{1}, Oriane Siméoni\textsuperscript{2}, Gilles Puy\textsuperscript{2}, Georgy Ponimatkin\textsuperscript{1}, Renaud Marlet\textsuperscript{1,2}, Vincent Lepetit\textsuperscript{1}

\textsuperscript{1}LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-Vallée, France \textsuperscript{2}Valeo.ai, Paris, France

Abstract

We propose SeedAL, a method to seed active learning for efficient annotation of 3D point clouds for semantic segmentation. Active Learning (AL) iteratively selects relevant data fractions to annotate within a given budget, but requires a first fraction of the dataset (a ‘seed’) to be already annotated to estimate the benefit of annotating other data fractions. We first show that the choice of the seed can significantly affect the performance of many AL methods. We then propose a method for automatically constructing a seed that will ensure good performance for AL. Assuming that images of the point clouds are available, which is common, our method relies on powerful unsupervised image features to measure the diversity of the point clouds. It selects the point clouds for the seed by optimizing the diversity under an annotation budget, which can be done by solving a linear optimization problem. Our experiments demonstrate the effectiveness of our approach compared to random seeding and existing methods on both the S3DIS and SemanticKitti datasets. Code is available at https://github.com/nerminsamet/seedal.

1. Introduction

We are interested in the efficient annotation of sparse 3D point clouds (as captured indoors by depth cameras or outdoors by automotive lidars) for semantic segmentation.

Modern AI systems require training on large annotated datasets to reach a high performance on such a task. As annotating is costly (more than capturing the data itself), several approaches have been proposed to achieve a more frugal learning, such as semi-supervision [71], weak supervision [88], few-shot [52] and zero-shot learning [56], self-supervision [37, 1] and, as studied here, active learning [57].

Active learning (AL) methods iteratively select relevant fractions of a dataset to be annotated within a given budget so that, after a few iterations, the model learned on the annotated fractions reaches a performance close to the performance of the model learned on the fully-annotated dataset, although at a much lower annotation cost. This AL selection typically targets the most uncertain [38] or most diverse [62] data, that are assumed to have the most positive impact when annotated and used for training.

The criteria used in AL methods for selecting data to annotate generally require a preliminary fraction of the dataset to be already annotated. Assuming this fraction is representative enough of the rest of the dataset, it can be used to estimate the benefit of annotating other data fractions. But such a preliminary fraction is not available at the first AL iteration, when confronted to a completely unannotated dataset.

Figure 1: Impact of active learning seed on performance. We show the variability of results obtained with 20 different random seeds (blue dashed lines), within an initial annotation budget of 3% of the dataset, when using various active learning methods for 3D semantic segmentation of S3DIS. We compare it to the result obtained with our seed selection strategy (solid red line), named SeedAL, which performs better or on par with the best (lucky) random seeds among 20, and "protects" from very bad (unlucky) random seeds.
There is thus a cold start problem, which is to determine a first fraction of the dataset to annotate — the seed.

**Initial set.** Nevertheless, most AL publications pay little attention to the choice of this seed before iterating their AL method. They just pick a random fraction of the dataset. Although results are sometimes presented averaged over a few runs, it can be insufficient given the high variance level. As shown in Figure 1, it is particularly true in our 3D semantic segmentation context. Besides, nothing prevents in practice from drawing an “unlucky” seed, leading to significant underperformance or annotation overheads.

To address this issue, our method, named SeedAL, automatically selects extremely good AL seeds, and thus also protects from drawing unlucky seeds, as illustrated on Figure 1. Only few approaches have recently been proposed to construct such good seeds to get AL on track right away [84, 54, 40, 29, 48, 81, 13, 50], and they mostly regard text or image classification. To the best of our knowledge, no such AL seeding approach exists for 3D point clouds.

**Self-supervised features.** for text or images [20, 31, 12], are the key to most of these approaches. It thus seems natural to use self-supervised 3D features in our case. But existing pre-trained 3D backbones are not as versatile as their 2D counterpart, and provide lower quality features (see Sect. 3.3). Our approach to seed AL for 3D point cloud is to leverage high-quality pretrained features [12] for images that are views of the scanned scenes.

As we rely on image features like [54, 40, 29, 81, 13], it seems natural as well to reuse these AL seeding methods for 3D scenes. But a direct transposition to 3D is not sensible because, in our case, the initial budget for the seed is not given in terms of number of images but in the size of scenes (number of 3D points): selecting a scene just because it contains a single image of interest would be suboptimal.

**Our approach** integrates scenes and views to make a better use of the initial annotation budget. While 2D approaches use feature clustering to guide seed selection, we show it is suboptimal in our context (2D views of 3D scenes) and propose a better formulation based on binary linear programming. As our algorithm has a higher complexity than clustering, we also develop heuristics to scale the annotation context. Besides, nothing prevents in practice from drawing an “unlucky” seed, leading to significant underperformance or annotation overheads.

**Contributions.** To the best of our knowledge, we are the first to address AL seeding for 3D point clouds:

- We study the sensitivity of AL seeding for the 3D semantic segmentation of point clouds and the relevance of 2D images features to select 3D scenes.
- Leveraging these studies, we propose a general approach to seed any AL method for point clouds originating from scenes with available image views.
- We present experimental evidence of the effectiveness of our AL seeding method, which consistently improves over random seeds and 2D-inspired baselines, requiring less iterations (thus less annotations) and/or reaching higher segmentation performance.

2. Related Work

We discuss in this section different approaches to reduce the annotation work on point clouds, then focus on Active Learning methods, and finally existing works on seed construction for active learning—even though none of them consider point cloud annotations.

Towards less supervision for 3D point clouds. The best approaches for 3D semantic segmentation of dense indoor point clouds [55, 17, 87, 70] or sparse outdoor point clouds [79, 32, 89, 68] are trained under full supervision, i.e., they require manual annotations of all points in all scenes, which is a notoriously costly task [5, 26]. Several approaches are currently explored to mitigate these costs. For example, one can leverage self-supervision [76, 86, 35, 42, 51, 61] to pre-train a neural network using many unlabeled data with an annotation-free pretext task and then fine-tune this network using few annotations for the task of interest. One can also rely on domain adaptation techniques [83, 59, 60, 77, 80] to exploit existing annotated datasets on a source domain and avoid annotating a new dataset for a target domain. Weak supervision avoids the burden of complete point cloud annotation by, e.g., annotating a few points [45, 78, 16, 85], annotating few regions in a scene [44, 74], or using scene level labels [58, 74]. Alternatively, active learning techniques, which we describe next, allow for a smart selection of the data to be annotated.

Active Learning for 3D point clouds. To minimize annotation costs, active learning strategies [9, 25, 62] aim to select the most relevant data to annotate for the considered model and task. Most selection strategies rely on diversity criteria [25, 62, 3] or on a measure of the uncertainty of the model [9, 72]. In computer vision, active learning has been mostly developed for classification tasks of 2D images [63]. It has been adapted to the semantic segmentation of 3D point clouds combined with markov random fields [46] and more recently deep learning for object detection [21, 47] and semantic segmentation [43, 75, 34]. Diversity of selected scenes has been ensured by selecting core sets [62, 75] in the feature space—in that space each scene is represented by aggregating the features of its corresponding points. The model uncertainty can be measured over each point class-scores [72, 43]—e.g., using the entropy or the margin between the two highest scores—
Seeding active learning. In [43], the authors improve results by computing the uncertainty scores at the level of pre-computed segments. Requiring several forward passes, ensembling techniques [6, 22, 23] have been used to evaluate the model’s confidence for 3D scenes [21, 75]. Moving away from scene-based AL, [65, 75, 34] propose region-based strategies—which mix difficulty and diversity—that allow them to further reduce cost by selecting only local set of points to be annotated. In this work, instead of proposing a new AL strategy, we show that a smart selection of the first set of data with SeedAL—usually over-looked and randomly selected—can boost all methods drastically.

Cold start problem. The need for an initially annotated fraction of the data to bootstrap an AL method is similar to what has been identified as a cold start problem in collaborative filtering and recommendation systems, where new users start off with an empty profile [49].

In active learning, the term “cold start problem” has in fact been used with two close but different meanings. On the one hand [33, 39, 82], it refers to the strong bias induced on the first AL iterations by a too small initial annotated set [19]. In particular, uncertainty-based methods are not effective when trained with too little data [30]. Note that, in this setting, a seed has nevertheless to be given as input, although if it is possible to set its size automatically to limit this bias, by deciding when to stop annotating the first data fraction constituting the seed [24]. On the other hand [84, 13], the cold start problem also refers to the lack of a priori information to select a relevant seed in the first place, which is what we are addressing here.

Seeding active learning. Despite the high variance of performance with random seed selection, it appears it is difficult for AL approaches to create good seeds [13]. In fact, only few recent approaches propose to automatically construct AL seeds [84, 54, 40, 29, 48, 81, 13, 50]. To the best of our knowledge, none applies to 3D point clouds, and there is only one concerning semantic segmentation [50].

All approaches, except [50], only address text or image classification, which is a much coarser-grained task than semantic segmentation. The general idea of most approaches for image classification [54, 40, 29, 81, 13] is to use a self-supervised pretrained network, clustering the image features to estimate diversity groups and picking images close to cluster centers to get representatives, possibly focusing on hard-to-contrast and diverse data [13] by exploiting contrastive self-supervised features [14]. Alternatively, [48] creates seeds using a core-set approach based on the Wasserstein distance between feature distributions of candidate data for the seed and the unannotated dataset. While this form of feature clustering or distance minimization between distributions also makes sense for 3D data, semantic segmentation requires extracting much more detailed information from each single datum to account for complex scenes (whether indoors or outdoors) that capture sets of objects, as opposed to object-centered pictures targeted by image classification in other approaches.

The only AL seeding approach we know for 3D, and in fact for semantic segmentation too, applies to dense 3D medical images (CT-scans) [50], which significantly differ from sparse 3D point clouds captured by depth cameras or lidar scanners. The seeding strategy in [50] is based on a heuristic pseudo-labeling defined by a hand-parameterized, rule-based segmentation. It consists in thresholding CT data within a typical window of values for abdominal soft tissues, then extracting largest connected components and selecting major organs of the abdomen as foreground. This pseudo-labeling is specific to that peculiar kind of medical images and cannot be directly transposed to point clouds, although heuristic hand-designed pre-segments have also been used on 3D point clouds for label-efficient semantic segmentation [44].

3. Preliminary Study

In this section, after introducing our notations and formally describing the problem, we highlight the sensitivity of various AL methods to the seed and then motivate the use of DINO [12] features for the selection of the seed. We use the results of this preliminary study to construct our method, SeedAL, which is fully described in the next section.

3.1. Problem setup and notations

Our method works in the following generic setting. We assume that a scene $i$ is captured by a depth sensor, providing a point cloud $P^i$, and by one or multiple cameras, providing several views $V_1^i,\ldots,V_N^i$ of the scene. For example, RGB-D cameras capture conveniently both the depth and the views simultaneously. Such a multi-modal setting is common, whether indoors [2, 18] or outdoors [5, 10].

Our dataset is made of a set of point clouds, with their respective views, and our goal is to select a good seed $S$, i.e., the initial set of the point clouds which will be fully annotated for the first active learning cycle. As mentioned earlier, our approach for selecting $S$ is based on a pre-trained self-supervised image backbone. We denote this backbone $\phi(\cdot)$ and use it to extract $D$-dimensional $\ell_2$-normalized features from each view: $\phi(V^i_s) \in \mathbb{R}^D$.

3.2. The high variance induced by random seeds

Most AL publications use as AL seeds random fraction of the dataset — which is one of the many issues for reproducibility [40, 36]. A seed is typically picked within a maximum size, given as an initial annotation budget. For AL method evaluation, because of a high variance level due
Figure 2: Influence of the choice of the seed. We visualize results obtained with several AL methods (introduced later in section 5) when using 4 different random seed. With the exception of ReDAL, which is consistently better, the ranking of the methods varies with the seed.

to the random AL seed, a common practice is to average results over few runs, randomizing the AL seed and the training iterations [24, 41, 11, 40, 54, 66, 36]. Even so, it can be not enough given the variance level. As shown in Figure 1, it is particularly true in our problem. Besides, nothing prevents from drawing an “unlucky” seed, leading to significant underperformance or annotation overheads.

We first consider the common case of random seeds, i.e., picking random scenes until the annotation budget is reached. In Figure 1, we illustrate the effect of drawing 20 such random seeds: it shows significant variations of performance, both at initial stage and after some iterations. Furthermore, as shown in Figure 2, the performance hierarchy of scene-based AL methods differs depending on the seed.

A natural way to deal with both intra-method and inter-method variations is to consider the average and variance of the method performance over a large-enough number of random seeds. It allows evaluating the sensitivity of an AL method to the seed and it allows a better comparison of different AL methods, although still with a high variance. However, it does not provide any practical hint how to choose and cold-start these methods. Besides, it is suboptimal as it does not make the most of best seeds.

3.3. Image features to characterize 3D scenes

At first glance, it seems more natural and appropriate to directly use 3D features, as opposed to rely on extra 2D images. However, 3D backbones are not as versatile as 2D ones because of the huge domain gaps in 3D data, compared to picture data. 3D data require specific architectures and tuning due to the extreme variations of 3D point distribution, depending on the type of scenes (indoors vs outdoors, static vs dynamic) and on the kind of sensor used (photogrammetry, variants of depth cameras, lidars with widely differing acquisition patterns and resolutions). Besides, current self-supervised 3D features have a lower quality than their 2D counterpart in the sense that self-supervised 3D features can be improved by distilling, in 3D, high-quality 2D image features of views of the same scenes [61]. In fact, the quality of self-trained 2D image features [28, 27, 12, 4], is now such that, even if self-trained on different datasets, it can be used for object discovery in new data [67, 73, 53]. The best results have actually been achieved with DINO [12]; [81] also exploit DINO in their AL method. There is no such result with pretrained 3D features yet. Last, 2D backbones are pretrained efficiently and on very large datasets, which boosts quality and favors generalizability, compared to the computation burden and practical size of pretraining data in 3D.

Here, rather than relying on a distillation from 2D to 3D as in [61], it is simpler and more efficient to directly use 2D image features to select corresponding 3D fractions of the dataset. Moreover, it makes our approach agnostic to the 3D backbone, which avoids having to handle the domain gaps of 3D data. Last, we can leverage very large image datasets, such as ImageNet, as it has been shown to transfer well to tasks on other image datasets [12]; pretraining specifically on the images of the target dataset is not needed, and could even be detrimental if the dataset is not large enough.

3.4. Relevance of DINO features

To evaluate the effectiveness of DINO features, we first select a subset of the random seeds generated previously. Next, we create the most similar seed (in terms of DINO features) to each selected random seed, and run active learning methods with all of the seeds. Finally, we calculate the correlation between the random seeds and their newly generated most similar correspondences. More specifically, for
each selected seed, denoted as $S$, we replace all scenes in each seed by identifying alternative scenes with the minimum distance in DINO space, and create a new seed denoted as $\tilde{S}$. Formally, for each scene $i$, we compute the average feature $\Phi^i = \sum_{n=1}^{N_i} \phi(V^i_n)/N_i$ of all views. Then, we create $\tilde{S}$ by selecting, for each scene $i \in S$, the scene arg min$_{j \in \tilde{S}} ||\Phi^j - \Phi^i||_2$. Then, we apply several active learning (AL) methods to the newly created initial sets, and compare the results obtained from $S$ and $\tilde{S}$. Our analysis reveals a strong positive correlation between the two sets of results with a Pearson correlation coefficient of 0.78. The results indicate the applicability of leveraging DINO features for selecting the initial seed.

4. SeedAL: Method

Our approach to seed selection posit that a good set should have two key characteristics: Firstly, each scene within the set should exhibit a sufficient level of internal diversity. Secondly, there should be a notable degree of diversity between the scenes themselves. To identify the scenes that should be in the seed, we represent the unlabeled dataset as a fully connected graph $G$, in which each node represents a scene. The edge weights, denoted by $e_{ij}$, between two scenes $i$ and $j$, encapsulate both intra and inter-scene diversity measures. These measures are formally introduced below. By looking for a sub-graph that optimizes the sum of edge values while satisfying a budget constraint, we can find a seed with good diversity. We propose a linear optimization framework for the selection of this sub-graph and explain its details in the following paragraphs. We illustrate our method in Figure 4.

4.1. Diversity measures

Intra-scene diversity measure. Given that a scene may contain near-duplicate views, e.g., captured from very similar viewpoints, it is important to eliminate redundancy to obtain representative features that accurately capture the characteristics of the scene. To this end, for each scene $i$, we cluster the set of view-features $\phi(V^i_1), \ldots, \phi(V^i_{N_i})$ in $K$ clusters using k-means. The resulting set of representative features, denoted by $\Phi^i_1, \ldots, \Phi^i_K$, contains $K$ distinctive features of the scene. To calculate the diversity within a scene $i$, we calculate the pairwise dissimilarities between the cluster centers $\Phi^i_1, \ldots, \Phi^i_K$: $(1 - \Phi^i_k \cdot \Phi^i_{k'})$ for all $k > k' \in \{1, \ldots, K\}$, where "." denotes the scalar product in $\mathbb{R}^D$. Finally, we average all these pairwise dissimilarities to obtain the internal diversity $d_i$ of scene $i$:

$$d_i = \frac{2}{K(K-1)} \sum_{k=1}^{K} \sum_{k'=k+1}^{K} (1 - \Phi^i_k \cdot \Phi^i_{k'}) . \tag{1}$$

Inter-scene diversity measure. Given two different scenes $i$ and $j$, we compute their inter-scene diversity $d_{ij}$ by relying on their respective cluster centers: $\Phi^i_1, \ldots, \Phi^i_K$ and $\Phi^j_1, \ldots, \Phi^j_K$. Specifically, we average the pairwise dissimilarities between all pairs of cluster centers:

$$d_{ij} = \frac{1}{K^2} \sum_{k=1}^{K} \sum_{k'=1}^{K} (1 - \Phi^i_k \cdot \Phi^j_{k'}) . \tag{2}$$

Combined intra and inter-scene diversity measure. To combine the intra and inter-scene diversities, we simply compute $e_{ij} = d_{ij} d_i d_j$. Hence, scenes $i$ and $j$ in the graph $G$ are highly connected if they are both highly diverse internally and mutually diverse.

4.2. Linear optimization for seed selection

We cast our seed selection as the identification of a fully connected subgraph of $G$ such that the sum of edge weights is maximized under the constraint of a budget $b$ for the initial annotations. Here, we define this budget in terms of the number of points to be annotated. The connected subgraph of $G$ could be identified by solving the following problem:

$$\arg \max_{\{x_i\}_i} \sum_{i=1}^{M} \sum_{j=i+1}^{M} e_{ij} x_i x_j \quad \text{s. t.} \quad \sum_{i=1}^{M} c_i x_i \leq b . \tag{3}$$
where the $x_i$ are boolean variables, indicating whether scene $i$ is selected or not. $c_i$ the annotation cost for scene $i$, which in practice we take as the number of points in scene $i$. The objective term $\sum_{i=1}^{M} \sum_{j=i+1}^{M} e_{ij} x_i x_j$, where $M$ is the number of scenes, permits us to select scenes with the highest diversity possible while the constraint $\sum_{i=1}^{M} c_i x_i \leq b$ ensures that we do not exceed our annotation budget $b$.

However, this is a quadratic problem and $M$ is large in practice. We can turn this problem into the linear problem

$$\arg \max_{\{y_{ij}\}_{i,j>i}} \sum_{i=1}^{M} \sum_{j=i+1}^{M} e_{ij} y_{ij} \text{ s. t. } \begin{cases} \sum_{i=1}^{M} c_i x_i \leq b \\ y_{ij} \leq x_i \\ y_{ij} \leq x_j \end{cases}$$

by introducing the boolean variables $y_{ij}$. Because it is an optimization problem and thanks to the additional constraints, $y_{ij} = 1 \iff x_i = x_j = 1$. The advantage of this formulation is that it is much more efficient to solve when the number of scenes $M$ is large.

In practice, we observed that it can still be computationally challenging for very large $M$. To address this issue, we first extract the top $L$ edges in $G$ with the highest weights $e_{ij}$ and collect all scenes on both ends on these edges. Then, we rebuild a smaller graph using only these scenes and apply the optimization to this reduced graph. We use the solver from SCIP [7] to find the optimal solution.

5. Experiments

We present in this section the experimental protocol that we use to evaluate our method SeedAL and discuss results. We also provide an ablation study.

5.1. Experimental setup

Active learning strategies. We test our method with diverse active learning strategies that exploit either the model uncertainty or enforce diversity, that are either scene-based or region-based. In particular, we compare to a random selection of scenes (rand), the diversity-based method CoreSet [62] (CoreSet) and the uncertainty-based methods Softmax Confidence [72] (S-conf), Softmax Margin [72] (S-margin), Softmax Entropy [72] (S-ent), MC-Dropout [23] (MC-Drop) and Segment Entropy [43] (SegEnt). We also consider the region-based method ReDAL [75] (ReDAL).

Seeding baselines. We compare SeedAL to two clustering-based baselines inspired from [84]. For this purpose, for each scene $i$, we first compute the average feature $\Phi_i = \frac{\sum_{n=1}^{N_i} \phi(V^n_{ik})}{N_i}$ of all views. The first simple baseline, KMcentroid, starts with a clustering of the features $\Phi_1, \ldots, \Phi_M$ into $K$ clusters using k-means. Then, for each cluster, we search the scene $i$ whose feature $\Phi_i$ is the closest to the cluster centroid, we select these $K$ scenes, and repeat this search process until we reach the annotation budget. The most costly samples are then removed iteratively until the annotation budget is satisfied. We also consider a variant of KMcentroid, called KMfurthest, inspired by the core-set [62] selection. It starts as in KMcentroid with a selection of the $K$ scenes closest to each cluster centroid. Then, we continue spending the annotation budget by selecting, iteratively, the scene $j$ whose feature $\Phi_j$ is the farthest away from all cluster centroids, to favor a selection of diverse scenes. We also compare to the natural baseline random.

Datasets. Following [75], we focus on the semantic segmentation task and evaluate SeedAL on two datasets representative of indoor and outdoor scenes, namely S3DIS [2] and SemanticKITTI [5]. S3DIS is composed of 271 scenes extracted from 6 major indoor areas; for each scene is provided a dense point cloud with color. As common practice, we provide AL results on the Area 5 validation set, and use the rest for training. We also evaluate on SemanticKITTI, a large-scale autonomous driving dataset composed of 22 driving sequences, totaling 43,552 point clouds, each accompanied with images. Following the official protocol, we evaluate on the validation split (seq 08) and train the models on the entire official training split (seq 00-07 and 09-10).

Network architecture and evaluation metric. Following previous works [75], we consider SPVCNN [69], that is based on point-voxel CNN and achieves good results both on indoor and outdoor scenes. We evaluate the semantic segmentation task in all experiments with the mIoU metric.

Technical details. In all our experiments, we set $K$ as the number of the classes for each dataset: 13 for S3DIS, 19 for SemanticKITTI. $L$ is set as 100 for S3DIS and 1000 for SemanticKITTI. On S3DIS, the RGB images depicting each scene constitute our views. We use the class token extracted at the last layer of the DINO-pretrained ViT-B/8 as our view features. On SemanticKITTI, each scene is depicted by a single image frame. To account for multi-object images in outdoor scenes, we consider that each $8 \times 8$ patches of a frame constitute one view and we take the corresponding patch feature at the last layer of the ViT-B/8 as the view feature. Also, as there is a significant number of redundant and highly similar scenes in SemanticKITTI, we use a greedy algorithm to sparsify the sequences. The details of this sparsification are in the supplementary material.

5.2. Indoor evaluation

We evaluate here our method on S3DIS. We present in Figure 5 the results obtained with our AL seeding strategy, SeedAL, and with the considered baselines. First, we observe that using SeedAL boosts significantly all 8 AL methods evaluated over randomly selected seeds (dashed blue curves). SeedAL is consistently at the top of the graphs while random seeds yields, overall, unstable results.
Regarding our baselines, KMfurthest achieves consistently better performances than KMcentroid over all AL strategies. We explain this result by the fact that KMfurthest contains more diverse scenes than KMcentroid. We observe that for certain methods, KMfurthest achieves similar levels of performance as SeedAL. However, it is important to note that SeedAL is more robust and it consistently obtains strong performance in both early and late active learning iterations.

**Making a good first impression.** We highlight here the boost obtained with SeedAL at the *very first cycle* (when no AL has been applied yet). We observe that with SeedAL, results are +1.9 pt over the best random seed, +7.8 pt v.s. the average of random seeds and +18.8 pt when comparing to the most unlucky seed. We surpass also our two baselines KMcentroid and KMfurthest by 4.9 pt and 2.4 pt, respectively. With only 3% of data well-selected thanks to SeedAL, we obtain better results than with 4 out of 5 random seeds with 5%, or even 7% (when employing S-conf, S-margin, S-ent, CoreSet and MC-Drop), of the data. These
results prove that our seeding selection is a much better strategy than betting on the quality of the random seed.

We also compare our method with SSDR-AL [64]. Differently than other AL methods, SSDR-AL’s initialization picks superpoints to label, rather than full scenes. As our current formulation selects scenes rather than regions, it cannot directly be applied in this case. The only way we can compare quantitatively with [64] is on S3DIS regarding the proportion of points to label in order to reach the same level as 90% of full supervision: we boost ReDAL to outperform SSDR-AL, only requiring 9% labeled points instead of 11.7%, as shown in Table 1.

<table>
<thead>
<tr>
<th>Method</th>
<th>SSDR ReDAL</th>
<th>ReDAL (rand)</th>
<th>ReDAL (ours-DINO)</th>
<th>ReDAL (ours-MoCo)</th>
</tr>
</thead>
<tbody>
<tr>
<td>% labeled pts (%)</td>
<td>11.7 %</td>
<td>13%</td>
<td>11%</td>
<td>9%</td>
</tr>
</tbody>
</table>

Table 1: Comparing SSDR’s initialization with SeedAL in terms of % of pts to label to get to 90% of full supervision.

5.3. Outdoor evaluation

We now focus on the challenging SemanticKitti dataset, which depicts highly redundant urban images. We present results in Figure 6. We also show results with the baseline rand, which is known to surpass all AL methods except the region-based ReDAL.

Boosting active learning. First, we observe that SeedAL significantly surpasses all seeding baselines, either random or diversity-based. Second, it achieves better results compared to any randomly sampled seed with all AL methods and at all cycles but one (with ReDAL). SeedAL yields an improvements of 3.3 pt and 9.7 pt in the first cycle over the best and worst random seed, respectively. We also observe that the results obtained with KMcentroid and KMFurthest are rather disappointing: the AL results are falling below the worst random seed at the first and second cycles. Surprisingly, ReDAL reaches better results with KMFurthest when using 3% of the dataset. However, our seeding strategy appears to be agnostic to the AL method and boost all of them consistently. Overall, the best results are again obtained with our seeding strategy and ReDAL.

SeedAL is especially effective for large-scale datasets, enabling scene-based AL methods to achieve almost 90% of the fully supervised performance using just 4% labeled points. This is comparable to the region-based method ReDAL which achieves 93% of the fully supervised performance. SeedAL bridges the gap between complex and costly methods and simpler AL methods on the large-scale SemanticKITTI dataset.

Improving random selection. rand is known to be a hard-to-beat baseline on this dataset [75]. We observe that thanks to SeedAL, the performance of rand goes as high as 58.0%, v.s. 58.9% for the best considered method (ReDAL). The baseline rand is not itself an active learning strategy — given that scenes are randomly sampled — but these results show that a good seeding strategy impacts greatly all results, even those of the random baseline.

We also compare our method with LiDAL [34] on SemanticKITTI. LiDAL is an AL method specifically developed for 3D LiDAR semantic segmentation. As it is shown in Table 2, LiDAL benefits more from SeedAL than random selection. It gets 90% of full supervision’s mIoU (58.0%) with 2.1% labels, vs 2.6% with a random seed.

<table>
<thead>
<tr>
<th>Seeding method</th>
<th>Init (1%)</th>
<th>2%</th>
<th>3%</th>
<th>4%</th>
<th>5%</th>
</tr>
</thead>
<tbody>
<tr>
<td>rand [34]</td>
<td>48.8</td>
<td>57.1</td>
<td>58.7</td>
<td>59.3</td>
<td>59.5</td>
</tr>
<tr>
<td>SeedAL (ours)</td>
<td>52.6</td>
<td>57.8</td>
<td>59.3</td>
<td>60.3</td>
<td>60.6</td>
</tr>
</tbody>
</table>

Table 2: LiDAL [34] with random or SeedAL’s seeds.

5.4. Ablation experiments

We study the relevance of the different components used to construct SeedAL. We conduct experiments on S3DIS with all the active learning methods described in Sec. 5.1. Due to space constraint, we report CoreSet and ReDAL results and include the others in the supplementary material.

Intra-scene diversity. We highlight the relevance of selecting scenes with high intra-scene diversity instead of high inter-scene similarity. We compute a diversity and a similarity score per scene. Then we select the top scenes according to each metric until exhausting our annotation budget. The diversity and similarity scores are respectively computed by averaging \((1 - \phi(V_i^k, \cdot) \cdot \phi(V_j^k, \cdot))\) and \(\phi(V_i^k, \cdot) \cdot \phi(V_j^k, \cdot))\) between all views of a scene. Results in Figure 7 (a) show that using the intra-scene diversity yields the best results.

Clustering views. We show that computing the dissimilarity score from the cluster centroids as in (1) is a better strategy than computing it as in the previous paragraph. Performances obtained with both strategies are presented in Figure 7 (b). Clustering removes redundant features and yields a better estimate of the scene diversity, hence selection of a significantly better seed as demonstrated by the results for 3% of labeled points.

Inter-scene diversity. We justify why we select scenes with large inter-scene diversity rather than scenes with large inter-scene similarity. In a first experiment, we replace the edge weight \(e_{ij}\) in problem (4) by \(d_{ij}\), the inter-scene dissimilarity of (2), and, in a second experiment, by the inter-scene similarity, obtained by averaging \((\Phi_i^k - \Phi_j^k)\) over all pairs \((k, k')\). Figure 7 (c) shows that selecting scenes with high inter-scene dissimilarity yields the best seeds.

Combined intra and inter-scene diversity. Finally, we show the interest of combining the intra and inter-scene diversity, i.e., using SeedAL, in Figure 7 (d). In this figure, we compare the performance of seeds selected by solving
We evaluate here results obtained with different seeding strategies. (a) Seeds made of scenes with high intra-diversity (intra-div.) or high intra-similarity (intra-sim.). (b) Seeds selected with two different intra-diversity metrics: view features (feats.) or computed after clustering the view features (cls. feat.). (c) Seeds made of scenes with high inter-diversity (inter-div.) or high inter-similarity (inter-sim.). (d) Seeds selected with SeedAL, considering only inter-diversity (inter-div.) or intra-diversity (intra-div.).

SeedAL results on S3DIS (first row) and SemanticKITTI (second row) using features from DepthContrast, SegContrast, ALSO, MoCo-v3, DINO. ‘random’ is an average over the random seeds.

Next we show the effect of different 2D and 3D self-supervised features to construct SeedAL. Due to space constraint, we report results from two different methods for each dataset and include the rest in supplementary material.

SeedAL with different 2D features. We experimented with self-supervised 2D features from MoCo-v3 [15] as an alternative to DINO. The results are reported in Figure 8. On SemanticKITTI, SeedAL with MoCo-v3 performs on par or slightly better than SeedAL with DINO, but is slightly inferior to SeedAL with DINO on S3DIS, except for ReDAL. DINO and MoCo-v3 are pretrained on ImageNet, which is very different from S3DIS and SemanticKITTI; it shows the foremost generalizability of these features.

SeedAL with pure 3D features. We also experimented with self-supervised 3D features: DepthContrast [86] pretrained on ScanNet for indoor RGBD (no model is available for outdoors), as well as SegContrast [51] and ALSO [8] pretrained on SemanticKITTI. To adapt SeedAL to work with 3D features, we simply replace the set of 2D image features of a scene with 3D point features obtained from the point cloud of the same scene, which we thus average to obtain whole-scene features. Results are illustrated in Figure 8: DepthContrast is only slightly below 2D features, SegContrast is not as good but still better than rand, while ALSO underperforms, likely because it is not contrastive. Note that there is no pretrained 3D network that works both for indoors and outdoors, contrary to 2D-based models.

6. Conclusion

In this work, we have shown the influence of seeds on the performance of AL methods for point clouds, and proposed a method for efficiently selecting a seed that yields to good performance. Our approach works out of the box: it is agnostic to the AL method and to the 3D backbone used for the downstream task, and it does not require training on the dataset. We hope it will help scaling up point cloud semantic segmentation for practical applications.

Acknowledgments

We thank Samet Hicsonmez for insightful discussions. This work was granted access to HPC resources of IDRIS under GENCI allocations 2023-AD011013267R1, 2023-AD011013262R1 and 2023-AD011013413R1.
References


[38] Ajay J. Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-Class Active Learning for Image Classification. In Conference on Computer Vision and Pattern Recognition (CVPR), 2009. 1


[40] Adrian Lang, Christoph Mayer, and Radu Timofte. Best Practices in Pool-Based Active Learning for Image Classification, 2022. OpenReview. 3, 4


[77] Qianguang Xu, Yuchen Li, Houwen Peng, Jinjie Mai, Yuan Yuan, James L. Crowley, and Dominique Vaufreydz. PointContrast: Unsupervised Pre-Training for 3D Point Cloud Understanding. In International Conference on Computer Vision (ICCV), 2021. 2
[79] Xu Yan, Jiatao Guo, Chaoda Zheng, Chao Zheng, Ruimao Zhang, Shuguang Cui, and Zhen Li. 2DPASS: 2D Priors Assisted Semantic Segmentation on LiDAR Point Clouds. In European Conference on Computer Vision, 2022. 2


[85] Yachao Zhang, Zonghao Li, Yuan Xie, Yanyun Qu, Cuihua Li, and Tao Mei. Weakly Supervised Semantic Segmentation for Large-Scale Point Cloud. In American Association for Artificial Intelligence Conference, 2021. 2


