Distribution Guidance Network for Weakly Supervised Point Cloud Semantic Segmentation

Zhiyi Pan
SECE, Peking University
Peng Cheng Laboratory
panzhiyi@stu.pku.edu.cn

Wei Gao
SECE, Peking University
gaowei262@pku.edu.cn

Shan Liu
Media Laboratory, Tencent
shanl@tencent.com

Ge Li*
SECE, Peking University
geli@ece.pku.edu.cn

Abstract

Despite alleviating the dependence on dense annotations inherent to fully supervised methods, weakly supervised point cloud semantic segmentation suffers from inadequate supervision signals. In response to this challenge, we introduce a novel perspective that imparts auxiliary constraints by regulating the feature space under weak supervision. Our initial investigation identifies which distributions accurately characterize the feature space, subsequently leveraging this priori to guide the alignment of the weakly supervised embeddings. Specifically, we analyze the superiority of the mixture of von Mises-Fisher distributions (moVMF) among several common distribution candidates. Accordingly, we develop a **D**istribution **G**uidance **Net**work (DGNet), which comprises a weakly supervised learning branch and a distribution alignment branch. Leveraging reliable clustering initialization derived from the weakly supervised learning branch, the distribution alignment branch alternately updates the parameters of the moVMF and the network, ensuring alignment with the moVMF-defined latent space. Extensive experiments validate the rationality and effectiveness of our distribution choice and network design. Consequently, DGNet achieves state-of-the-art performance under multiple datasets and various weakly supervised settings.

1 Introduction

As a fundamental task in 3D scene understanding, point cloud semantic segmentation [42, 32, 29] is widely entrenched in 3D applications, such as 3D reconstruction [15, 33], autonomous driving [20], and embodied intelligence [14, 50]. Despite significant accomplishments in tackling the disorder and disorganization, point cloud semantic segmentation remains annotation-intensive, hindering its expansion in big datasets and large models. For this reason, the academic community explores achieving point cloud semantic segmentation in a weakly supervised manner. However, due to the lack of supervision signals, learning point cloud segmentation on sparse annotations is nontrivial.

In recent years, considerable effort has been made to pursue additional constraints in weak supervision. As shown in Fig. 1, representative work can be broadly categorized into several paradigms: 1) *Contrastive Learning / Perturbation Consistency* imposes contrastive loss or consistency constraint between network embeddings of original and perturbed point clouds, respectively. 2) *Self-training* progressively enhances segmentation quality by treating reliable predictions as pseudo-labels, with

*Ge Li is the corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

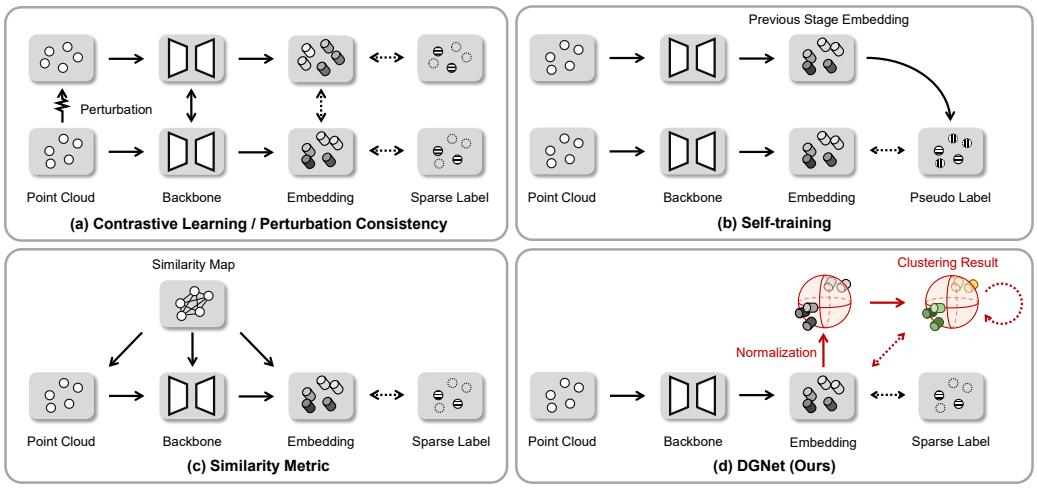

Figure 1: Visual comparisons of mainstream weakly supervised point cloud semantic segmentation paradigms and our DGNet. The solid and dashed lines represent the network forward process and the loss function, respectively.

given sparse annotations as initial labels. 3) *Similarity Metric* transfers supervision signals from annotated points to unlabeled regions via leveraging the low-level features or the embedding similarity. Nevertheless, most constraints stem from heuristic assumptions and ignore the inherent distribution of network embedding, resulting in ambiguous interpretations of point-level predictions. In contrast to existing paradigms, in this paper, we re-examine two fundamental issues: *How to characterize the semantic segmentation feature space for weak supervision and how to intensify this intrinsic distribution under weakly supervised learning?*

For the first issue about `oughtness`, we expect to provide a mathematically describable distribution for weakly supervised features. Consequently, we attempt to precisely describe the feature space in terms of two dimensions: distance metric and distribution modeling. In distance metric, we compare the Euclidean norm and cosine similarity between representations, and in distribution modeling, the category prototype model and the mixture model are considered. Among our candidate combinations, the mixture of von Mises-Fisher distributions (moVMF) with cosine similarity is finalized, due to its powerful fitting capability to segment head and insensitivity to the Curse of Dimensionality [44]. We believe that a superior weakly supervised feature space should adhere to this distribution.

For the second issue about `practice`, we dynamically align the embedding distribution in the hidden space to moVMF during weakly supervised learning. Accordingly, we propose a **D**istribution **G**uidance **Net**work (DGNet), comprising a weakly supervised learning branch and a distribution alignment branch. Specifically, the weakly supervised learning branch learns semantic embeddings under sparse annotations, while the distribution alignment branch constrains the distribution of the network embeddings. Via a Nested Expectation-Maximum Algorithm, the semantic features are dynamically refined. Therefore, restricting and fitting is a mutually reinforcing, iterative optimization process. To curtail the pattern of feature distribution, we derive the vMF loss based on the maximum likelihood estimation and the discriminative loss inspired by metric learning [22]. For joint optimization, consistency loss is imposed between the segmentation predictions and the posterior probabilities. During the inference phase, only the weakly supervised learning branch is activated to maintain inference consistency with fully supervised learning.

We validate DGNet on three prevailing point cloud datasets, *i.e.*, S3DIS [1], ScanNetV2 [11], and SemanticKITTI [5]. After the constraints of feature distribution, DGNet provides significant performance improvements over multiple baselines. Across various label rates, our method achieves state-of-the-art weakly supervised semantic segmentation performance. Extensive ablation studies also confirm the effectiveness of each loss term we proposed. In addition, posterior probabilities under the moVMF provide a plausible interpretation for predictions on unlabeled points.

2 Related Work

Weakly Supervised Point Cloud Semantic Segmentation. Weakly supervised point cloud semantic segmentation methods aim to provide reliable additional supervision with sparse annotations. Four paradigms have been successively proposed in recent years, *i.e.*, perturbation consistency, contrastive learning, self-training, and similarity metric. Perturbation consistency methods are based on the assumption of perturbation invariance of network features, imposing diverse perturbations (such affine transforms with point jitter [59, 54], downsampling [56], masking [35], etc.) to construct pairs of point clouds. Several methods [31, 34] introduce contrastive learning in weak supervision to encourage the discriminability of hidden layer features. Additionally, pre-training methods [55, 17] with contrastive learning similarly demonstrate the ability to bias induction in the face of downstream semantic segmentation tasks with sparse annotations. Self-training methods generate reliable dynamic pseudo-labels based on previous stage predictions for subsequent training stages. Via CAMs [62], MPRM [51] and J2D3D [25] dynamically generate point-wise pseudo-labels from subcloud-level annotations and image-level annotations, respectively. Recently, REAL [24] integrate SAM [23] to self-training. Similarity metric methods measure the similarity between labeled and unlabeled points to propagate supervision information, in which the similarity is elaborated on low-level features [49, 52], network embedding [36, 18, 40] or category prototypes [58, 46]. Most similar to our method is the similarity metric strategy. However, the distinction is that DGNet focuses on describing network embeddings holistically, rather than constructing pair relationships between features.

Feature Distribution Constraints. The constraints of feature distribution are always presented in the form of feature clustering. DeepClustering [6], which integrates clustering and unsupervised feature learning, utilizes the clustering results as pseudo-labels to extract visual features dynamically. Following this groundbreaking work, a series of subsequent studies [7, 2, 30] apply feature clustering in unsupervised learning to obtain discriminative visual features for pretraining. For point cloud semantic segmentation, PointDC [10] delineates semantic objects by aligning the features on the same super-voxel in an unsupervised manner. Feng *et al*. [12] imposes a clustering-based representation learning to enhance the discrimination of embeddings under full supervision. In contrast, our DGNet is oriented towards weakly supervised semantic segmentation by restricting the feature distribution.

Mixture of von Mises-Fisher Distributions. The moVMF [3] describes the embeddings of multiple categories on the unit hypersphere in feature space, where the parameters are jointly optimized with the clustering results by Expectation-Maximum algorithm [3, 4]. Some work attempts to combine neural networks with the moVMF in the deep learning era. For example, [16] view face verification as a direct application of clustering, introducing a vMF loss to align the distribution of face features. Segsort [21], on the other hand, utilizes the prior of the moVMF to over-segment images. DINO-VMF [13] achieve a more stable pre-trained method by precisely describing DINO [8] as a moVMF. In this work, we discuss the superiority of moVMF in characterizing semantic embeddings and trust it as a priori to guide weakly supervised learning.

3 Methodology

3.1 Preliminaries

Task Definition. Without loss of generality, a point cloud for weakly supervised learning is denoted as $\{(\mathbf{X}_l, \mathbf{Y}), (\mathbf{X}_u, \varnothing)\} = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_m, y_m), \mathbf{x}_{m+1}, \cdots, \mathbf{x}_n\}$, where $\mathbf{X}_l$ and $\mathbf{X}_u$ are the point sets with and without annotations, respectively. $\mathbf{Y}$ is the corresponding annotations on $\mathbf{X}_l$, in which $y_i \in \mathbb{C}$ and $\mathbb{C}$ is the set of category indices. n and m are the point numbers of the point cloud and labeled set, respectively. Fed into the segment head, the network embedding $\mathbf{f}_i$ is projected into the category probability vector $\mathbf{p}_i$. The partial cross-entropy loss is employed in conventional weakly supervised semantic segmentation:

$$\mathcal{L}_{\mathrm{pCE}} = -\frac{1}{m}\sum_{i=1}^{m}\log(\mathbf{p}_i^{y_i}), \tag{1}$$

where $\mathbf{p}_i^{y_i}$ represents the probability of y_i-th category in $\mathbf{p}_i$.

von Mises-Fisher Distribution (vMF). The vMF has demonstrated strong data fitting and generalization capabilities in the fields of self-supervised learning [9, 13], classification [44], variational inference [47], and online continual learning [38]. This distribution describes the distribution of normalized embedding $\mathbf{v}_i = \mathrm{norm}(\mathbf{f}_i)$ on the unit hypersphere, with the probability density function:

$$f(\mathbf{v}_i|\mathbf{u}, \kappa) = C_d(\kappa) \exp(\kappa \mathbf{u}^\top \mathbf{v}_i), \tag{2}$$

where $\mathbf{u}$ represents the mean vector of vMF and $\kappa \geq 0$ is a concentration parameter that controls the probability concentration around μ. $C_d(\kappa)$ is the normalization constant.

Mixture of vMF (moVMF). Similar to other mixture models, moVMF treats vMF as a sub-distribution to describe the overall distribution of multiple categories. Over the entire set of categories $\mathbb{C}$, the probability density function of moVMF is formulated as:

$$P(\mathbf{v}_i|\mathbb{C}, \Theta) = \sum_{c \in \mathbb{C}} \alpha_c f(\mathbf{v}_i|\mathbf{u}_c, \kappa_c) = \sum_{c \in \mathbb{C}} \alpha_c C_d(\kappa_c) \exp(\kappa_c \mathbf{u}_c^\top \mathbf{v}_i), \tag{3}$$

where $\Theta = \{\alpha_c, \kappa_c, \mathbf{u}_c | c \in \mathbb{C}\}$ is the parameters of moVMF. α_c denotes the proportion of the von Mises-Fisher distribution for the c-th category and $\sum \alpha_c = 1$.

3.2 Feature Space Description

We intend to provide additional supervision signals for weakly supervised learning by portraying and enhancing its inherent distribution. Specifically, We explore it in two dimensions, *i.e.*, the distance metric and the distribution modeling:

- **Distance metric.** Distance metric influences the similarity relationship between features. We consider the two most commonly used distance measures in clustering, *i.e.*, Euclidean norm and cosine similarity. For given vectors $\mathbf{u}$ and $\mathbf{v}$, the Euclidean norm is defined as $\|\mathbf{u} - \mathbf{v}\|_2$ and the cosine similarity is defined as $\frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2}$. Cosine similarity can be viewed as the inner product of normalized $\mathbf{u}$ and $\mathbf{v}$.
- **Distribution modeling.** Distribution modeling determines the clustering results of features. A straightforward model is the Category Prototype [45]. In this model, clusters are assigned by comparing the distance between the features and each category prototype. In addition, we incorporate mixture models into the comparison. Depending on the distance measure, we categorize it into the Gaussian Mixture Model (GMM) with Euclidean norm and the mixture of von Mises-Fisher distributions (moVMF) with cosine similarity, respectively.

Describing the feature space of a neural network remains an open problem. Various factors influence the feature space, including network architecture, training data, parameter configurations, and the optimization (loss) function. Given this intractability, deriving a universally optimal mathematical description is impractical. Therefore, we discuss the merits and demerits of these candidate distributions from the following perspectives:

- **Segment head.** To facilitate the analysis, we simplify the structure of the segment head as $\mathrm{SegHead}(\mathbf{f}) = \mathrm{argmax}(\mathrm{softmax}(\mathbf{w}\mathbf{f}^\top))$, where $\mathbf{f}$ is the semantic feature extracted by the decoder and $\mathbf{w}$ is the parameter of the output layer. Consider a group of feature vectors $\{k\mathbf{f} | k \geq 0 \ \& \ \mathbf{f} \neq \mathbf{0}\}$. For any two feature vectors $k_1\mathbf{f}$ and $k_2\mathbf{f}$ within this group, the segmentation predictions are identical, *i.e.*, $\mathrm{SegHead}(k_1\mathbf{f}) = \mathrm{argmax}(\mathrm{softmax}(k_1\mathbf{w}\mathbf{f}^\top)) = \mathrm{argmax}(\mathrm{softmax}(k_2\mathbf{w}\mathbf{f}^\top)) = \mathrm{SegHead}(k_2\mathbf{f})$. If the general case of using an activation function is taken into account, it does not change the result after argmax since the activation function is usually monotonically nondecreasing. Therefore, the segment head is a radial classifier with a more pronounced classification performance on the angles, so cosine similarity describes the feature space better than the Euclidean norm.
- **Curse of dimensionality.** Another advantage of cosine similarity can be explained in terms of the Curse of Dimensionality. Most high-dimensional features are far from each other, causing the Euclidean distance to become ineffective in distinguishing differences between feature vectors. Cosine similarity, on the other hand, is more effective in distinguishing differences between features by measuring the angle between the vectors. Besides, the Euclidean norm is sensitive to scale while cosine similarity is not affected by the length of the vectors.

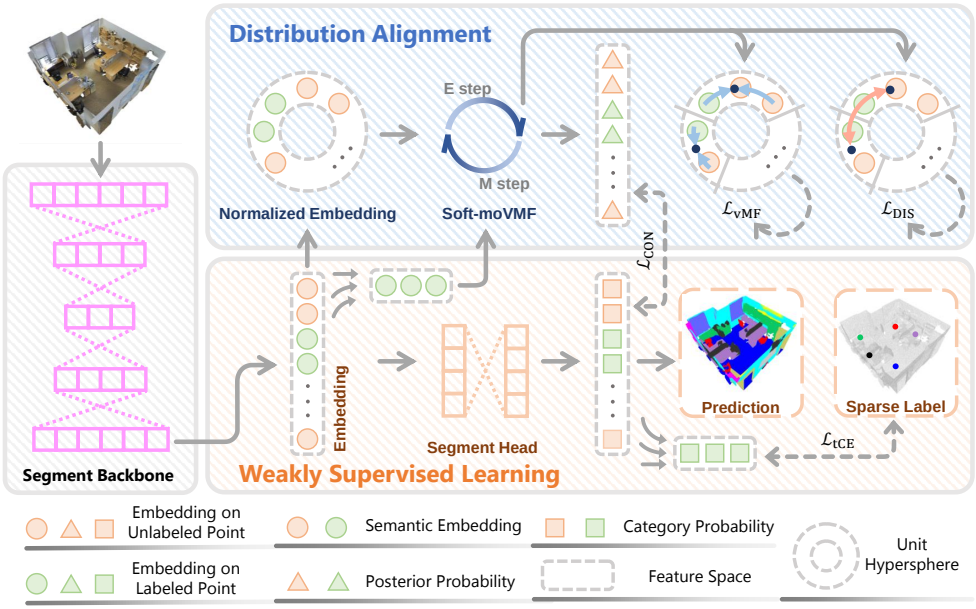

Figure 2: Structure of Distribution Guidance Network.

- **Fitting ability.** Despite its computational simplicity, the Category Prototype Model clusters the features by comparing the distance between features and category prototypes. This means that the Category Prototype Model ignores the distribution within categories and the variability between categories. In contrast, the Mixture Model possesses intra-category fitting and inter-category perception capabilities.

Based on the above analysis and experimental validation in Sec. 4.3, we characterize the feature space as moVMF and propose the Distribution Guidance Network to enhance this distribution.

3.3 Distribution Guidance Network

To enhance the intrinsic distributions discussed in Section 3.2, we propose a **D**istribution **G**uidance **Net**work (DGNet). The structure of DGNet is shown in Fig. 2, which comprises the weakly supervised learning branch and the distribution alignment branch.

3.3.1 Weakly Supervised Learning Branch

Sparse annotations pose two challenges for the point cloud semantic segmentation. The first is underfitting the entire dataset, as the supervision signals are insufficient for complex structured point clouds. To address underfitting, we introduce additional signals by reinforcing the inherent distribution of the feature space. The second challenge is overfitting within the labeled set $\mathbf{X}_l$, as the model capacity is more than adequate to fit the labeled points. To mitigate overfitting, we replace the conventional partial cross-entropy loss Eq. 1 with the truncated cross-entropy loss [60] in the weakly supervised learning branch, which is defined as:

$$\mathcal{L}_{\text{tCE}} = -\frac{1}{m}\sum_{i=1}^{m} \min\Big(\log(\mathbf{p}_i^{y_i}), \log(\beta)\Big), \tag{4}$$

where $\beta \in [0, 1]$ represents the threshold for truncating the cross-entropy loss. With $\mathcal{L}_{\text{tCE}}$, the gradient of the cross-entropy loss is curtailed when the predicted class probability for an annotated point exceeds β. Over-optimization for that point is halted, thereby preventing overfitting.

The weakly supervised learning branch also provides robust initialization for the distribution alignment branch by utilizing the average feature vector of the labeled points. According to the Central Limit Theorem [27], the difference between the initialization vector from the weakly supervised

learning branch and the theoretical optimal average vector conforms to a Gaussian distribution with a mean of 0. The initialized mean vector in DGNet has a high probability of appearing in the vicinity of the optimal solution, which facilitates the clustering algorithm in achieving rapid and stable convergence.

3.3.2 Distribution Alignment Branch

The feature space descriptor moVMF, investigated in Sec. 3.2, is employed to regulate the feature space under weakly supervised learning. Initially, the network embeddings are normalized and projected onto the unit hyperspherical surface of the feature space, *i.e.*, $\mathbf{v}_i = \text{norm}(\mathbf{f}_i)$. Following this, the optimization objective function is defined utilizing maximum likelihood estimation[1]:

$$\max_{\phi,\mathcal{Z},\Theta} P(\mathcal{V}|\mathcal{Z},\Theta) = \min_{\phi,\mathcal{Z},\Theta} -\sum_{i=1}^{n}\Big[\log(\alpha_{z_i}) + \kappa\mathbf{u}_{z_i}^{\top}\mathbf{v}_i\Big], \tag{5}$$

where ϕ, $\mathcal{V} = \{\mathbf{v}_i\}$, $\mathcal{Z} = \{z_i\}$ and $\Theta = \{\alpha_c, \kappa, \mathbf{u}_c\}$ denote the learnable network parameters, the normalized network embeddings, the corresponding clustering results and the parameters of moVMF, respectively. To avoid the long-tail problem and simplify computations, the concentration parameter κ is fixed as a constant in our implementation. Since the clustering initialization is category-aware, the clustering results $z_i \in \mathbb{C}$ are with category labels. While the primary optimization goal is the learning of network parameters ϕ, we dynamically resolve $\mathcal{Z}$ and Θ to furnish a more precise feature space description. Consequently, we develop a Nested Expectation-Maximum Algorithm to manage the challenge associated with the three optimization variables delineated in Eq. 5.

- **E Step (Optimize Θ and $\mathcal{Z}$):** Regarding network embeddings as input conditions, we integrate the soft-moVMF EM algorithm [3] into the network to alternately optimize Θ and $\mathcal{Z}$. To enhance the stability and computational efficiency of the algorithm, we utilize the average features of labeled points from the weakly supervised branch to initialize $\mathbf{u}$. The posterior probability set $\mathcal{Q} = \{\mathbf{q}_i\}$ serves as the soft assignment for the clustering results $\mathcal{Z}$, where $\mathbf{q}_i$ is defined as

$$\mathbf{q}_i = P(c|\mathbf{v}_i,\Theta) = \frac{\alpha_c \exp(\kappa\mathbf{u}_c^{\top}\mathbf{v}_i)}{\sum_{l\in\mathbb{C}} \alpha_l \exp(\kappa\mathbf{u}_l^{\top}\mathbf{v}_i)}. \tag{6}$$

  Compared to other algorithms in [3], the soft-moVMF algorithm updates Θ by weighting all features according to their posterior probabilities, considering inter-cluster similarities, thereby achieving more accurate parameter updates. The posterior probability $\mathbf{q}_i \in [0,1]^{|\mathbb{C}|\times 1}$ is employed not only as a weighting factor for updates but also in the calculation of the loss function for joint optimization. Furthermore, $\mathcal{Q}$ provides a probabilistic explanation for the predictions during the inference phase.

  The complexity of the soft-voVMF is $O(tn|\mathbb{C}|)$, where t is the iteration number, n is the point number of the point cloud, and $|\mathbb{C}|$ is the number of semantic categories. Since t, n, and $|\mathbb{C}|$ are all set to constant values during network training, the extra computation introduced by the distribution alignment branch is trivial.

- **M Step (Optimize ϕ):** With the converged parameters Θ and $\mathcal{Z}$ fixed, we optimize ϕ by the backpropagation process. Consistent with the philosophy of the soft-moVMF, we incorporate the posterior probability $\mathcal{Q}$ into Eq. 5, and reformulate it into the loss function as follows:

$$\mathcal{L}_{\text{vMF}} = -\sum_{i=1}^{n}\sum_{c\in\mathbb{C}} \mathbf{q}_i^c\Big[\log(\alpha_c) + \kappa\mathbf{u}_c^{\top}\mathbf{v}_i\Big]. \tag{7}$$

  Additionally, acknowledging the significance of distinct decision boundaries within the mixture model, we incorporate a discriminative loss derived from metric learning [22] which is defined as:

$$\mathcal{L}_{\text{DIS}} = \frac{1}{|\mathbb{C}|(|\mathbb{C}|-1)} \sum_{c_1,c_2\in\mathbb{C}\&c_1\neq c_2} \mathbf{u}_{c_1}^{\top}\mathbf{u}_{c_2}. \tag{8}$$

The interpretation of Bayesian posterior probabilities for predictions based on the moVMF is an attractive property of DGNet. Fig. 3 visualizes the posterior probabilities for some categories. Taking the `floor` as an example, according to the Bayesian theorem, those points with relatively high posterior probabilities are more likely to be `floor`, which explains the prediction results.

[1] A complete reasoning process can be found in the Appendix.

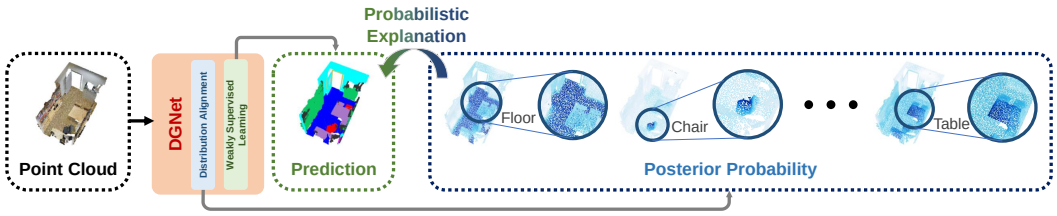

Figure 3: DGNet provides segmentation predictions from the weakly supervised learning branch and explains it probabilistically by posterior probabilities from the distribution alignment branch.

3.3.3 Loss Function

In addition to the previously mentioned initialization, we introduce a consistency loss to fortify the exchange of information between the two branches. This consistency loss is imposed on the class probability map $\mathbf{p}$ from the weakly supervised learning branch and the posterior probability $\mathbf{q}$ from the distribution alignment branch, in the form of cross-entropy:

$$\mathcal{L}_{\text{CON}} = -\frac{1}{n}\sum_{i=1}^{n} \mathbf{q}_i^\top \log(\mathbf{p}_i). \tag{9}$$

If regard the posterior probability $\mathbf{q}$ as pseudo-labels, the consistency loss is proved to diminish prediction uncertainty and alleviate distribution discrepancies in [48].

Without laborious adjustments to the weights[2], the overall loss function is defined as follows:

$$\mathcal{L} = \mathcal{L}_{\text{tCE}} + \mathcal{L}_{\text{vMF}} + \mathcal{L}_{\text{DIS}} + \mathcal{L}_{\text{CON}}. \tag{10}$$

4 Experimental Analysis

4.1 Experiment Settings

Datasets. S3DIS [1] encompasses six indoor areas, constituting a total of 271 rooms with 13 categories. Area 5 within S3DIS serves as the validation set, while the remaining areas are allocated for network training. ScanNetV2 [11] offers a substantial collection of 1,513 scanned scenes originating from 707 indoor environments with 21 indoor categories. Adhering to the official ScanNetV2 partition, we utilize 1,201 scenes for training and 312 scenes for validation. SemanticKITTI [5] with 19 classes is also considered. Point cloud sequences 00 to 10 are used in training, with sequence 08 as the validation set. To simulate sparse annotations, we randomly discard the dense annotations proportionally.

Implementation details. ResGCN-28 in DeepGCN [29] and PointNeXt-l [43] are reimplemented as the segment backbones with OpenPoints library [43]. We discard the last activation layer of the decoder to extract orientation-completed feature space. We maintain a memory bank [53] to store class prototypes across the entire dataset. In cases where class annotations are absent from the scene, the class prototypes from the memory bank are employed as supplementary initialization. We employ the LaDS [39] to maintain a higher rate of training supervision after point cloud sampling. For truncated cross-entropy loss, $\beta = 0.8$. The concentration constant $\kappa = 10$ and the iteration number $t = 10$. The distribution alignment branch is not activated in the first 50 epochs to stabilize the feature learning. In our implementation, the DGNet is trained with one NVIDIA V100 GPU on S3DIS, eight NVIDIA TESLA T4 GPUs on ScanNetV2, and one NVIDIA V100 GPU on SemanticKITTI. In the inference stage, only the weakly supervised learning branch is activated to produce predictions.

4.2 Comparative Analysis

Results on S3DIS. We detail the segmentation performance at 0.1% and 0.01% label rates on S3DIS Area 5. DGNet boosts performance for each baseline, which is evenly distributed across categories.

[2]The experiments demonstrate that DGNet is not sensitive to loss term weights.

Table 1: Quantitative comparisons on S3DIS Area 5 under various weakly supervised settings. The **bold** denotes the best performance.

Setting	Method	mIoU	ceiling	floor	wall	beam	column	window	door	chair	table	bookcase	sofa	board	clutter
100% (Fully)	PointNet [41]	41.1	88.8	97.3	69.8	**0.1**	3.9	46.3	10.8	59.0	52.6	5.9	40.3	26.4	33.2
	SQN [18]	63.7	92.8	96.9	81.8	0.0	25.9	50.5	65.9	79.5	85.3	55.7	72.5	65.8	55.9
	HybridCR [31]	65.8	93.6	98.1	82.3	0.0	24.4	59.5	66.9	79.6	87.9	67.1	73.0	66.8	55.7
	ERDA [48]	68.3	93.9	**98.5**	83.4	0.0	28.9	62.6	70.0	**89.4**	82.7	75.5	69.5	75.3	58.7
	DeepGCN [29]	60.0	90.8	97.5	76.7	0.0	24.9	51.4	52.7	76.7	83.0	61.1	62.2	58.5	44.6
	PointNeXt [43]	69.2	**94.7**	**98.5**	82.9	0.0	24.2	59.9	**74.3**	83.0	**91.4**	**76.3**	75.5	**78.6**	**60.4**
	PointTransV1 [61]	**70.4**	94.0	**98.5**	**86.3**	0.0	**38.0**	**63.4**	**74.3**	89.1	82.4	74.3	**80.2**	76.0	59.3
0.1%	SQN [18]	64.1	91.7	95.6	78.7	0.0	24.2	55.9	63.1	70.5	83.1	60.7	67.8	56.1	50.6
	CPCM [35]	66.3	91.4	95.5	**82.0**	0.0	**30.8**	54.1	70.1	79.4	87.6	67.0	70.0	**77.8**	56.6
	PointMatch [52]	63.4	-	-	-	-	-	-	-	-	-	-	-	-	-
	AADNet [39]	67.2	93.7	98.0	81.5	0.0	19.4	**59.5**	**72.0**	80.9	88.5	**78.3**	73.0	72.1	56.1
	DeepGCN [29]	43.9	93.4	97.6	68.3	0.0	19.6	39.6	4.9	47.4	35.2	59.3	50.2	2.2	32.2
	+ DGNet	58.4	91.2	97.3	76.5	0.0	22.7	47.2	42.8	73.2	85.0	62.0	59.7	58.1	44.2
	PointNeXt [43]	65.0	93.7	97.8	79.5	0.0	27.3	59.2	62.2	79.4	88.6	64.2	70.1	69.3	53.6
	+ DGNet	**67.8**	**94.3**	**98.4**	81.6	0.0	28.9	57.2	70.5	**82.3**	**90.7**	74.1	**75.2**	70.3	**58.5**
0.03%	PSD [59]	48.2	87.9	96.0	62.1	0.0	20.6	49.3	40.9	55.1	61.9	43.9	50.7	27.3	31.1
0.03%	HybridCR [31]	51.5	85.4	91.9	65.9	0.0	18.0	51.4	34.2	63.8	78.3	52.4	59.6	29.9	39.0
0.03%	DCL [57]	59.6	91.7	95.8	76.4	0.0	21.2	58.3	29.6	72.6	83.3	**64.2**	69.6	63.4	48.6
0.02%	MILTrans [56]	51.4	86.6	93.2	75.0	0.0	29.3	45.3	46.7	60.5	62.3	56.5	47.5	33.7	32.2
0.02%	ERDA [48]	48.4	87.3	96.3	61.9	0.0	11.3	45.9	31.7	73.1	65.1	57.8	26.1	36.0	36.4
0.02%	MulPro [46]	47.5	90.1	96.3	71.8	0.0	6.7	46.7	39.2	67.2	67.4	21.8	39.2	33.0	38.0
0.01%	SQN [18]	45.3	89.2	93.5	71.3	0.0	4.1	34.7	41.0	54.9	66.9	25.7	55.4	12.8	39.6
0.01%	CPCM [35]	59.3	-	-	-	-	-	-	-	-	-	-	-	-	-
0.01%	AADNet [39]	60.8	92.5	96.6	77.2	0.0	20.9	57.0	61.1	72.2	83.1	**60.1**	67.8	52.9	49.0
0.01%	DeepGCN [29]	35.9	76.7	97.1	65.2	0.0	0.0	0.4	8.1	57.6	58.0	14.9	49.1	8.5	28.2
0.01%	**+ DGNet**	52.8	92.0	97.9	75.2	0.0	23.4	22.7	33.4	74.1	83.6	29.8	62.0	47.1	45.1
0.01%	PointNeXt [43]	58.4	89.4	96.5	75.7	**0.1**	22.6	**55.3**	44.5	74.2	84.3	54.2	62.8	52.5	47.0
0.01%	**+ DGNet**	**62.4**	**93.3**	**98.1**	**80.1**	0.0	**23.3**	47.9	**53.1**	**79.4**	**87.2**	60.0	**70.6**	**65.2**	**53.1**

The lower the label rate, the more distinct the enhancement brought by DGNet, which suggests that the guidance on feature distribution is more valuable with extremely sparse annotations. Specifically, DGNet achieves more than 97% performance of fully-supervision with only 0.1% points labeled. The 0.02% label rate denotes a sparse labeling form of "one-thing-one-click". DGNet outperforms these methods without introducing super-voxel information. In addition, Fig. 4 visualizes a qualitative comparison. DGNet provides a holistic enhancement to the baseline. Although the photos on the wall are misclassified, DGNet captures consistent objects more accurately than the baseline.

Results on ScanNetV2. Compared to S3DIS, ScanNetV2 involves diverse categories and versatile scenes. Therefore, following the super-voxel setting in OTOC [36], we report the segmentation performances with 20 labeled points per scene and 1% points labeled. The cross-entropy loss term generates relatively sufficient supervised information to train the network due to introducing pseudo-labeling, resulting in a less pronounced DGNet improvement than S3DIS. However, DGNet is still slightly superior to the latest SOTA methods.

Results on SemanticKITTI. DGNet performs excellently on indoor datasets and demonstrates strong weakly supervised learning efficiency on outdoor SemanticKITTI. For a fair comparison, we replace the DGNet backbone with RandLA-Net [19]. DGNet outperforms SQN [18] with 1.0% and 2.1% mIoU on 0.1% and 0.01% label rates, respectively, demonstrating the necessity of supervision on feature space.

Table 2: Quantitative comparisons on ScanNet.

Setting	Method	mIoU (%)
100% (Fully)	PointNet [41]	33.9
	HybirdCR [31]	59.9
	PointNeXt [43]	71.2
1%	Zhang *et al*. [58]	51.1
	PSD [59]	54.7
	HybirdCR [31]	56.8
	DCL [57]	59.6
	GaIA [28]	65.2
	EDRA [48]	63.0
	AADNet [39]	66.8
	DGNet (PointNeXt)	**67.4**
20pts	Hou *et al*. [17]	55.5
	OTOC [36]	59.4
	MILTrans [56]	54.4
	DAT [54]	55.2
	PointMatch [52]	62.4
	EDRA [48]	57.0
	AADNet [39]	62.5
	DGNet (PointNeXt)	**62.9**

Table 3: Quantitative comparisons on SemanticKITTI.

Setting	Method	mIoU (%)
0.1%	RPSC [26]	50.9
	SQN [18]	50.8
	DGNet (RandLA-Net)	**51.8**
0.01%	CPCM [35]	34.7
	SQN [18]	39.1
	DGNet (RandLA-Net)	**41.2**

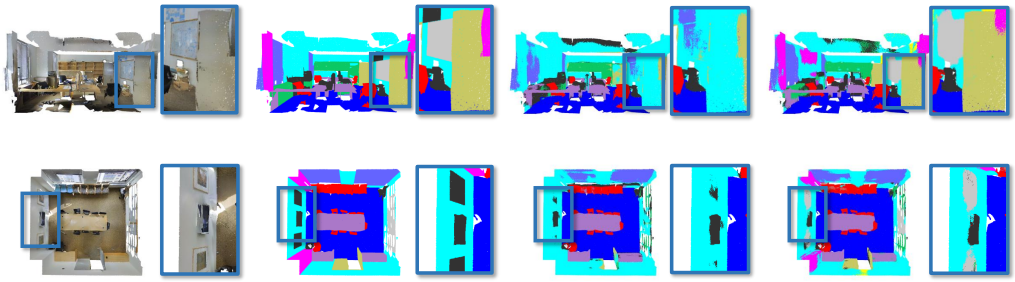

Figure 4: Visual comparisons between baseline and our DGNet on S3DIS Area 5 at 0.01% label rate.

Table 4: Comparisons on feature distribution description selection in distribution alignment branch.

Distribution	Distribution Modeling	Distance Metric		mIoU (%)
		Euclidean Norm	Cosine Similarity	
PN [45]	Category Prototype	✓	○	59.9
HPN [37]		○	✓	60.3
GMM	Mixture Models	✓	○	61.3
moVMF		○	✓	**62.4**

4.3 Ablations and Analysis

All ablation studies are performed on S3DIS with PointNeXt-l as baseline.

Distribution comparison. We impose a comparison experiment in the distribution alignment branch of DGNet for distribution selection. The relevant experimental results are reported in Tab. 4. For category prototype models, we discard the vMF loss $\mathcal{L}_{\text{vMF}}$ and consistency loss $\mathcal{L}_{\text{CON}}$ due to the lack of corresponding forms. For the mixture model with Euclidean Norm (GMM), we replace the maximum likelihood estimation in GMM form with the $\mathcal{L}_{\text{vMF}}$ in DGNet. In terms of distance metrics, cosine similarity trumps Euclidean norm. In terms of distribution modeling, mixture models have a significant performance advantage over the category prototype models. Integrating these two aspects, the stronger fitting ability of moVMF leads to more accurate and effective supervised signals for weakly supervised learning in DGNet.

Ablation study for loss terms. Tab. 5 demonstrates the validity of each loss term in DGNet. Compared with partial cross-entropy loss, the truncated cross-entropy loss improves segmentation performance due to its avoidance of overfitting. Performance improvements are obtained by imposing $\mathcal{L}_{\text{vMF}}$ with soft assignment form, $\mathcal{L}_{\text{DIS}}$ and $\mathcal{L}_{\text{CON}}$ individually, and optimal performance is achieved by using these loss terms simultaneously. In contrast to the soft assignment, the hard assignment does not take into account the inter-cluster similarity and is mismatched with the soft-moVMF algorithm. Therefore, $\mathcal{L}_{\text{vMF}}$ with hard assignment form in hard-moVMF algorithm and KNN-moVMF algorithm undermines the segmentation efficiency.

Ablation study for Nested EM Algorithm. We ablate the proposed Nested Expectation-Maximum Algorithm in two respects. First, we optimize certain parameters on moVMF and fix other parameters with initialized values. The first, second, third, and last rows in Tab. 6 reveal that individually optimizing parts of the parameters impairs the segmentation performance. Secondly, we ablate how the parameters of moVMF are updated. Compared with kNN-moVMF (fourth raw) and hard-moVMF (fifth raw) in [3], the soft assignment strategy delivers 2.6% and 2.2% mIoU improvements, respectively. This shows that the optimization of moVMF parameters benefits from the soft assignment strategy.

Hyperparameter selection. In Tab. 7, we search the parameter space for suitable κ, t, and β. We observe that (a) the segmentation performance shows an increasing and then decreasing trend as the

Table 5: Ablation study for loss terms.

$\mathcal{L}_{\text{CE}}$		$\mathcal{L}_{\text{vMF}}$		$\mathcal{L}_{\text{DIS}}$	$\mathcal{L}_{\text{CON}}$	mIoU (%)
$\mathcal{L}_{\text{pCE}}$	$\mathcal{L}_{\text{tCE}}$	hard	soft			
✓						58.4
	✓					59.1
	✓	✓				58.9
	✓		✓			60.4
	✓			✓		59.8
	✓				✓	61.1
	✓		✓	✓	✓	**62.4**

Table 6: Ablation studies for Nested Expectation-Maximum Algorithm.

E step	α	μ	mIoU (%)
None	○	○	61.0
soft-moVMF	✓	○	60.9
soft-moVMF	○	✓	60.0
hard-moVMF	✓	✓	59.8
KNN-moVMF	✓	✓	60.2
soft-moVMF	✓	✓	**62.4**

concentration constant κ increases. Our analysis suggests that too small κ leads to a dispersion of features within the class, which can be easily confused with other classes. And too large κ forces overconcentration of features within the class and overfits the network. (b) As the iteration number t increases, the segmentation performance gradually rises and then stabilizes. We believe that the soft-moVMF algorithm gradually converges as t increases, and increasing t after convergence will no longer bring further gains to the network. (c) As the truncated threshold β decreases, the segmentation performance shows a tendency to first increase and then decrease. The conventional cross-entropy loss function is the truncated cross-entropy loss function with $\beta = 1$. When β decreases, the overfitting on sparse annotations is alleviated, but when β is too small, it weakens the supervised signal on sparse labeling leading to performance degradation.

Table 7: Hyperparameter selection for the (a) concentration constant κ, (b) iteration number t and (c) truncated threshold β.

(a)

κ	mIoU (%)
0.1	57.7
1	60.3
10	**62.4**
20	59.5

(b)

t	mIoU (%)
0	61.0
5	61.5
10	**62.4**
15	62.3

(c)

β	mIoU (%)
0.7	61.9
0.8	**62.4**
0.9	62.2
1	61.7

5 Limitations and Future Work

Despite the promising performance achieved by DGNet, exploring the distribution of embeddings is preliminary. Feng *et al*. [12] proposes a more sophisticated distribution to restrict feature learning with full supervision. However, such refinement restrictions will lead to overfitting under sparse annotations. Therefore, how to prevent weakly-supervised learning overfitting with enhanced feature description is a promising research topic.

6 Conclusion

In this paper, we propose a novel perspective by regulating the feature space for weakly supervised point cloud semantic segmentation and develop a distribution guidance network to verify the superiority of this perspective. Based on the investigation of the distribution of semantic embeddings, we choose moVMF to describe the intrinsic distribution. In DGNet, we alleviate the underfitting across the entire dataset and overfitting within the labeled points. Extensive experimental results demonstrate that DGNet rivals or even surpasses the recent SOTA methods on S3DIS, ScanNetV2, and SemanticKITTI. Moreover, DGNet demonstrates the interpretability of network predictions and scalability to various label rates. We expect our work to inspire the point cloud community to strengthen the inherent properties of weakly supervised learning.

Acknowledgements

This work was supported by Shenzhen Science and Technology Program under Grant KQTD20180411143338837.

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1534–1543, 2016.

[2] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[3] Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, Suvrit Sra, and Greg Ridgeway. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(9), 2005.

[4] Florian Barbaro and Fabrice Rossi. Sparse mixture of von mises-fisher distribution. In *ESANN*, 2021.

[5] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307, 2019.

[6] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[10] Zisheng Chen, Hongbin Xu, Weitao Chen, Zhipeng Zhou, Haihong Xiao, Baigui Sun, Xuansong Xie, et al. Pointdc: Unsupervised semantic segmentation of 3d point clouds via cross-modal distillation and super-voxel clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14290–14299, 2023.

[11] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[12] Tuo Feng, Wenguan Wang, Xiaohan Wang, Yi Yang, and Qinghua Zheng. Clustering based point cloud representation learning for 3d analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8283–8294, 2023.

[13] Hariprasath Govindarajan, Per Sidén, Jacob Roll, and Fredrik Lindsten. DINO as a von mises-fisher mixture model. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[14] Agrim Gupta, Silvio Savarese, Surya Ganguli, and Li Fei-Fei. Embodied intelligence via learning and evolution. *Nature communications*, 12(1):5721, 2021.

[15] Christian Hane, Christopher Zach, Andrea Cohen, Roland Angst, and Marc Pollefeys. Joint 3d scene reconstruction and class segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 97–104, 2013.

[16] Md Abul Hasnat, Julien Bohné, Jonathan Milgram, Stéphane Gentric, and Liming Chen. von mises-fisher mixture model-based deep learning: Application to face verification. *arXiv preprint arXiv:1706.04264*, 2017.

[17] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15587–15597, 2021.

[18] Qingyong Hu, Bo Yang, Guangchi Fang, Yulan Guo, Aleš Leonardis, Niki Trigoni, and Andrew Markham. Sqn: Weakly-supervised semantic segmentation of large-scale 3d point clouds. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, pages 600–619. Springer, 2022.

[19] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11108–11117, 2020.

[20] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023.

[21] Jyh-Jing Hwang, Stella X Yu, Jianbo Shi, Maxwell D Collins, Tien-Ju Yang, Xiao Zhang, and Liang-Chieh Chen. Segsort: Segmentation by discriminative sorting of segments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7334–7344, 2019.

[22] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019.

[23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[24] Hyeokjun Kweon, Jihun Kim, and Kuk-Jin Yoon. Weakly supervised point cloud semantic segmentation via artificial oracle. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3721–3731, 2024.

[25] Hyeokjun Kweon and Kuk-Jin Yoon. Joint learning of 2d-3d weakly supervised semantic segmentation. *Advances in Neural Information Processing Systems*, 35:30499–30511, 2022.

[26] Yuxiang Lan, Yachao Zhang, Yanyun Qu, Cong Wang, Chengyang Li, Jia Cai, Yuan Xie, and Zongze Wu. Weakly supervised 3d segmentation via receptive-driven pseudo label consistency and structural consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1222–1230, 2023.

[27] Pierre Simon Laplace. *Théorie analytique des probabilités*. Courcier, 1820.

[28] Min Seok Lee, Seok Woo Yang, and Sung Won Han. Gaia: Graphical information gain based attention network for weakly supervised point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 582–591, 2023.

[29] Guohao Li, Matthias Müller, Guocheng Qian, Itzel C Delgadillo, Abdulellah Abualshour, Ali Thabet, and Bernard Ghanem. Deepgcns: Making gcns go as deep as cnns. *IEEE transactions on pattern analysis and machine intelligence*, 45(6):6923–6939, 2021.

[30] Junnan Li, Pan Zhou, Caiming Xiong, and Steven C. H. Hoi. Prototypical contrastive learning of unsupervised representations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[31] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14930–14939, 2022.

[32] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31, 2018.

[33] Zhuangzi Li, Ge Li, Thomas H Li, Shan Liu, and Wei Gao. Semantic point cloud upsampling. *IEEE Transactions on Multimedia*, 25:3432–3442, 2022.

[34] Kangcheng Liu, Yuzhi Zhao, Qiang Nie, Zhi Gao, and Ben M Chen. Weakly supervised 3d scene segmentation with region-level boundary awareness and instance discrimination. In *European conference on computer vision*, pages 37–55. Springer, 2022.

[35] Lizhao Liu, Zhuangwei Zhuang, Shangxin Huang, Xunlong Xiao, Tianhang Xiang, Cen Chen, Jingdong Wang, and Mingkui Tan. Cpcm: Contextual point cloud modeling for weakly-supervised point cloud semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18413–18422, 2023.

[36] Zhengzhe Liu, Xiaojuan Qi, and Chi-Wing Fu. One thing one click: A self-training approach for weakly supervised 3d semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1726–1736, 2021.

[37] Pascal Mettes, Elise Van der Pol, and Cees Snoek. Hyperspherical prototype networks. *Advances in neural information processing systems*, 32, 2019.

[38] Nicolas Michel, Giovanni Chierchia, Romain Negrel, and Jean-François Bercher. Learning representations on the unit sphere: Investigating angular gaussian and von mises-fisher distributions for online continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 14350–14358, 2024.

[39] Zhiyi Pan, Nan Zhang, Wei Gao, Shan Liu, and Ge Li. Adaptive annotation distribution for weakly supervised point cloud semantic segmentation. *arXiv preprint arXiv:2312.06259*, 2023.

[40] Zhiyi Pan, Nan Zhang, Wei Gao, Shan Liu, and Ge Li. Less is more: label recommendation for weakly supervised point cloud semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4397–4405, 2024.

[41] 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*, pages 652–660, 2017.

[42] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[43] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 35:23192–23204, 2022.

[44] Tyler R Scott, Andrew C Gallagher, and Michael C Mozer. von mises-fisher loss: An exploration of embedding geometries for supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10612–10622, 2021.

[45] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[46] Yongyi Su, Xun Xu, and Kui Jia. Weakly supervised 3d point cloud segmentation via multi-prototype learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[47] Jalil Taghia, Zhanyu Ma, and Arne Leijon. Bayesian estimation of the von-mises fisher mixture model with variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 36(9):1701–1715, 2014.

[48] Liyao Tang, Zhe Chen, Shanshan Zhao, Chaoyue Wang, and Dacheng Tao. All points matter: Entropy-regularized distribution alignment for weakly-supervised 3d segmentation. *Advances in Neural Information Processing Systems*, 36, 2024.

[49] An Tao, Yueqi Duan, Yi Wei, Jiwen Lu, and Jie Zhou. Seggroup: Seg-level supervision for 3d instance and semantic segmentation. *IEEE Transactions on Image Processing*, 31:4952–4965, 2022.

[50] Weikang Wan, Haoran Geng, Yun Liu, Zikang Shan, Yaodong Yang, Li Yi, and He Wang. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3891–3902, 2023.

[51] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4384–4393, 2020.

[52] Yushuang Wu, Zizheng Yan, Shengcai Cai, Guanbin Li, Xiaoguang Han, and Shuguang Cui. Pointmatch: A consistency training framework for weakly supervised semantic segmentation of 3d point clouds. *Computers & Graphics*, 116:427–436, 2023.

[53] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

[54] Zhonghua Wu, Yicheng Wu, Guosheng Lin, Jianfei Cai, and Chen Qian. Dual adaptive transformations for weakly supervised point cloud segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, pages 78–96. Springer, 2022.

[55] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 574–591. Springer, 2020.

[56] Cheng-Kun Yang, Ji-Jia Wu, Kai-Syun Chen, Yung-Yu Chuang, and Yen-Yu Lin. An mil-derived transformer for weakly supervised point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11830–11839, 2022.

[57] Baochen Yao, Hui Xiao, Jiayan Zhuang, and Chengbin Peng. Weakly supervised learning for point cloud semantic segmentation with dual teacher. *IEEE Robotics and Automation Letters*, 2023.

[58] Yachao Zhang, Zonghao Li, Yuan Xie, Yanyun Qu, Cuihua Li, and Tao Mei. Weakly supervised semantic segmentation for large-scale point cloud. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3421–3429, 2021.

[59] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15520–15528, 2021.

[60] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

[61] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021.

[62] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

Appendix A Derivation of Optimization Objective Function

According to maximum likelihood estimation, the optimization objective function is defined as:

$$
\begin{aligned}
&\max_{\phi,\mathcal{Z},\Theta} P(\mathcal{V}|\mathcal{Z},\Theta) \\
&= \min_{\phi,\mathcal{Z},\Theta} -\prod_i P(\mathbf{v}_i|z_i,\Theta_{z_i}) \\
&= \min_{\phi,\mathcal{Z},\Theta} -\sum_i \log(\alpha_{z_i} f(\mathbf{v}_i|\kappa_{z_i},\mathbf{u}_{z_i})) \\
&= \min_{\phi,\mathcal{Z},\Theta} -\sum_i \log(\alpha_{z_i} C_d(\kappa_{z_i}) \exp(\kappa_{z_i}\mathbf{u}_{z_i}^\top \mathbf{v}_i)) \\
&= \min_{\phi,\mathcal{Z},\Theta} -\sum_i \Big[\log(C_d(\kappa_{z_i})) + \log(\alpha_{z_i}) + \kappa_{z_i}\mathbf{u}_{z_i}^\top \mathbf{v}_i\Big].
\end{aligned}
\tag{11}
$$

To avoid the long-tail problem and simplify computations, we set a constant value for κ on each class c. Therefore, the objective optimization function Eq. 11 can be further simplified as:

$$
\max_{\phi,\mathcal{Z},\Theta} P(\mathcal{V}|\mathcal{Z},\Theta) = \min_{\phi,\mathcal{Z},\Theta} -\sum_{i=1}^{n} \Big[\log(\alpha_{z_i}) + \kappa\mathbf{u}_{z_i}^\top \mathbf{v}_i\Big]. \tag{12}
$$

Algorithm 1: soft-moVMF Algorithm

Input: Normalized Embeddings $\mathcal{V} = \{\mathbf{v}_i | i = 1, 2, \cdots, n\}$, Initial Clustering Centers $\mathcal{H} = \{\mathbf{h}_c | c \in \mathbb{C}\}$
Output: Soft Assignments $\mathcal{Q}$, Clustering Results $\mathcal{Z}$, Parameters of moVMF Θ

```
/* Initialize α, u in Θ */
for category index c of ℂ do
    α_c, u_c = 1/|ℂ|, h_c
end
repeat
    /* The Expectation step of EM */
    for point index i = 1 to n do
        for category index c of ℂ do
            f(v_i|κ, u_c) = C_d(κ) exp(κ u_c^⊤ v_i)
        end
        /* Compute the posterior probability q_i */
        for category index c of ℂ do
            P(c|v_i, Θ) = α_c exp(κ u_c^⊤ v_i) / Σ_{l∈ℂ} α_l exp(κ u_l^⊤ v_i)
        end
    end
    /* The Maximization step of EM */
    for category index c of ℂ do
        α_c = (1/n) Σ_{i=1}^n P(c|v_i, Θ)
        u_c = Σ_{i=1}^n v_i P(c|v_i, Θ) / ‖Σ_{i=1}^n v_i P(c|v_i, Θ)‖
    end
until Convergence
return Q = P(ℂ|V, Θ), Z = argmax_{c∈ℂ}(Q), Θ = {α_c, u_c | c ∈ ℂ}
```

Appendix B The soft-moVMF Algorithm

Initialization. Due to the sparsity of the annotations, some classes in the scene may lack any labeled points, thereby hindering proper initialization. Consequently, we maintain a memory bank [53] to store class prototypes across the entire dataset. Specifically, the mean embedding directions on labeled points are set as the initial vectors for categories with labeled points in the point cloud scene.

For the missing categories of this scene, we retrieve the category prototypes ρ from the memory bank as supplementary initialization. Consequently, the initial vector $\mathbf{h}_c$ is formulated as:

$$\mathbf{h}_c = \begin{cases} \frac{\sum_{y_i=c} \mathbf{v}_i}{\|\sum_{y_i=c} \mathbf{v}_i\|} & c \in \mathbf{Y} \\ \rho_c & c \notin \mathbf{Y} \end{cases} . \tag{13}$$

Pseudo Code. As presented in Algorithm 1, we incorporate prior knowledge about the semantics during the optimization process based on soft-moVMF [3]. The weights $\alpha = 1/|\mathbb{C}|$ are initialized uniformly. Subsequently, based on the cosine similarity between features on each point and mean directions of each category, the prior probability f and the posterior probability $\mathbf{q}_i$ are estimated. Finally, we determine the clustering result z_i for each point by $\mathrm{argmax}(\mathbf{q}_i)$. After updating the mean directions $\mathbf{u}$ and weights α, the process is repeated until the clustering results converge.

Appendix C More Experimental Results

Impact of label rates. To demonstrate the capability of DGNet on extreme label rates, we compare the segmentation performance on sparse annotations over a larger range of rates. Tab. 8 reports the mIoU performance of the DGNet and baseline at 10%, 1%, 0.1%, 0.01%, and 0.001% label rates. It can be observed that at 100,000 times less sparse annotations, the baseline fails to learn accurate semantic embedding from it. At the same time, our DGNet still maintains acceptable segmentation performance since it can be conducted unsupervised.

Method	10%	1%	0.1%	0.01%	0.001%
PointNeXt	69.3	68.3	67.0	60.8	44.7
DGNet (PointNeXt)	69.5	68.8	67.8	62.4	51.5

Table 8: Performance comparison on various label rates.

Varying labeled points. Following SQN [18], we verified the sensitivity of DGNet (PointNeXt) to different labeled points at the same label rate. We repeated the experiment five times for each label setting, keeping the network and label rate unchanged and changing only the labeled points' locations. In Tab. 9, we observe a slight performance fluctuation within a reasonable range.

Setting	Trail#1	Trail#2	Trail#3	Trail#4	Trail#5	Mean	STD
0.1%	**67.8**	66.9	66.7	67.6	67.3	67.3	0.42
0.01%	62.0	**62.4**	61.4	61.7	62.0	61.9	0.33

Table 9: Sensitivity analysis of DGNet on S3DIS Area 5.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claim of this paper is using mathematically definable feature distributions to promote the learning of point cloud semantic segmentation under weak supervision.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of our work can be found in Sec. 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The complete derivation of Eq. 5 is given in Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The relevant code and data will be open-sourced upon acceptance of the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The relevant code and data will be open-sourced upon acceptance of the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The main details are shown in Sec. 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Following other weak supervision methods, we do not provide error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: We only provide information on the computer resources for the main experiment.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We conduct in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of this work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original papers that produce the codes and datasets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.