COARSE3D: Class-Prototypes for Contrastive Learning in Weakly-Supervised 3D Point Cloud Segmentation

Rong Li[1]
selirong@mail.scut.edu.cn
Anh-Quan Cao[2]
anh-quan.cao@inria.fr
Raoul de Charette[2]
raoul.de-charette@inria.fr

[1] South China University of Technology
Guangzhou, China
[2] Inria
Paris, France

Abstract

Annotation of large-scale 3D data is notoriously cumbersome and costly. As an alternative, weakly-supervised learning alleviates such a need by reducing the annotation by several order of magnitudes. We propose COARSE3D, a novel architecture-agnostic contrastive learning strategy for 3D segmentation. Since contrastive learning requires rich and diverse examples as *keys* and *anchors*, we leverage a *prototype memory bank* capturing class-wise global dataset information efficiently into a small number of prototypes acting as *keys*. An *entropy-driven sampling* technique then allows us to select good pixels from predictions as *anchors*. Experiments on three projection-based backbones show we outperform baselines on three challenging real-world outdoor datasets, working with as low as 0.001% annotations.

1 Introduction

Semantic segmentation is the holy task of any scene understanding system. For driving, this is of utmost importance, because driving requires a fine-grained 3D understanding for navigation and planning. Subsequently, in recent years, 3D semantic segmentation attracted a plethora of works [15, 29, 74, 93]. However, the vast majority of the literature relies on supervised learning – thus assuming dense point-wise semantic labels – which are laborious and costly to acquire. A striking example is the popular urban dataset SemKITTI [3], which took more than 1700 hours to label by human operators.

To alleviate this cost, weakly-supervised 3D segmentation uses several orders of magnitude fewer labels [30, 44, 73, 77]. Still, researches are mostly focused on point-based backbones which are efficient but often slow at inference. Conversely, projection-based methods [1, 15, 46, 83] were shown to operate at high speed with competitive performance [15].

In this work, we propose an architecture-agnostic contrastive learning framework for weakly-supervised semantic segmentation. Since our approach does not alter the backbone segmentation, we demonstrate its usage on a lightweight projection backbone, thus enabling both efficient trainings with extreme scarce labels and fast inference. Of paramount importance, contrastive learning needs abundant representative examples [12, 26, 79] which is non-trivial to get in weakly-supervised settings. To solve this we introduce two mechanisms. First, we propose a *prototype memory bank* that stores the global dataset per-class information into a small number of rich and compact prototypes. Second, we employ an entropy-driven sampling to select sufficiently good *anchors* from inaccurate predictions. Experiments show that our method outperform existing weakly-supervised baselines on three real-world outdoor datasets SemKITTI [3], SemPOSS [49], and nuScenes [8]. On SemKITTI we also show that we *on par* with full-supervision (i.e., 100% labels) performance, using only 1% of the labels. **Our code is released at:** `https://github.com/cv-rits/COARSE3D`.

In brief, our contributions can be summarized as follows:

- We propose an architecture-agnostic framework for weakly-supervised 3D point cloud segmentation and, to the best of our knowledge, first, demonstrate its usage on a lightweight projection backbone.
- We introduce a *prototype memory bank* that captures per-class dataset information with an *entropy-driven sampling* technique to sample more confident pixels as anchors.
- Our strategy significantly outperforms all comparable baselines on three real-world outdoor datasets, with as few labels as 0.001% – roughly corresponding to a single labelled point per frame.

2 Related Work

Label-efficient 3D Semantic Segmentation. With the abundant annotated large-scale 3D datasets [3, 8, 19, 41, 49], point cloud semantic segmentation advanced rapidly. The methods can be categorized into point-based [14, 29, 37, 53, 63, 64, 65] and projection-based [1, 15, 46, 74, 75, 83] approaches. To learn from fewer labels, various strategies were proposed with point-based methods: enforcing geometric prior [85], local neighborhood propagation [30, 44], for smarter use of the rare labels some used active learning [77], pseudo labelling [38, 59, 73], graph propagation [44, 59], self-training [38, 89], temporal matching [59, 67].

Contrastive Learning for 3D. Contrastive learning [11, 12, 26], initially designed for 2D, was extended to 3D using similar points from different views in [27, 36, 81, 91]. Limited training data is considered in [27]. Invariant representation is learned from spatio-temporal cues in [13, 31, 40]. Pairs of point and pixels regions were matched in [56] while point-pixel pairs are regarded as a whole in [43]. We focus the literature on crucial aspects for our work.

Sampling Strategy is crucial for constrastive learning that requires many good examples [12, 26, 34, 79]. Prior knowledge is used to guide sampling in [4, 25]. Multiple pairs of positive and negative data are sampled in [7, 62]. To improve convergence, hard negative sampling is exploited in [6, 33, 35, 52, 57, 60] and point proxies serve as examples in [47]. To avoid local minima, semi-hard negative sampling is proposed in [9, 57, 80]. In [17], false positive images are used as hard negative examples. Since noisy samples were shown to deteriorate performance [76], hard positive/negative examples sampling is applied in [35] while high-quality anchors are sampled by considering correct prediction in [70]. We instead propose entropy-driven anchors sampling techniques to learn more robust representations.

Memory Bank introduced in [79] allows storing many examples, *e.g*. pixels in 2D, typically sampled per image [10, 71, 82] or mini-batch [70] to increase variety. However, such storage comes at the cost of large memory and asynchronous updates. These issues are mitigated using momentum encoder [12, 26] or keeping examples in recent batches [12, 26, 72]. Pixels

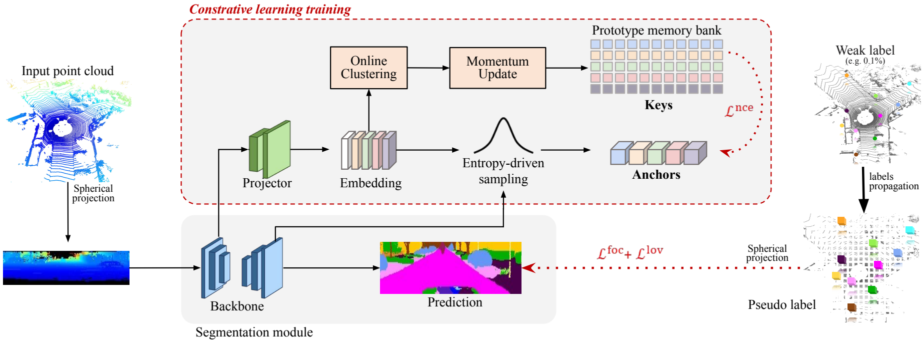


Figure 1: Our COARSE3D uses a contrastive learning strategy (Sec. 3.1) to learn high dimensional semantic embeddings from the encoder features, later clustered into a lightweight prototype memory bank (Sec. 3.2). At each step, contrastive learning is applied on a scarce set of ‘anchors’, selected from an entropy-based sampling (Sec. 3.3), which features are pushed towards (pulled away from) positive (negative) prototypes acting as ‘keys’.

are selected randomly [70] or selectively [2] from each image. Still, only a few pixels are stored due to memory constraints, losing full dataset content. Moreover, using pixels embeddings is redundant [70]. Regions [28, 70] can be employed as more compact representation but not with label sparsity settings due to the need of per-pixel annotation. Hence, we propose an efficient memory bank, storing prototypes (instead of pixels) that can capture the class-wise dataset-wise information using only a small number of prototypes.

Prototype Learning aims to represent the embedding space as a set of prototypes. It is used for many tasks like classification [16, 21, 22, 23, 45, 55, 78, 87], few-/zero-shot [6, 32, 61], regression [45], unsupervised learning [79, 84], explanability [39], and semantic segmentation [20, 68, 69, 92]. Different from the literature, we use prototypes as a memory bank to gather the rich class-wise global context information from the whole dataset efficiently.

3 Proposed approached

We address the problem of 3D semantic segmentation of point clouds and propose, a novel architecture-agnostic training strategy for weakly-supervised learning coined ‘COARSE3D’. To learn efficient semantic representation from scarce labels, we leverage a custom contrastive learning module whose originality resides in a lightweight memory footprint and sparse embedding optimization. Results show our methodology is efficient with as little as 0.001% of the labelled data – stepping ahead of prior works [30, 85, 88, 91].

COARSE3D, depicted in Fig. 1, uses a custom contrastive module (Sec. 3.1) that projects features of the segmenter encoder into a different embedding space. As for the segmenter backbone, we employ the fast and memory efficient SalsaNext [15] which processes 3D point cloud as 2D range images. Rather than the greedy pixel-wise memory [70], our projected features are clustered into a prototype memory bank (Sec. 3.2) that captures compact class-wise semantic embeddings. This enables preserving the dataset context in a significantly lighter fashion. In each iteration, we optimize only a small subset of pixels (i.e. anchors) sampled with an entropy-based strategy (Sec. 3.3) considering prototypes as positive/negative samples (i.e. keys). Our training strategy (Sec. 3.4) combines standard segmentation and contrastive losses, along with a simple labels propagation strategy.

3.1 Contrastive Learning Pipeline

We leverage a contrastive learning scheme since prior works have shown its effectiveness to learn from limited labels [27, 54, 81]. We build on the standard InfoNCE pixel-wise contrastive loss [24, 48] but apply the latter between our class-wise representations, hereafter *prototypes* (Sec. 3.2), and a subset of sampled pixels (Sec. 3.3).

More specifically, at each training iteration we select a set of pixels, called *anchors* having features $\mathcal{A}=\{a_j,\ldots,a_{N_a}\}$. For each anchor, we select the corresponding *prototypes* of similar semantic class, i.e. *positive keys* with features $\mathcal{P}^+=\{p_1^+,\ldots,p_{N^+}^+\}$, or different semantic class, i.e. *negative keys* with features $\mathcal{P}^-=\{p_1^-,\ldots,p_{N^-}^-\}$. The InfoNCE optimization objective is to simultaneously pull the embedding of *anchors* towards *positive keys* while pushing them away from the *negative keys*. Formally, it writes:

$$\mathcal{L}^{\text{Pix2Proto}} = \frac{1}{N_\text{a}} \sum_{a_i \in \mathcal{A}} -\log \frac{\sum_{p_j^+ \in \mathcal{P}^+} \exp\left(a_i \cdot p_j^+ / \tau\right)}{\sum_{p_j^+ \in \mathcal{P}^+} \exp\left(a_i \cdot p_j^+ / \tau\right) + \sum_{p_j^- \in \mathcal{P}^-} \exp\left(a_i \cdot p_j^- / \tau\right)}, \quad (1)$$

where τ is the temperature hyperparameter. All feature vectors are extracted by first linearly interpolating multi-scale features to the full image size. Then, they are concatenated and mapped to 256-dim using two 1×1 2D convolutional layers. Finally, they are ℓ^2-normalized. Following previous works [34, 70] that demonstrated the benefit of a custom selection of keys and anchors, we consider a set of anchors sampled using the entropy of the segmentation output and keys from a newly introduced prototype memory bank, that we describe next.

3.2 Prototype Memory Bank

The memory bank requires massive data to learn good representation which is traditionally achieved by storing a large number of pixels as in [26, 70, 72, 79]. Still, this is very costly in terms of memory and computation for contrastive sampling. Furthermore, storing pixels is semantically redundant which can be mitigated by storing semantic regions [70]. However, the latter would be poorly reliable given the limited labels. Therefore, we introduce a *prototype memory bank* that encodes class-wise embedding into compact prototypes.

More in-depth, the memory bank stores $\{P_k \in \mathbb{R}^{N_p \times D}\}_{k=1}^{K}$ prototypes of feature dimension D, corresponding to N_p prototypes per class for each of the K semantic classes. For each semantic class k, at each training iteration, we cluster the N_k labelled pixels of such class into the corresponding N_p class prototypes. We frame this as an optimal transport problem and employ the Sinkhorn algorithm [18], similar to [92] because it balances the pixels assignment between all prototypes. Here, note that the prototypes are solely updated from the embeddings of labelled pixels which guarantees reliable information.

In pratice, we compute the optimal transport matrix $T \in \mathbb{R}^{N_k \times N_p}$ by unrolling three iterations of the Sinkhorn algorithm on the cost matrix $\mathcal{C} \in \mathbb{R}^{N_k \times N_p}$ – where $\mathcal{C}_{ij}$ is the cosine distance between the feature maps of pixel i and prototype j. The pixel-prototype mapping of pixel p_i is obtained with $m(p_i)$=argmax$_j T_{ij}$. In practice, to encourage exploration we use a Gumbel-Softmax (τ=0.5) as differentiable argmax. Given a random initialization, we update each prototype with a $\sigma = 0.999$ momentum towards the mean of the clustered pixels embeddings in each training iteration. Hence, j^{th} prototype $\{P_k\}_j$ of class k is updated as:

$$\{P_k\}_j = \sigma\{P_k\}_j + (1-\sigma)\frac{1}{\sum_{i=1}^{N_k} [\![m(x_i)=j]\!]} \sum_{i=1}^{N_k} x_i [\![m(x_i)=j]\!], \quad (2)$$

with $[\![.]\!]$ the Iverson brackets.

3.3 Entropy-driven Anchors Sampling

Prior works have shown the importance of proper sampling in contrastive learning [6, 9, 33, 57, 62, 70, 80]. However, such strategies are inappropriate in our setup given the limited labels and the need for abundant samples in contrastive learning. Instead, we build on information theory [58] and introduce an *entropy-driven sampling* technique to select the most appropriate pixels to serve as *anchors* among the abundant pseudo-labels predictions.

Given an image x processed by the segmentation module, we estimate the quality of the semantic prediction for pixel x_i using its Shannon entropy $H(x_i) = -\sum_{k=1}^{K} f(x_i)\log(f(x_i))$, where K is the number of classes and $f(x_i) \in \mathcal{R}^K$ is the softmax prediction. Subsequently, we define the sampling probability as $\rho(x_i) = \frac{\exp -H(x_i)^2}{\sum_{x_j \in \mathcal{X}} \exp -H(x_j)^2}$ where $\mathcal{X}$ is set of pixels with the same predicted label as x_i. Intuitively, pixels with low entropy will have higher chances to be selected. The power of 2 is used to make the distribution steeper. Finally, we perform weighted sampling to select the N_a pixels as a set of anchors $\mathcal{A}$. Unlike standard pixel-wise contrastive loss [70] requiring thousands of pixels, we start with only one anchor and linearly increase during training – up to 50% pseudo label. This follows the intuition that the segmentation module improves as training progresses.

3.4 Training Strategy

To train the method, we first apply a simple voxel propagation scheme to expand labels illustrated in Fig. 1 and detailed in Appendix B.1. We then train with two supervised and one unsupervised loss. First, the focal loss [42] is applied on all labelled points to alleviate classes imbalance:

$$\mathcal{L}_i^{\text{foc}} = -\alpha(1-(f(x_i))^{\gamma}\log(f(x_i))\,, \tag{3}$$

where $\alpha = \{w_k \mid y_i = k\}$ is the weight of class k, y_i the pseudo label of point i, and $w_k = \log(1 + \frac{1}{\text{freq}_k})$ with freq_k is the class frequency before voxel propagation. Second, we use the lovász-softmax loss [5] to directly optimize the mean intersection-over-union, defined as:

$$\mathcal{L}_i^{\text{lov}} = \frac{1}{k}\sum_{k=1}^{K} \overline{\Delta_{J_k}}(\tilde{\mathbf{m}}(c))\,, \text{ with } \quad \tilde{\mathbf{m}}_i(k) = \begin{cases} 1 - f(x_i) & \text{if } k = y_i, \\ f(x_i) & \text{otherwise,} \end{cases} \tag{4}$$

where $\overline{\Delta_{J_s}}$ indicates the Lovász extension of the Jaccard index for class k. y_i is the label of point x_i. Finally, the overall training target is

$$\mathcal{L} = \sum_i \left(\lambda_{\text{ce}}\mathcal{L}_i^{\text{foc}} + \lambda_{\text{lov}}\mathcal{L}_i^{\text{lov}} + \lambda_{\text{nce}}\mathcal{L}_i^{\text{nce}}\right)\,. \tag{5}$$

4 Experiments

We evaluate our proposal by comparing against recent baselines on 3 challenging real-world LiDAR datasets, being SemKITTI [3], nuScenes [8] and SemPOSS[49] and report main results in Sec. 4.2 and ablations in Sec. 4.3. For memory-efficient processing, we evaluate solely on projection-based backbones but COARSE3D is applicable to any architecture (*e.g.*, point-based). In particular main results use SalsaNext [15] backbone and we experiment with Rangenet-21 [46] and SqueezeSegV3-21 [83] backbones in ablations. Importantly, all our components are removed at inference, leaving the segmenter backbone unchanged.

4.1 Experimental Setup

Datasets. SemanticKITTI [3] has 22 German scenes recorded at 10Hz with an HDL-64E LiDAR. We follow common practice and train on sequences 00 to 10, except 08 for validation. The *hidden* test set uses sequences 11 to 21. **nuScenes** [8] has 850 scenes from U.S.A./Singapore acquired with a VLP-32 LiDAR, labelled at 2Hz. Again, we follow standards using 700 scenes for training and 150 scenes for validation. **SemanticPOSS** [49] recorded Chinese campus scenes with @10Hz 40-layers LiDAR. Though much smaller, the dataset has many pedestrians. We train on sequences 00 to 05, except 02 for validation.

Model Selection. For SemKITTI [3], we cherrypick the best model after hyperparameter tuning on the validation set. The main evaluation of the chosen model is done using an online official benchmark (hidden test set). Evaluations on SemPOSS [49] and nuScenes [8] use the exact same setting but *without any* hyperparameters tuning thus reporting only results of the *last training model* on their validation sets.

Annotation Strategy. To avoid the human labeling efforts of active learning [77] or one-point-per-object annotation [44], we apply random subsampling to existing dense ground truths (i.e. 100% annotation) to obtain sparse labels (*e.g*. 0.01%) as in [30, 85, 88]. For a given setting, all methods share the same labels.

Implementation and Training. We train on 4 NVIDIA Tesla V100 for 100 epochs, using AdamW with 0.01 learning rate and batch size 16. We balance losses with $\lambda_{\text{foc}} = 1.0$, $\lambda_{\text{lov}} = 1.0$ and $\lambda_{\text{nce}} = 0.1$ and use standard data augmentation as in [15]. Because early prototypes are too noisy, we apply a 5 epochs warm-up without contrastive learning.

4.2 Performance

Quantitative Results. We report performance on SemKITTI[1] [3] *hidden* test set in Tab. 1, and on SemPOSS [49] and nuScenes-Lidarseg [8] using their validation sets both in Tab. 2. On all datasets, we consistently outperform other weakly-supervised baselines being projection-based networks like SalsaNext [15] or point-based networks like SQN [30]. Specifically, comparing Ours and SalsaNext - our backbone - the mIoU is +5.6/+3.6 mIoU on SemKITTI (Tab. 1) with only 0.1%/0.01% annotations and +4.1/+3.7 on SemPOSS (Tab. 2a). On nuScenes (Tab. 2b), we get +2.2 mIoU with 0.1% labels but -1.6% in the 0.01% labels case. We argue the drop relates to nuScenes (24x sparser than SemKITTI) having 10 of the 16 classes with less than 20 labels in the 0.01% setting. Subsequently, our clustering fails to associate labels with our 20 prototypes per class and the latter act as noise. Importantly, in Tab. 1 note that despite the projection-based nature of our backbone we significantly outperform SQN, a point-based method. Furthermore, we even beat *some* recent fully supervised baselines on each dataset despite the fact that we use 1000x or even 10000x fewer labels. This demonstrates our ability to learn robust semantic embeddings with significantly less labels. The ablation in Tab. 4, later discussed, even advocates that 1% of labels is sufficient to perform on par with full supervision.

Qualitative Results. Fig. 2 shows qualitative outputs on SemKITTI [3] validation set, along with ground truth and SalsaNext [15] baseline given the absence of public implementation for SQN [30]. In both 0.1% and 0.01% settings, our method surpasses SalsaNext, especially on thin objects. More qualitative results are in the Appendix C.

[1] For fair comparison with our SalsaNext backbone on the *hidden* test set of SemKITTI we also use the kNN post-processing [46] to mitigate the risk of overlapping points in projection-output having same 3D label.

Anno.(%)	Method	Projection-based	mIoU (%)	road (19.87%) ■	sidewalk (14.39%) ■	parking (1.47%) ■	other-grnd (0.39%) ■	building (13.26%) ■	car (4.08%) ■	truck (0.21%) ■	bicycle (0.02%) ■	m.cycle (0.04%) ■	other-veh. (0.16%) ■	vegetation (26.68%) ■	trunk (0.60%) ■	terrain (7.81%) ■	person (0.18%) ■	bicyclist (1.11e-6%) ■	m.cyclist (5.53e-9%) ■	fence (7.23%) ■	pole (0.28%) ■	traf.-sign (0.06%) ■
100	TangentConv [55]	✗	40.9	83.9	63.9	33.4	15.4	83.4	90.8	15.2	2.7	16.5	12.1	79.5	49.3	58.1	23.0	28.4	8.1	49.0	35.8	28.5
	RandLA-Net [29]		55.9	90.5	74.0	61.8	24.5	89.7	94.2	43.9	47.4	32.2	39.1	83.8	63.6	68.6	48.4	47.4	9.4	60.4	51.0	50.7
	SPVNAS [64]		67.0	90.2	75.4	67.6	21.8	91.6	97.2	56.6	50.6	50.4	58.0	86.1	73.4	71.0	67.4	71.0	50.3	66.9	64.3	67.3
	(AF)^{2}S3Net [14]		69.7	91.3	72.5	68.8	53.5	87.9	94.5	39.2	65.4	86.8	41.1	70.2	68.5	53.7	80.7	74.3	74.3	63.2	61.5	71.0
	DarkNet53Seg [3]	✓	49.9	91.8	74.6	64.8	27.9	84.1	86.4	25.5	24.5	32.7	22.6	78.3	50.1	64.0	36.2	33.6	4.7	55.0	38.9	52.2
	RangeNet53++ [46]		52.2	91.8	75.2	65.0	27.8	87.4	91.4	25.7	25.7	34.4	23.0	80.5	55.1	64.6	38.3	38.8	4.8	58.6	47.9	55.9
	SqueezeSegV3 [83]		55.9	91.7	74.8	63.4	26.4	89.0	92.5	29.6	38.7	36.5	33.0	82.0	58.7	65.4	45.6	46.2	20.1	59.4	49.6	58.9
	SalsaNext [15]		59.5	91.7	75.8	63.7	29.1	90.2	91.9	38.9	48.3	38.6	31.9	81.8	63.6	66.5	60.2	59.0	19.4	64.2	54.3	62.1
0.1	SQN [30]	✗	50.8	**90.5**	**72.9**	56.8	19.1	84.8	**92.1**	36.7	**39.3**	30.1	26.0	**80.8**	59.1	**67.0**	36.4	25.3	7.2	53.3	44.5	44.0
	SalsaNext [15]	✓	50.1	87.2	66.0	51.6	18.4	86.2	88.2	31.6	25.3	29.9	31.8	77.0	59.0	60.1	44.6	44.2	15.6	54.9	39.6	41.4
	Ours		**55.7**	88.4	68.2	**57.6**	**23.4**	**87.7**	89.7	**41.0**	36.2	**36.3**	**38.3**	79.1	**62.3**	60.2	**52.6**	**45.0**	**24.6**	**58.1**	**49.4**	**59.9**
0.01	SQN [30]	✗	39.1	86.6	66.4	43.0	16.9	80.0	85.5	12.9	4.0	1.4	18.4	72.7	49.6	58.8	16.9	22.3	4.3	42.3	31.7	16.6
	SalsaNext [15]	✓	42.6	86.0	65.4	43.7	14.5	85.1	83.1	19.5	21.9	14.9	17.6	73.8	45.9	59.7	29.4	23.6	4.2	50.7	32.6	38.1
	Ours		**46.2**	**88.4**	**68.0**	**52.7**	**17.4**	**87.0**	**87.3**	**28.5**	**25.3**	**16.0**	**23.6**	**79.8**	**55.5**	**62.8**	28.6	**25.3**	1.5	**55.0**	**40.3**	**43.9**

Table 1: On SemKITTI [3] *hidden* test set, our method outperforms all weakly-supervised baselines with either 0.1% or 0.01% labels and *some* fully supervised methods.

Method	Anno.(%)	mIoU%
RandLA-Net [29]	100	53.5
KPConv [66]		55.2
JS3C-Net [86]		60.2
SqueezeSegV2 [75]	100	29.8
SalsaNext [15]		45.0
SalsaNext [15]	0.1	38.9
Ours		**43.0**
SalsaNext [15]	0.01	27.4
Ours		**31.1**

(a) SemPOSS

Method	Anno.(%)	mIoU(%)
RandLA-Net [29]	100	53.5
PolarNet [90]		72.2
Cylinder3D [93]		76.1
RangeNet++ [46]	100	65.5
SalsaNext [15]		72.2
SalsaNext [15]	0.1	56.5
Ours		**58.7**
SalsaNext [15]	0.01	**44.5**
Ours		42.9

(b) nuScenes

Table 2: Performance on SemPOSS [49] (a) and nuScenes [8] (b) validation sets.

4.3 Ablation Study

We first vary the backbone used and then ablate our method. Unless otherwise mentioned, we employ SalsaNet backbone with 0.1% annotation evaluated on SemKITTI val. set.

Choice of Backbones. To demonstrate our approach is architecture-agnostic, Tab. 3 shows experiments on the val. sets of SemKITTI [3] and SemPOSS [49] with various backbones, namely: RangeNet-21 [46], SqueezeSegV3-21 [83], and SalsaNext [15] (details in Appendix B.2). For all backbones and datasets, we boost results significantly w.r.t. to the original segmentation backbone, with mIoU difference of $[+3.8, +6.3]$. The results also advocate for our choice of SalsaNext backbone which performs best over all three.

Architectural Components. Tab. 4a shows that all of our designs contribute to the best results. More in depth, 'w/o contrast module' shows that the contrastive module (Sec. 3.1) contributes to the biggest gain. To evaluate the benefit of our anchor sampling (Sec. 3.3), 'w/o anchor sampling' instead uses all pseudo-labels as anchors showing our strategy is indeed beneficial. In 'w/o prototype' we replace our 20 compact prototypes (Sec. 3.2) with 5,000 pixels as in [70], showing that our method is lighter and more efficient. Finally, Focal loss (Eq. 3) and Lovasz loss (Eq. 4) are better combined as the two pursue slightly

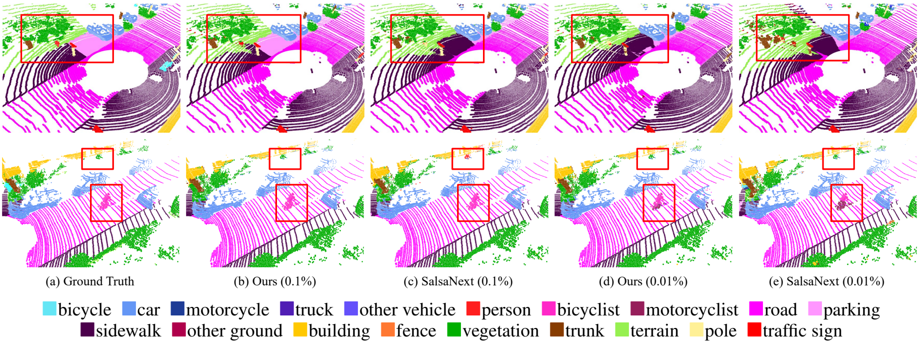

(a) Ground Truth (b) Ours (0.1%) (c) SalsaNext (0.1%) (d) Ours (0.01%) (e) SalsaNext (0.01%)

Figure 2: Sampled predictions on SemKITTI [3] validation set.

Method	SemPOSS [49] mIoU (%)	SemKITTI [3] mIoU (%)
Rangenet-21 [46]	25.1 ↴	40.7 ↴
Ours (Rangenet-21)	**28.9** +3.8	**44.5** +3.8
SqueezeSegV3-21 [83]	30.4 ↴	42.5 ↴
Ours (SqueezeSegV3-21)	**36.7** +6.3	**48.5** +6.0
SalsaNext [15]	38.9 ↴	52.4 ↴
Ours (SalsaNext)	**43.0** +4.1	**57.6** +5.2

Table 3: COARSE3D with different backbones using 0.01% annotation (val. set).

different objectives since Focal loss mitigates imbalance while Lovasz maximizes IoU.

Memory Bank. We now compare other types of memory banks of various per-class sizes. From Tab. 4b, we compare against two variations from [70]: 'Pixel' stores abundant random pixel-wise embeddings and 'Region' which instead stores frame-wise pooled features of semantically consistent regions. Compared to both, our method is more efficient and lighter. The combination of 'Pixel+Region' using 10,000 bank size, proves to be better but still gets outperformed with only 20 of our prototypes. Finally, we also compare against prototypes from mini-batch embeddings [94] – which logically perform worse since they fail at capturing the complete dataset knowledge. In a nutshell, Tab. 4 reveals that more stable contrastive targets – like regions or prototypes – are better and that our momentum update mechanism preserves the dataset context thus reaching better performance.

Number of Prototypes. Tab. 4c studies the impact of varying the number of per class prototypes from 1 to 100. With only 1 prototype, each class uses a single high-dimensional embedding which is also referred to as class-wise memory bank [2]. Altogether, 20 prototypes are optimal among tested choices since fewer or more prototypes degrade performance.

Choice of Anchors. We study the choice of anchors in Tab. 4d, by varying the basic function used for sampling (col 'Basis') and the strategy for sampling segmentation features (col 'Strategy'). Without any basis function, we tried using all pseudo labels predictions 'All' or a 'Random X' number of X pixels. Results show that more support from pseudo labels boosts performance but the strategy is suboptimal given the prediction inaccuracies. To mitigate this, we evaluate the predictions quality with a basis function: either a 'Softmax' as [50] or our 'Entropy' proposal (Sec. 3.3). For both 'Softmax' and 'Entropy' basis, we consider sampling with a finetuned 'hard-threshold', the 'Top-X' elements, or selection from a random 'Probability' sampling. Our 'Entropy' driven probability sampling performs best. We relate these results to the fact that Shannon entropy stores joint information distribution

Method	mIoU (%)
Ours	**57.57**
w/o contrast module	55.44
w/o anchors sampling	56.32
w/o prototype (5k pxl)	56.10
w/o voxel propagation	56.26
w/o Focal loss	42.41
w/o Lovász loss	56.10

(a) Method ablation

Memory bank	Bank size	mIoU (%)
w/o contrast	-	55.44
Pixel	5,000	56.10
Region	5,000	56.66
Pixel + Region	10,000	56.79
Prototype (mini-batch)	20	56.46
Prototype (ours)	20	**57.57**

(b) Memory bank

# Prototype	mIoU (%)
1	55.47
10	56.15
20	**57.57**
50	56.18
100	56.89

(c) Prototypes number

Strategy	Basis	mIoU(%)
All	No	56.32
Random K		56.06
Hard-threshold	Softmax	57.09
Top-K		56.11
Probability		56.93
Hard-threshold	Entropy	56.38
Top-K		56.30
Probability (Ours)		**57.57**

(d) Anchor sampling

Strategy	mIoU(%)
Easy	56.15
Hard	57.00
Semi Hard	56.59
Random	56.41
All (Ours)	**57.57**

(e) Key sampling

Anno.	mIoU (%)	
	SalsaNext [5]	Ours
0.001%	30.39	31.69
0.01%	44.00	47.13
0.1%	52.43	56.61
1%	56.16	58.30
100%	56.44	58.39

(f) Annotation

Table 4: Ablation study results on SemKITTI[3] validation set.

for all class predictions, which makes it more insightful for a proper anchor selection.

Choice of Keys. Prior research [51, 70] show the benefit of using smart key sampling strategies which differs to our use of all keys. In Tab. 4e we compare the common key sampling strategies being 'easy', 'hard', 'semi-hard', 'random' or 'All'. Except for the 'All' strategy, we consider 100 prototypes and use the key sampling mentioned to select 20 of these prototypes as keys. In short, 'easy'/'hard' indicates the complexity for anchors to learn with such keys. We refer to [70] for details. Intuitively, using semi-hard or hard keys encourage the network to learn more robust representations. However, we denote that using all keys performs in fact better with our prototypes. We conjecture that this relates to the prototype capturing a more robust dataset context which is beneficial to preserve.

Effect of Annotation. In Tab. 4f we vary the annotation ratio from 100% (full supervision) to only 0.001% (roughly 1 point label per frame), showing our robustness to the most extreme cases – despite a drastic mIoU drop for 0.001% annotation due to rare classes being absent from labels. In fact, reporting for the latter only mIoU for 'classes in labels', SalsaNext and Ours get respectively, 45.50% and 47.61% which again demonstrates robustness. We further study the effect of random label in Appendix A, showing in a nutshell that variances of 0.1%/0.01% settings are ±0.93/±0.32 – far smaller than the baseline gaps (+4.18/+3.13).

5 Conclusion

In this paper, we present a weakly supervised LiDAR point cloud semantic segmentation framework. Specifically, we develop a compact class-prototype contrastive learning scheme based on online clustered embeddings and use this prototype as the key to pulling entropy-driven sampled anchors. Extensive experiments on three projection-based backbones and three real-world datasets demonstrate the effectiveness of our method.

Acknowledgement. Rong Li was supported by the SMIL lab of South China University of Technolog, received support and advices from Prof. Mingkui Tan, Prof. Caixia Li and Zhuangwei Zhuang. Rong Li was also partly supported by Key-Area Research and Development Program Guangdong Province (2019B010155001). Inria members were partly funded by French project SIGHT (ANR-20-CE23-0016). This work was performed using HPC resources from GENCI–IDRIS (Grant 2021-AD011012808 and 2022-AD011012808R1).

We study the effect of annotation sampling in Appendix A and report implementation details with new ablation in Appendix B and more qualitative results in Appendix C.

A Effect of Annotations

In weak supervision settings, changing the set of labelled points impact performance. We now study more in depth the effect of sampled labels by randomly resampling labels 3 times on 2 datasets. As it appears in Tab. 5, our method is relatively stable (*i.e.*, std<1). We also measure the gap between random annotations and those of human by asking 2 operators to annotate the entire SemPOSS, labeling roughly 0.01% points per frame. Again, from Tab. 5 our 'human' labels are within 3 std of the mean 0.01% performance (*i.e.*, 29.27 vs 31.48±0.43).

	SemKITTI [3]		**SemPOSS [49]**		
Anno.	0.10%	0.01%	0.10%	0.01%	human (≈ 0.01%)
run 1	57.57	47.35	43.00	31.10	29.27
run 2	56.54	47.28	42.88	31.95	-
run 3	55.71	46.76	42.47	31.38	-
all	56.61±0.93	47.13±0.32	42.78±0.28	31.48±0.43	-

Table 5: Effect of annotation sampling on SemKITTI and SemPOSS.

B Implementation Details

B.1 Label Voxel Propagation

We replicate SQN [30] and apply their random grid downsampling with 0.06 voxel size. Our trivial scheme simply propagates existing labels to all points within the same voxel – thus densifying the labels at no extra labelling cost. In the extremely rare case of conflicting labels within a single voxel (e.g. a voxel having two labelled points with different classes), the voxel label will be randomly assigned.

We evaluate the effect of this voxel propagation scheme using the SalsaNext backbone on SemKITTI val. set in the 0.1% annotation setting. With/without our scheme we get **57.57**/56.26 mIoU. Since baselines do not use our label propagation, it is important to note that the mIoU gap obtained (+1.31 mIoU) is smaller than the gap with the original SalsaNext (+5.14, cf. main paper Tab. 3f) in the same 0.1% setting. This advocates that our method only *partly* benefits from our voxel propagation scheme and performs best thanks to our overall contrastive learning strategy.

B.2 Segmentation Backbones

SalsaNext [15]. We use the official implementation[2] for the SemKITTI dataset[3] and applied our best effort to fairly re-implement it for nuScenes [8] and SemPOSS [49]. Specific to SemPOSS[49], we used an input padding to get a compatible size. When trained with our method, we finetune our contrastive learning hyperparameters.

[2] https://github.com/TiagoCortinhal/SalsaNext

SqueezeSegV3 [83]. We use the lighter *SqueezeSegV3-21* from the official implementation[3]. To boost performance on weakly supervised tasks, we replaced the multi-layer cross-entropy loss – improper for sparse weak labels due to its downsampling –, with normal cross-entropy loss. When trained with our method, we simply use the contrastive learning hyperparameter found with SalsaNext.

RangeNet++ [46]. We use the ligher *RangeNet-21* from the official implementation[4]. Inputs are pad to get compatible size. When trained with our method, we simply use the contrastive learning hyperparameter found with SalsaNext.

C Additional Results

We report additional qualitative results using SalsaNext on SemKITTI, SemPOSS, and nuScenes in Figs. 3, 4 and 5, respectively, for both 0.1% and 0.01% settings. Overall, our method surpasses SalsaNext, especially in ambiguous and cluttered regions, illustrated in Fig. 3 (sidewalk/parking - row 1; vegetation/building - row 3), in Fig. 4 (car - row 1; rider - row 3; pole/plants - row 4), and in Fig. 5 (terrain/other flat - row 1). Also, SalsaNext makes more mistake regarding classes with close semantical meaning as in Fig. 3 (truck/car - row 1), in Fig. 4 (building/fence - row 5), and in Fig. 5 (bus/trailer/truck - row 2, 3, 4, 5). Furthermore, our method shows superiority in predicting small objects with similar structure or spatial position, demonstrated in Fig. 3 (fence/vegetation - row 4, 5; trunk/traffic sign - row 5), in Fig. 4 (rider/people - row 2, 3, 5; pole/plants - row 4), and in Fig. 5 (terrain/other flat - row 1). Additionally, our method infers better far away, low density regions e.g. Fig. 4 (car - row 1), and Fig. 5 (vegetation/barrier - row 2).

[3] https://github.com/chenfengxu714/SqueezeSegV3

[4] https://github.com/PRBonn/lidar-bonnetal

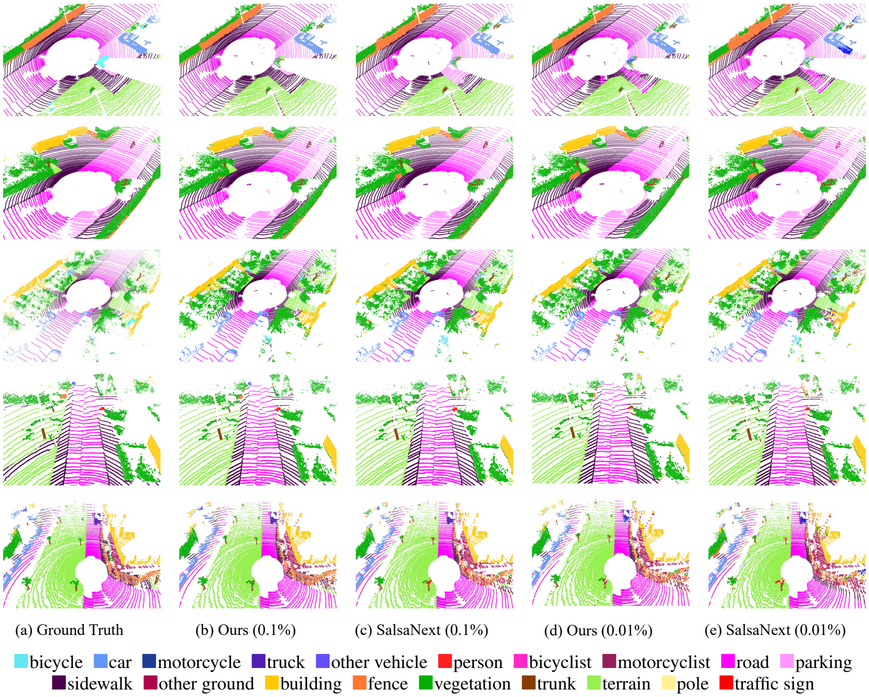


Figure 3: Additional qualitative results on SemKITTI [3]

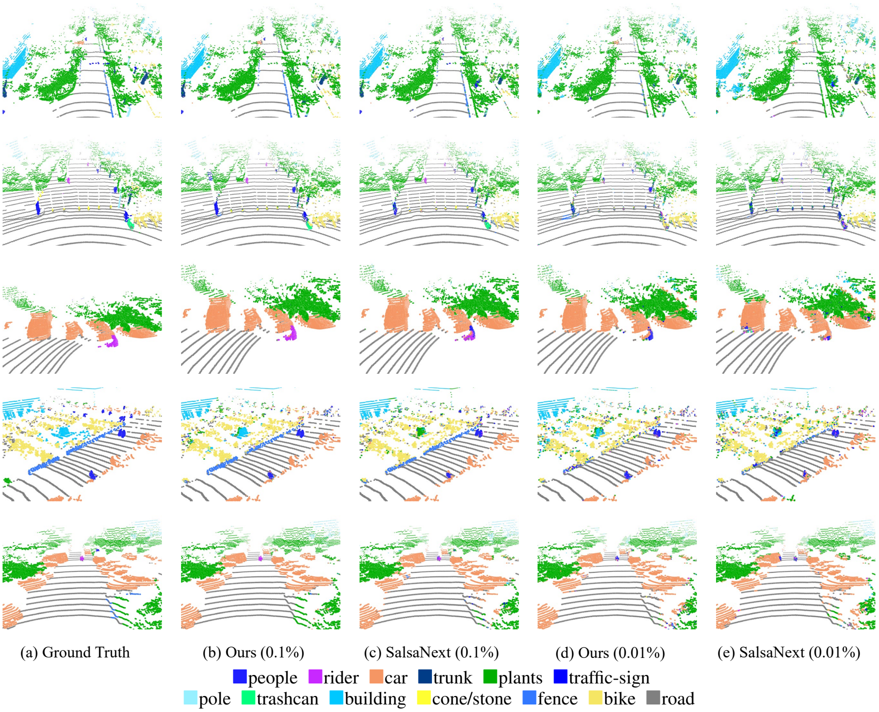
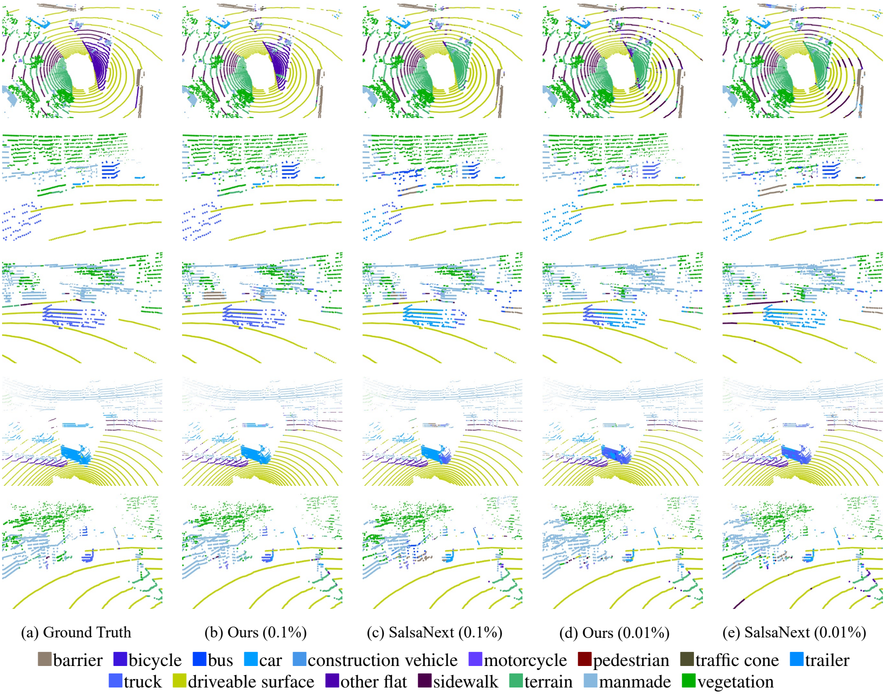


Figure 4: Qualitative results on SemPOSS [49]



(a) Ground Truth (b) Ours (0.1%) (c) SalsaNext (0.1%) (d) Ours (0.01%) (e) SalsaNext (0.01%)

■barrier ■bicycle ■bus ■car ■construction vehicle ■motorcycle ■pedestrian ■traffic cone ■trailer
■truck ■driveable surface ■other flat ■sidewalk ■terrain ■manmade ■vegetation

Figure 5: Qualitative results on nuScenes [8]

References

[1] Eren Erdal Aksoy, Saimir Baci, and Selcuk Cavdar. Salsanet: Fast road and vehicle segmentation in lidar point clouds for autonomous driving. In *IV*, 2020.

[2] Inigo Alonso, Alberto Sabater, David Ferstl, Luis Montesano, and Ana C Murillo. Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank. In *ICCV*, 2021.

[3] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, C. Stachniss, and Juergen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, 2019.

[4] Sean Bell and Kavita Bala. Learning visual similarity for product design with convolutional neural networks. In *TOG*, 2015.

[5] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *CVPR*, 2018.

[6] Maxime Bucher, Stéphane Herbin, and Frédéric Jurie. Hard negative mining for metric learning based zero-shot classification. In *ECCV*, 2016.

[7] Maxime Bucher, Stéphane Herbin, and Frédéric Jurie. Improving semantic embedding consistency by metric learning for zero-shot classiffication. In *ECCV*, 2016.

[8] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[9] Tiffany Tianhui Cai, Jonathan Frankle, David J Schwab, and Ari S Morcos. Are all negatives created equal in contrastive instance discrimination? In *arXiv*, 2020.

[10] Krishna Chaitanya, Ertunc Erdil, Neerav Karani, and Ender Konukoglu. Contrastive learning of global and local features for medical image segmentation with limited annotations. In *NeuRIPS*, 2020.

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[12] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. In *arXiv*, 2020.

[13] Yujin Chen, Matthias Nießner, and Angela Dai. 4dcontrast: Contrastive learning with dynamic correspondences for 3d scene understanding. In *ECCV*, 2022.

[14] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. 2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In *CVPR*, 2021.

[15] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In *ISVC*, 2020.

[16] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. In *TIT*, 1967.

[17] Yin Cui, Feng Zhou, Yuanqing Lin, and Serge Belongie. Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. In *CVPR*, 2016.

[18] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeuRIPS*, 2013.

[19] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017.

[20] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, 2018.

[21] Salvador Garcia, Joaquin Derrac, Jose Cano, and Francisco Herrera. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. In *TPAMI*, 2012.

[22] Jacob Goldberger, Geoffrey E Hinton, Sam Roweis, and Russ R Salakhutdinov. Neighbourhood components analysis. In *NeurIPS*, 2004.

[23] Samantha Guerriero, Barbara Caputo, and Thomas Mensink. Deepncm: Deep nearest class mean classifiers. In *ICLR Workshops*, 2018.

[24] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010.

[25] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

[26] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[27] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *CVPR*, 2021.

[28] Hanzhe Hu, Jinshi Cui, and Liwei Wang. Region-aware contrastive learning for semantic segmentation. In *ICCV*, 2021.

[29] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Agathoniki Trigoni, and A. Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *CVPR*, 2020.

[30] Qingyong Hu, Bo Yang, Guangchi Fang, Yulan Guo, Ales Leonardis, Niki Trigoni, and Andrew Markham. Sqn: Weakly-supervised semantic segmentation of large-scale 3d point clouds with 1000x fewer labels. In *ECCV*, 2022.

[31] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *ICCV*, 2021.

[32] Saumya Jetley, Bernardino Romera-Paredes, Sadeep Jayasumana, and Philip Torr. Prototypical priors: From improving classification to zero-shot learning. In *BMVC*, 2015.

[33] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. In *NeurIPS*, 2020.

[34] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. In *Symmetry*, 2019.

[35] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020.

[36] Shamit Lal, Mihir Prabhudesai, Ishita Mediratta, Adam W Harley, and Katerina Fragkiadaki. Coconets: Continuous contrastive 3d scene representations. In *CVPR*, 2021.

[37] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, 2018.

[38] Mengtian Li, Yuan Xie, Yunhang Shen, Bo Ke, Ruizhi Qiao, Bo Ren, Shaohui Lin, and Lizhuang Ma. Hybridcr: Weakly-supervised 3d point cloud semantic segmentation via hybrid contrastive regularization. In *CVPR*, 2022.

[39] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *AAAI*, 2018.

[40] Hanxue Liang, Chenhan Jiang, Dapeng Feng, Xin Chen, Hang Xu, Xiaodan Liang, Wei Zhang, Zhenguo Li, and Luc Van Gool. Exploring geometry-aware contrast and clustering harmonization for self-supervised 3d object detection. In *ICCV*, 2021.

[41] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. In *arXiv*, 2021.

[42] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *TPAMI*, 2020.

[43] Yunze Liu, Li Yi, Shanghang Zhang, Qingnan Fan, Thomas Funkhouser, and Hao Dong. P4contrast: Contrastive learning with pairs of point-pixel pairs for rgb-d scene understanding. In *arXiv*, 2020.

[44] Zhengzhe Liu, Xiaojuan Qi, and Chi-Wing Fu. One thing one click: A self-training approach for weakly supervised 3d semantic segmentation. In *CVPR*, 2021.

[45] Pascal Mettes, Elise van der Pol, and Cees Snoek. Hyperspherical prototype networks. In *NeurIPS*, 2019.

[46] Andres Milioto, Ignacio Vizzo, Jens Behley, and C. Stachniss. Rangenet ++: Fast and accurate lidar semantic segmentation. In *IROS*, 2019.

[47] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *ICCV*, 2017.

[48] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. In *arXiv*, 2018.

[49] Yancheng Pan, Biao Gao, Jilin Mei, Sibo Geng, Chengkun Li, and Huijing Zhao. Semanticposs: A point cloud dataset with large quantity of dynamic instances. In *IV*, 2020.

[50] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh Singh Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *ArXiv*, 2021.

[51] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *ICLR*, 2020.

[52] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *ICLR*, 2021.

[53] Radu Alexandru Rosu, Peer Schutt, Jan Quenzel, and Sven Behnke. Latticenet: Fast point cloud segmentation using permutohedral lattices. In *RSS*, 2020.

[54] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *ECCV*, 2022.

[55] Ruslan Salakhutdinov and Geoff Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In *AISTATS*, 2007.

[56] Corentin Sautier, Gilles Puy, Spyros Gidaris, Alexandre Boulch, Andrei Bursuc, and Renaud Marlet. Image-to-lidar self-supervised distillation for autonomous driving data. In *CVPR*, 2022.

[57] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[58] Claude Elwood Shannon. A mathematical theory of communication. *SIGMOBILE*, 2001.

[59] Hanyu Shi, Jiacheng Wei, Ruibo Li, Fayao Liu, and Guosheng Lin. Weakly supervised segmentation on outdoor 4d point clouds with temporal matching and spatial graph propagation. In *CVPR*, 2022.

[60] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *ICCV*, 2015.

[61] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.

[62] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*, 2016.

[63] Hang Su, V. Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *CVPR*, 2018.

[64] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *ECCV*, 2020.

[65] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *CVPR*, 2018.

[66] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, 2019.

[67] Ozan Unal, Dengxin Dai, and Luc Van Gool. Scribble-supervised lidar semantic segmentation. In *CVPR*, 2022.

[68] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, 2016.

[69] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *ICCV*, 2019.

[70] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *ICCV*, 2021.

[71] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, 2021.

[72] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R Scott. Cross-batch memory for embedding learning. In *CVPR*, 2020.

[73] Jiacheng Wei, Guosheng Lin, Kim-Hui Yap, Tzu-Yi Hung, and Lihua Xie. Multi-path region mining for weakly supervised 3d semantic segmentation on point clouds. In *CVPR*, 2020.

[74] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *ICRA*, 2018.

[75] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *ICRA*, 2019.

[76] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *ICCV*, 2017.

[77] Tsung-Han Wu, Yueh-Cheng Liu, Yu-Kai Huang, Hsin-Ying Lee, Hung-Ting Su, Ping-Chia Huang, and Winston H. Hsu. Redal: Region-based and diversity-aware active learning for point cloud semantic segmentation. In *ICCV*, 2021.

[78] Zhirong Wu, Alexei A. Efros, and Stella X. Yu. Improving generalization via scalable neighborhood component analysis. In *ECCV*, 2018.

[79] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

[80] Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Delving into inter-image invariance for unsupervised visual representations. In *arXiv*, 2020.

[81] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *ECCV*, 2020.

[82] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *CVPR*, 2021.

[83] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Péter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In *ECCV*, 2020.

[84] Wenjia Xu, Yongqin Xian, Jiuniu Wang, Bernt Schiele, and Zeynep Akata. Attribute prototype network for zero-shot learning. In *NeurIPS*, 2020.

[85] Xun Xu and Gim Hee Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *CVPR*, 2020.

[86] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In *AAAI*, 2021.

[87] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *CVPR*, 2018.

[88] Yachao Zhang, Zonghao Li, Yuan Xie, Yanyun Qu, Cuihua Li, and Tao Mei. Weakly supervised semantic segmentation for large-scale point cloud. In *AAAI*, 2021.

[89] Yachao Zhang, Yanyun Qu, Yuan Xie, Zonghao Li, Shanshan Zheng, and Cuihua Li. Perturbed self-distillation: Weakly supervised large-scale point cloud semantic segmentation. In *ICCV*, 2021.

[90] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *CVPR*, 2020.

[91] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *ICCV*, 2021.

[92] Tianfei Zhou, Wenguan Wang, Ender Konukoglu, and Luc Van Gool. Rethinking semantic segmentation: A prototype view. In *CVPR*, 2022.

[93] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *CVPR*, 2021.

[94] Chengxu Zhuang, Alex Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *ICCV*, 2019.