An Approach to Point Cloud Model Pretraining from Image Networks Based on Knowledge Distillation with Multi-Scale Feature Extraction

Aleksei Kuchkov[1], Alexey Kashevnik[2] Larisa Gonchar[1], Evgeniia Zinchik[1], Ekaterina Brovkina[1]
[1]ITMO University, St. Petersburg, Russia
[2]SPC RAS, St. Petersburg, Russia
kuchkovaleksei@gmail.com, alexey.kashevnik@iias.spb.su, lgonchar91@mail.ru, zinchike@mail.ru, brovkina20146@gmail.com

***Abstract*—Recent advances in building large-scale point cloud computer vision models have enabled a variety of high-capacity architectures for many computer vision tasks. The task of training point cloud models (for such tasks as segmentation, object detection, and other) involves a large amount of accurately labeled training data, which are not always available for particular scenarios. To mitigate that issue, the training process was adapted to pretrain these models using existing ones with a process called knowledge distillation (KD). KD allows to transfer knowledge (neural network weights) from one network to another. A special place for KD was found in multimodal point cloud computer vision models, where an image model (first modality) is trained on a larger dataset,becomes a teacher to the point cloud model (second modality) which lack labeled data. Despite the effectiveness of many KD methods in transferring knowledge between different modalities, they typically compute distillation losses at a single spatial scale, underutilizing the multiscale, multi-head structure of modern image models. In order to fully utilize these multiscale features, we introduce a multiscale KD approach to point cloud model pretraining that incorporates an Atrous Spatial Pyramid Pooling (ASPP) block to extract multiscale feature tensors and applies a contrastive loss to align them with point cloud branch representations. The proposed approach can be used with existing methods to calculate multiscale tensors extracted from 2D image networks for further distillation with 3D tensors from the 3D branch. Evaluations are performed on the nuScenes dataset and show improved performance over a baseline while maintaining a comparable parameter count.**

I. Introduction

Point clouds represent a rich geometric information expressed in a sparse 3D format. Sensors such as LiDAR perceive environmental information with time-of-flight (TOF) methods and represent their measurements as point clouds. This format is crucial for 3D world perception tasks, including 3D object detection, segmentation, and tracking. However, due to the complex nature of point clouds, the annotation task, required for models trained in supervised deep learning, can be tedious, especially if done manually. Moreover, large environment perception models typically require extensive datasets due to the high capacity of modern deep learning architectures [1], [2]. Supervised and self-supervised training methods can be useful in this context, as they can help pretrain models using knowledge distilled from pretrained models, either within the same modality or across different modalities.

Knowledge distillation from 3D to 2D representation was first introduced in Learning from 2D [3] which pioneered the 3D-to-2D knowledge distillation process with the CL_{PPNCE} function, incorporating semantic information from a trained 2D model. Self-supervised image-to-LiDAR distillation for autonomous driving [4] refined this technique by using pixel–point correspondences to generate superpixels with SLidR, thus reducing projection errors. Self-Supervised Image-to-Point Distillation via Semantically Tolerant Contrastive Loss [5] further improved the framework with a semantically tolerant contrastive constraint and a class-balancing loss function. More recently, Segment Any Point Cloud Sequences by Distilling Vision Foundation Models [6] employed Vision Foundation Models (VFMs) for superpixel generation, reducing the labeling burden. Finally, Olivine [7] proposed to incorporate both label and class semantics through a modified contrastive loss function (CL_{sup})

Despite all advances, the above methods lack multiscale feature distillation because they rely on single-head architectures. In this work, we investigate pretraining 3D models with a multiscale approach.

Our motivation is to improve knowledge distillation performance with multiscale feature extraction, thereby accelerating the training of large 3D models with minimal labeled data. Furthermore, we aim to simplify the multiscale extraction module to enhance compatibility with other networks.

Our contribution can be summarized as follows.

- We propose and evaluate multiscale feature extraction in two ways - with dilated convolution and raw ResNet $layer_i$ outputs;
- We demonstrate the effectiveness of dilated convolution for achieving multiscale feature maps from ResNet50 for multiscale feature extraction;

Based on previous achievements in knowledge distillation, we move forward to explore the multiscale nature of CNN networks in order to fully utilize 2D feature knowledge. In order to fully explore CNN features at several scales, we use

dilated convolution with four projection heads, each one of them ending up with an upsampling layer. The advantage of this approach is that the multiscale extraction module can be seamlessly attached to any network.

Building upon prior work in knowledge distillation, we extend the exploration of CNN features across multiple scales in order to fully exploit 2D feature representations. To achieve this, we employ dilated convolutions with four projection heads, each followed by an upsampling layer. The advantage of this approach is that the multiscale extraction module can be seamlessly attached to any network.

The remainder of this paper is organized as follows. Section II reviews knowledge distillation with contrastive loss. Section III discusses existing techniques for knowledge distillation. Section IV presents our proposed multiscale feature extraction method. Section V details the implementation and experimental results.

II. Related work

LiDAR-based detection and segmentation networks have advanced significantly in both size and performance. Recent approaches include transformers, 3D convolution networks with large kernels [8] [9], BEV-based methods [10], [11], and various other techniques which require an enormous amount of real-world data. This creates challenges when training such models for specialized tasks, as collecting sufficient labeled training data becomes infeasible. In general, the larger a model becomes, the more training data it requires.

To address the issue, various strategies have been proposed to reduce the dependence on high-quality labeled data. Among them, self-supervised and weakly supervised methods have gained popularity because they reduce or eliminate the need for labeled training data in the LiDAR modality. In self-supervised methods, feature maps from a teacher model are used to guide the training of a student model without ground-truth labels from a training dataset. In weakly supervised methods, the student exploits weak labels provided by a teacher model, which can be either another model or the same model at an earlier training stage. The process of transferring knowledge from one model to another is called knowledge distillation (KD). Broadly, KD occurs either at the feature level, where feature maps are distilled without class information, or at the semantic level, where class information is incorporated to match point–pixel pairs not only by geometry but also by semantic labels.

In computer vision, KD allows pretraining one model with the knowledge of another model, presumably larger or trained on a large-scale dataset. In other words, KD compresses the knowledge of one or more networks into another network. KD for training computer vision models was first applied in [12] to distill knowledge from an assembly of neural networks to a single one by matching soft class probabilities. Since then, KD has been widely used not only in single modal tasks, but also in cross-model architectures, transferring knowledge from one modality to another to enrich the model with additional information not presented in the first modality.

The most recent advancements in transferring knowledge in 2D dimension include Generalized Knowledge Distillation [13], which trains the student on its self-generated output sequences, feature-based KD methods: [14]–[16]; Similarity-based Distillation: [17]–[19]; Logit-based Distillation: [20]–[22].

In extension to 3D knowledge distillation, the most recent approaches include [23], which relies on a teacher's semantics to train a student model, [24], which uses self-distillation in addition to KD from a teacher model, [25], which transfers knowledge from one 3D model to another with KD Loss.

Another branch of methods adopts intermediate representations, such as text [26], [27] and depth (from LiDAR to monocular camera) [28].

The combination of supervised and self-supervised methods shows the best performance, which was shown in recent studies [5], [7]. Supervised methods offer, either weak or strong labels, for supervised knowledge distillation guidance. These labels are used to build label-to-point pairs instead of pixel-to-point. Label-to-point pairs in supervised methods respect labels of the corresponding points and pixels, which allows such methods to achieve better performance.

Supervised knowledge distillation with contrastive learning significantly improved KD performance [7]. Moreover, supervised methods work well in reverse order, as demonstrated in [29] with the training monocular depth detector using 3D network knowledge.

Another issue with cross-model KD is the mutual sensor calibration stability. For calculating pixel-to-point pairs, precise calibration is required to achieve high-quality pairs. However, that approach is impractical because of instability in real-world multisensory systems. To mitigate the problem, the authors of the paper [4] proposed a method for clustering visually coherent regions of images into superpixels. The proposed approach significantly reduced the dependency of KD on individual sensor calibration and was widely adapted in future works [7], [6], [30]. Instead of relying on precise calibration, point-to-pixel and label-to-point pairs are built with superpixels, which are more robust to calibration errors. To generate superpixels, SLIC (Simple Linear Iterative Clustering) [31] was used. Later, that task was delegated to Visual Foundation Models (VFM), such as SAM and Dino. In the current work, we also utilize VFM for superpixel generation while leveraging the advantages of multiscale feature extraction for KD.

Based on the presented analysis, we conclude that the next step toward enhancing knowledge distillation performance is to apply it across multiple scales. This can be achieved in several ways:

- By leveraging multi-head outputs from the image branch;
- By leveraging multi-head outputs from the LiDAR branch;
- By incorporating attention mechanisms in one or both branches.

In this work, we show that utilizing the multi-head outputs of the image network improves knowledge distillation per-

formance while introducing only minimal weight overhead compared to the original model.

III. Background

The core of most knowledge distillation approaches typically involves either Mutual Information (MI) or Contrastive Loss (CL). Among these, contrastive loss has become the most widely adopted in recent studies. The basic formulation, first introduced in [32], brings positive matches from two views closer together while pushing negative pairs farther apart:

$$\mathcal{L}_{PPNCE} = -\sum_{(i,j)\in\mathcal{P}} \log \frac{\exp\left(\frac{\mathbf{f}_i\cdot\mathbf{f}_j}{\tau}\right)}{\sum_{(\cdot,k)\in\mathcal{P},k\neq j}\exp\left(\frac{\mathbf{f}_i\cdot\mathbf{f}_k}{\tau}\right)}$$

Where:

- $\mathcal{P}$ is the set of positive feature matches between two views.
- $\mathbf{f}_i$ is a query feature (e.g., from view 1).
- $\mathbf{f}_j$ is the positive key (e.g., from view 2).
- $\mathbf{f}_k$ are negative keys (other points in view 2, except j).
- τ is a temperature scaling factor.

The later approach, firstly demonstrated in Olivine [7], takes the label information into consideration and uses a modified version alongside the original one:

$$\mathcal{L}_{\text{sup}} = -\frac{1}{M_s}\sum_{i=1}^{M_s}\log\left[\frac{1}{|A(i)|}\sum_{a\in A(i)}\frac{\exp\left(\langle G_i^{\text{3D}}, G_a^{\text{2D}}\rangle/\tau\right)}{\sum_{j=1}^{M_s}\exp\left(\langle G_i^{\text{3D}}, G_j^{\text{2D}}\rangle/\tau\right)}\right]$$

Where:

- M_s is the number of pairs of point-pixel sampled in the batch.
- G_i^{3D} and G_a^{2D} are feature vectors of the 3D point and the 2D pixel, respectively.
- $A(i)$ denotes the set of 2D pixel indices that have the **same semantic label** as the i-th 3D point.
- $|A(i)|$ is the cardinality (number of elements) of set $A(i)$.
- τ is the temperature scalar for contrastive scaling.

Olivine [7] has successfully used a combination of CL functions – CL_{PPNCE} and CL_{SUP}. However, that approach uses results of the last layer of the ResNet50 image backbone with a simple projection head, which leads to significant information loss.

IV. Point cloud models from distilled knowledge

In this section, we examine the process of knowledge distillation from an image network to a point cloud model. The procedure is divided into two main parts: pretraining and fine-tuning. Firstly, we explore the knowledge distillation flow from the 2D modality to the 3D (Section A). Then, in Section B we introduce a multiscale feature extraction method based on ASPP dilation layers, which allows extracting features on different scales from the 2D network. Finally, in Section C, we describe the underlying point cloud model baseline architecture.

A. Image baseline architecture

We utilize ResNet50 network pretrained using MoCov2 [33] on the ImageNet dataset [34]. The architecture is presented in Fig. 1. We don't use any projection heads after the last layer. Instead, we directly employ the outputs of layer4 in the ASPP block.

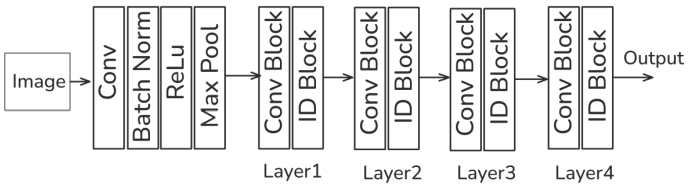


Fig. 1. Underlying resnet50 architecture with disconnected FC part

B. Point cloud baseline architecture

We utilize unmodified U-Net 34 (SR-UNet34) with a projection layer at the end, which projects point cloud features in a common feature space with image features.

$$z_{feat}^{3D} = h_{feat}^{3D}(x)$$

where z_{feat}^{3D} is a projection head with 64 output channels.

Fig. 2 shows the point cloud baseline architecture SR-UNet34.

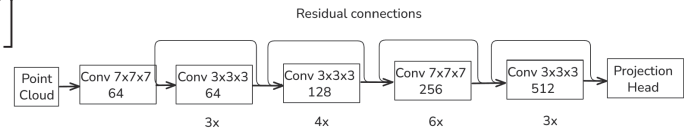


Fig. 2. Point cloud baseline SR-UNet34 architecture followed by the projection layer

C. Knowledge distillation flow

Firstly, in order to make the proposed approach flexible, the knowledge distillation flow is conducted in a way that allows only the final output tensors from the point-cloud model to be used, depicted as FM class and FM feat. in Fig. 3. Class- and feature-level projection heads are used in the pretrain mode only to propagate weight updates back to the model.

We adopt a one-directional knowledge distillation strategy, but extend it with distinct projection heads for each feature-map scale, implemented through four parallel projection heads. Our key hypothesis is that multiscale features, derived from a single feature map of an image network, can enhance KD performance in the $2D \Rightarrow 3D$ pipeline.

D. Multiscale feature extraction

To improve the performance of knowledge distillation, we propose a multiscale feature extraction method. This approach employs four dilated convolution layers organized as an ASPP block, enabling feature extraction across multiple scales.

Here and later, we use f^{DIM} and z^{DIM} to identify individual layers and feature maps accordingly. For image tensors, we

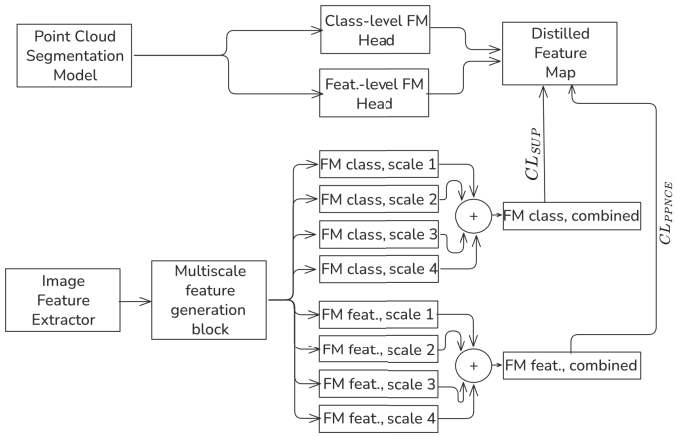


Fig. 3. Knowledge distillation process for point cloud model training. Operator + stands for feature map concatenation. CL_{sup} and $CLP\ PNCE$ stand for supervised and unsupervised knowledge distillation contrastive learning functions

use $I_{C\times H\times W}$, and $P_{C\times H\times W\times D}$ for point clouds. $h^{DIM}(x)$ identifies a projection function for an input x.

ResNet50 baseline image feature extraction network is followed by 4 dilation heads with dilation rates 1, 6, 12, and 18. The whole pipeline can be described as follows.

After ResNet50 layer4, we obtain a single feature map:

$$z^{2D} = f^{2D}(x)$$

Where:

- f^{2D} is ResNet50 encoder (layer4 output).
- z^{2D} is a feature map received from the encoder.

In the feature extraction stage, we apply 4 dilated convolution on the feature map received from the previous step:

$$z^{2D}_{d1} = h^{2D}_{d1}(x)$$
$$z^{2D}_{d2} = h^{2D}_{d2}(x)$$
$$z^{2D}_{d3} = h^{2D}_{d3}(x)$$
$$z^{2D}_{d4} = h^{2D}_{d4}(x)$$

where

- z^{2D}_{di} are feature maps obtained after the dilation layer i.
- h^{2D}_{di} are dilated layers.

All dilated convolution is performed in a pyramidal feature network pipeline, as depicted in Fig. 4.

In order to receive global features, we follow the original pipeline from [35]:

$$z^{2D}_{gp} = h^{2D}_{gp}(x)$$

where

- z^{2D}_{gp} is a pooled feature map
- h^{2D}_{gp} is a global average pooling layer implemented via AdaptiveAvgPool(1).

After performing feature extraction on all dilated convolution layers, a single feature tensor is built:

$$z_{conc} = [z^{2D}_{d1}, z^{2D}_{d2}, z^{2D}_{d3}, z^{2D}_{d4}, z^{2D}_{gp}] \in R^{4\times C\times W\times H}$$

The concatenated feature tensor has dimensions 4 times scaled down from their original sizes.

In order to receive both class-level and feature-level feature maps, we follow the original pipeline with two identical heads trained with CL_{PPNCE} and CL_{SUP}, respectively:

$$z^{2D}_{class} = h^{2D}_{class}(z_{conc})$$
$$z^{2D}_{feat} = h^{2D}_{feat}(z_{conc})$$

where

- h^{2D}_{class} - class-level projection head.
- h^{2D}_{feat} - feature-level projection head.

The projection heads project feature maps in a common feature space with the U-Net 34 output tensor.

In order to use weak labels obtained from vFM, an upsampling layer is used to restore the concatenated tensor to the original size of an image (416x224) with a bilinear interpolation.

All steps are implemented in the ASPP block, which is depicted in Fig. 4. ASPP block is completely detached from the ResNet50 network, making it easily transferable to other architectures.

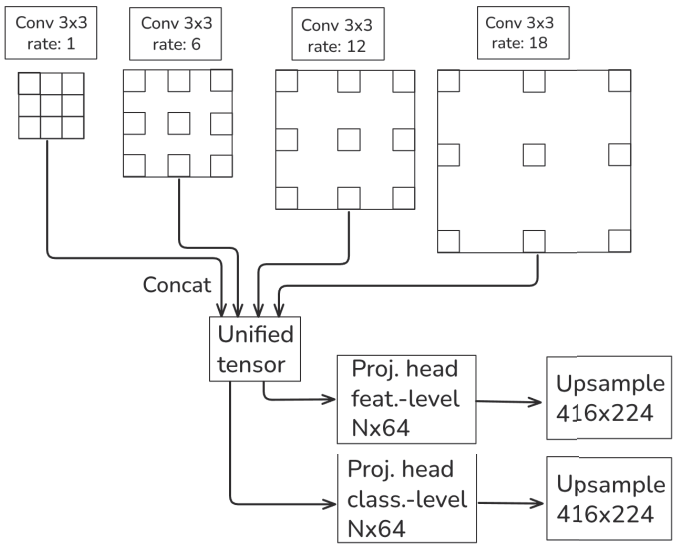


Fig. 4. ASPP block adapted to knowledge distillation

E. Ablation study

In the ablation study, we performed pretraining without the ASPP block, directly bypassing connections from ResNet50 layers to projection heads before computing the contrastive loss.

To further explore the potential of multiscale feature extraction for knowledge distillation, we also implemented a variant without dilated layers for comparison. In this setting, we use feature maps from layers 1, 2, 3, and 4 of ResNet-50, each followed by class-level and feature-level projection heads. We illustrate the complete in Fig. 5.

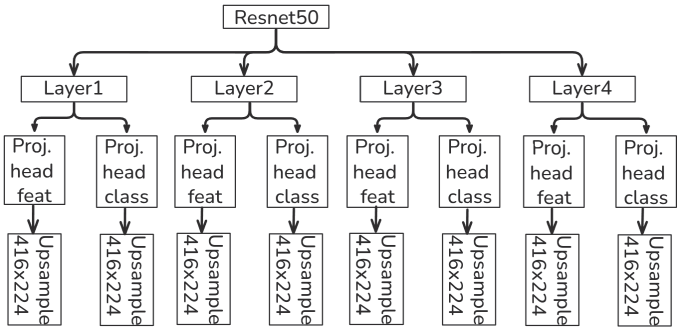


Fig. 5. Direct usage of ResNet internal layers for knowledge distillation

Both class level and feature level in the naive approach are obtained directly from internal ResNet50 layers with the subsequent projection and upsampling layers being applied.

After extraction features from layers 1, 2, 3 and 4, both class-level and feature-level projection heads are applied:

$$z^{2D}_{feat} = h^{3D}_{class}(x)$$
$$z^{2D}_{class} = h^{3D}_{feat}(x)$$

where

- h^{3D}_{class} - class-level projection head.
- h^{3D}_{feat} - feature-level projection head.
- z^{2D}_{feat} - feature-level projection head output.
- z^{2D}_{class} - class-level projection head output.
- x - input image tensor.

V. Experiments

We evaluated our method on the nuScenes dataset [2], a large-scale multimodal dataset widely used in autonomous driving research. Our reported metrics are mIoU, frequency-weighted IoU (fmIoU) as well for the pretrained model and the fine-tuned one with 1%, 5%, 10%, 25%, and 100 % of the nuScenes dataset.

For the teacher model, we adopt a pretrained ResNet50 from [7]. However, instead of original projection heads for semantic-level and class-level features, we utilize the ASPP module with 4 dilated heads followed by upsampling layers with the scale of 4.

For a student model, we use the same Sparse Residual 3D U-Net 34 (SR-UNet34) [4] followed by a projection head with channel size 64 to match the output dimensions of the image and the LiDAR branches in the common feature space.

The training process consists of two stages:

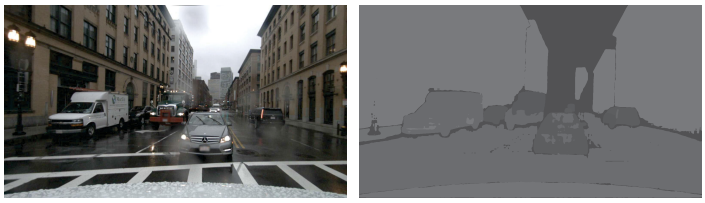
- Pretraining task with 50 epochs in total;
- Finetuning training with another 100 epochs using LiDAR data only.

Experiments were conducted using a single Nvidia A5000 gpu with 24Gb VRAM and batch sizes 11 for pretraining and 71 for funetuning accordingly.

For the pretraining task, we follow the original pipeline and perform pretraining task for 50 epochs. Having said that, we freeze the image model completely leaving the ASPP block trainable only.

In the fine-tuning stage, we train the 3D network using 1%, 5%, 10%, 25%, and 100% of the nuScenes data. In addition to the different stages of the fine-tuning scenario, we also utilize Linear Probing (LP). In the LP scenario, we train a segmentation head solely without modifying backbone weights leaving them in the frozen state. At the same time, during the fine-tuning stage, all of the weights are updated.

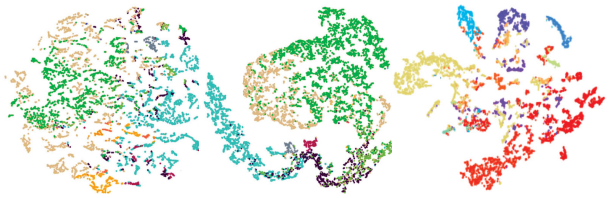
For weak label generation, we also utilize SEEM (Segment Everything All at Once) VFM [36]. Examples of generated weak labels are presented in Fig. 6 (a shows the original image and b the generated one). In other words, weak labels are semantic masks that contain class information for every pixel in the images and represent ground-truth information for the 3D branch. Each variation of color in the pictures represents superpixels, generated by VFM. Later, this semantic information is used to calculate the contrastive loss with respect to the semantic information - CL_{sup}.



(a) (b)

Fig. 6. Examples of weak labels used in CL_{sup} loss. Colors were strengthened for better visualization quality. a) represents the original picture and b) stands for clusters generated by VFM

Fig. 7 shows TSNE embedded class-level features for the original CL_{PPNCE} method taken from the work [3] (a), modified method with CL_{sup} from [7] (b), and the proposed method with multiscale feature extraction (c). Fig 7. c) demonstrates the greatest distance of semantic clusters from each other in the low-dimensional space.



(a) (b) (c)

Fig. 7. TSNE visualization for embedded class-level features obtained from h^{3D}_{sem} head in the output layer, image a) shows TSNE for CL_{PPNCE} [7], image b) shows TSNE for CL_{sup} [7], and image c) demonstrates TSNE for the proposed method (multiscale CL). Each color represents an individual semantic label

The fine-tuning results are presented in Table I. The last two rows represent the fine-tuning metrics for the multi-scale feature extraction and naive approaches, respectively.

LP column stands for Linear Probing, where we train a simple classifier with the frozen backbone. Other columns represent percentages of nuScenes data used for fine-tuning. The proposed method proves the advantages of multi-scale feature extraction from an image network for KD purposes. The results presented in the table outperform the baseline architecture. At the same time, the naive approach demonstrates the ineffectiveness of feature generation from different stages of the ResNet50 network, which can be explained by the poor feature information in the earliest ResNet layers.

The class-level results on 1% of the training data for ASPP and naive approaches are reported in Table II. The table demonstrates that the proposed approach consistently outperforms OLIVINE for all nuScenes classes. The multi-scale feature extraction block was used for both the class-level and feature-level branches, while in a second scenario (naive approach) we used identical MLP projection heads for both branches. The low quality results for the last row can be explained by the lack of useful semantic information in the early layers of the ResNet50 network. The multi-scale feature extraction row, on the contrary, demonstrates that semantic features obtained from the feature map from the last layer of the ResNet50 network, passed through dilation layers carry more useful semantic information. The room for improvement can be found in modifying classification heads to enhance results on small or infrequently observed such objects as 'bycicle', 'trailer', and 'construction vehicle'.

VI. Conclusion

We introduced a novel method for knowledge distillation that employs multiscale distillation heads connected to a ResNet50 network. The key contribution is that the proposed multiscale multi-scale feature extraction block can be connected to a wide range of architectures and seamlessly adapted to other designs. To validate this method, we compared both the naive and dilated approaches, showing a clear advantage of dilation heads over plain ResNet-50 output layers.

In addition to the multi-scale feature extraction method, we evaluated a naive variant consisting solely of ResNet-50 layers. The comparison demonstrated the superior performance of the multi-scale feature extraction architecture for multiscale knowledge distillation both on class-level, as well as feature level.

In future work, it may be worthwhile to explore KD with different layers of a student network to achieve greater performance. Moreover, it would be useful to explore feature separation technique for separating different spatial scales from each other as well as to investigate the influence of knowledge distillation on ViT-based architectures.

Acknowledgment

The research was supported by the Russian State Research grant FFZF-2025-0003.

TABLE I. Comparison of various pretraining techniques for semantic segmentation using fine-tuning (ft) and linear probing (lp). metrics are miou (%)

Initialization	**LP**	**1%**	**5%**	**10%**	**25%**	**100%**
			nuScenes-lidarseg			
Random Init	–	8.10	30.30	47.84	56.15	74.66
PointContrast [32]	21.90	32.50	–	–	–	–
DepthContrast [37]	22.10	31.70	–	–	–	–
PPKT [3]	35.90	37.80	53.74	60.25	67.14	74.52
SLidR [38]	38.80	38.30	52.49	59.84	66.91	74.79
ST-SLidR [5]	40.48	40.75	54.69	60.75	67.70	75.14
HVDistill [30]	39.50	42.70	56.60	62.90	69.30	76.60
Seal [6]	44.95	45.84	55.64	62.97	68.41	75.60
OLIVINE [7]	47.30	46.12	57.51	63.04	69.39	76.13
Ours (Multi-Scale Feature Extraction)	**49.7**	**47.8**	**58.2**	**64.1**	**70.5**	**76.7**
Ours (Basic Approach)	46.16	46.01	56.32	61.02	67.43	75.92

TABLE II. Per-class results on the nuscenes-lidarseg dataset using only 1% of the labeled data for fine-tuning using aspp block. iou scores are reported for each category. the highest scores are marked as bold

Method	barrier	bicycle	bus	car	const. veh.	motor	pedestrian	traffic cone	trailer	truck	driv. surf.	other flat	sidewalk	terrain	manmade	vegetation	**mIoU**
Random	0.0	0.0	8.1	65.0	0.1	6.6	21.0	9.0	9.3	25.8	89.5	14.8	41.7	48.7	72.4	73.3	30.3
PointCont. [32]	0.0	1.0	5.6	67.4	0.0	3.3	31.6	5.6	12.1	30.8	91.7	21.9	48.4	50.8	75.0	74.6	32.5
DepthCont. [37]	0.0	0.6	6.5	64.7	0.2	5.1	29.0	9.5	12.1	29.9	90.3	17.8	44.4	49.5	73.5	74.0	31.7
PPKT [3]	0.0	2.2	20.7	75.4	1.2	13.2	45.6	8.5	17.5	38.4	92.5	19.2	52.3	56.8	80.1	80.9	37.8
SLidR [4]	0.0	1.8	15.4	73.1	1.9	19.9	47.2	17.1	14.5	34.5	92.0	27.1	53.6	61.0	79.8	82.3	38.3
ST-SLidR [5]	0.0	2.7	16.0	74.5	3.2	25.4	50.9	20.0	17.7	40.2	92.0	30.7	54.2	61.1	80.5	82.9	40.8
OLIVINE [7]	0.0	6.7	55.6	82.7	10.3	26.8	57.8	23.1	18.8	47.3	93.9	32.4	55.8	62.4	82.0	82.3	46.1
Ours (Basic Approach)	0.0	5.4	47.4	75.2	9.3	23.2	55.1	21.6	17.3	44.9	92.2	31.1	51.2	60.4	80.4	80.7	43.5
Ours (Multi-Scale Feature Extraction)	0.0	**7.0**	**56.1**	**84.4**	**11.2**	**26.9**	**58.9**	**24.2**	**19.8**	**48.5**	**94.2**	**33.8**	**57.2**	**64.2**	**83.7**	**83.6**	**47.2**

REFERENCES

[1] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. [Online]. Available: http://arxiv.org/abs/1904.01416

[2] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuScenes: A multimodal dataset for autonomous driving. [Online]. Available: http://arxiv.org/abs/1903.11027

[3] Y.-C. Liu, Y.-K. Huang, H.-Y. Chiang, H.-T. Su, Z.-Y. Liu, C.-T. Chen, C.-Y. Tseng, and W. H. Hsu. Learning from 2D: Contrastive Pixel-to-Point Knowledge Transfer for 3D Pretraining. [Online]. Available: http://arxiv.org/abs/2104.04687

[4] C. Sautier, G. Puy, S. Gidaris, A. Boulch, A. Bursuc, and R. Marlet, "Image-to-Lidar Self-Supervised Distillation for Autonomous Driving Data," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 9881–9891. [Online]. Available: https://ieeexplore.ieee.org/document/9879430/

[5] A. Mahmoud, J. S. K. Hu, T. Kuai, A. Harakeh, L. Paull, and S. L. Waslander, "Self-Supervised Image-to-Point Distillation via Semantically Tolerant Contrastive Loss," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 7102–7110. [Online]. Available: https://ieeexplore.ieee.org/document/10204499/

[6] Y. Liu, L. Kong, J. Cen, R. Chen, W. Zhang, L. Pan, K. Chen, and Z. Liu. Segment Any Point Cloud Sequences by Distilling Vision Foundation Models. [Online]. Available: http://arxiv.org/abs/2306.09347

[7] Y. Zhang and J. Hou. Fine-grained Image-to-LiDAR Contrastive Distillation with Visual Foundation Models. [Online]. Available: http://arxiv.org/abs/2405.14271

[8] T. Feng, W. Wang, F. Ma, and Y. Yang. LSK3DNet: Towards Effective and Efficient 3D Perception with Large Sparse Kernels. [Online]. Available: http://arxiv.org/abs/2403.15173

[9] Y. Chen, J. Liu, X. Zhang, X. Qi, and J. Jia. LargeKernel3D: Scaling up Kernels in 3D Sparse CNNs. [Online]. Available: http://arxiv.org/abs/2206.10555

[10] H. Hu, F. Wang, J. Su, Y. Wang, L. Hu, W. Fang, J. Xu, and Z. Zhang. EA-LSS: Edge-aware Lift-splat-shot Framework for 3D BEV Object Detection. [Online]. Available: http://arxiv.org/abs/2303.17895

[11] T. Liang, H. Xie, K. Yu, Z. Xia, Z. Lin, Y. Wang, T. Tang, B. Wang, and Z. Tang. BEVFusion: A Simple and Robust LiDAR-Camera Fusion Framework. [Online]. Available: http://arxiv.org/abs/2205.13790

[12] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. [Online]. Available: http://arxiv.org/abs/1503.02531

[13] R. Agarwal, N. Vieillard, Y. Zhou, P. Stanczyk, S. Ramos, M. Geist, and O. Bachem. On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. [Online]. Available: http://arxiv.org/abs/2306.13649

[14] A. M. Mansourian, A. Jalali, R. Ahmadi, and S. Kasaei. Attention-guided Feature Distillation for Semantic Segmentation. [Online]. Available: http://arxiv.org/abs/2403.05451

[15] T. Liu, C. Chen, X. Yang, and W. Tan, "Rethinking Knowledge Distillation with Raw Features for Semantic Segmentation," in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, pp. 1144–1153. [Online]. Available: https://ieeexplore.ieee.org/document/10484265/

[16] J. Yuan, Q. Qi, F. Du, Z. Wang, F. Wang, and Y. Liu. FAKD: Feature Augmented Knowledge Distillation for Semantic Segmentation. [Online]. Available: http://arxiv.org/abs/2208.14143

[17] C. Yang, H. Zhou, Z. An, X. Jiang, Y. Xu, and Q. Zhang. Cross-Image Relational Knowledge Distillation for Semantic Segmentation. [Online]. Available: http://arxiv.org/abs/2204.06986

[18] T. Huang, Y. Zhang, M. Zheng, S. You, F. Wang, C. Qian, and C. Xu. Knowledge Diffusion for Distillation. [Online]. Available: http://arxiv.org/abs/2305.15712

[19] Y. Wang, W. Zhou, T. Jiang, X. Bai, and Y. Xu, "Intra-class Feature Variation Distillation for Semantic Segmentation," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Springer International Publishing, vol. 12352, pp. 346–362. [Online]. Available: https://link.springer.com/10.1007/978-3-030-58571-6_21

[20] Z. Hao, J. Guo, K. Han, H. Hu, C. Xu, and Y. Wang. VanillaKD: Revisit the Power of Vanilla Knowledge Distillation from Small Scale to Large Scale. [Online]. Available: http://arxiv.org/abs/2305.15781

[21] B. B. Sau and V. N. Balasubramanian. Deep Model Compression: Distilling Knowledge from Noisy Teachers. [Online]. Available: http://arxiv.org/abs/1610.09650

[22] L. Song, X. Gong, H. Zhou, J. Chen, Q. Zhang, D. Doermann, and J. Yuan, "Exploring the Knowledge Transferred by Response-Based Teacher-Student Distillation," in *Proceedings of the 31st ACM International Conference on Multimedia*. ACM, pp. 2704–2713. [Online]. Available: https://dl.acm.org/doi/10.1145/3581783.3612162

[23] J. Yang, S. Shi, R. Ding, Z. Wang, and X. Qi. Towards Efficient 3D Object Detection with Knowledge Distillation. [Online]. Available: http://arxiv.org/abs/2205.15156

[24] S. Wang, J. Yu, W. Li, W. Liu, X. Liu, J. Chen, and J. Zhu. Not All Voxels Are Equal: Hardness-Aware Semantic Scene Completion with Self-Distillation. [Online]. Available: http://arxiv.org/abs/2404.11958

[25] L. Zhang, R. Dong, H.-S. Tai, and K. Ma. PointDistiller: Structured Knowledge Distillation Towards Efficient and Compact 3D Detection. [Online]. Available: http://arxiv.org/abs/2205.11098

[26] Multi-aspect Knowledge Distillation with Large Language Model. [Online]. Available: https://arxiv.org/html/2501.13341v3

[27] LiDAR-LLM: Exploring the Potential of Large Language Models for 3D LiDAR Understanding. [Online]. Available: https://arxiv.org/html/2312.14074v1

[28] R. Chen, H. Luo, F. Zhao, J. Yu, Y. Jia, J. Wang, and X. Ma. Structure-Centric Robust Monocular Depth Estimation via Knowledge Distillation. [Online]. Available: http://arxiv.org/abs/2410.06982

[29] S. Kim, Y. Kim, S. Hwang, H. Jeong, and D. Kum. LabelDistill: Label-guided Cross-modal Knowledge Distillation for Camera-based 3D Object Detection. [Online]. Available: http://arxiv.org/abs/2407.10164

[30] S. Zhang, J. Deng, L. Bai, H. Li, W. Ouyang, and Y. Zhang. HVDistill: Transferring Knowledge from Images to Point Clouds via Unsupervised Hybrid-View Distillation. [Online]. Available: http://arxiv.org/abs/2403.11817

[31] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[32] S. Xie, J. Gu, D. Guo, C. R. Qi, L. J. Guibas, and O. Litany. PointContrast: Unsupervised Pre-training for 3D Point Cloud Understanding. [Online]. Available: http://arxiv.org/abs/2007.10985

[33] X. Chen, H. Fan, R. Girshick, and K. He. Improved Baselines with Momentum Contrastive Learning. [Online]. Available: http://arxiv.org/abs/2003.04297

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. [Online]. Available: https://ieeexplore.ieee.org/document/5206848

[35] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. [Online]. Available: http://arxiv.org/abs/1706.05587

[36] X. Zou, J. Yang, H. Zhang, F. Li, L. Li, J. Wang, L. Wang, J. Gao, and Y. J. Lee. Segment Everything Everywhere All at Once. [Online]. Available: http://arxiv.org/abs/2304.06718

[37] P. C. Chhipa, R. Upadhyay, R. Saini, L. Lindqvist, R. Nordenskjold, S. Uchida, and M. Liwicki. Depth Contrast: Self-Supervised Pretraining on 3DPM Images for Mining Material Classification. [Online]. Available: http://arxiv.org/abs/2210.10633

[38] A. Mahmoud, A. Harakeh, and S. Waslander. Image-to-Lidar Relational Distillation for Autonomous Driving Data. [Online]. Available: http://arxiv.org/abs/2409.00845