

# Bidirectional Knowledge Reconfiguration for Lightweight Point Cloud Analysis

Peipei Li, Xing Cui, Yibo Hu, Man Zhang, Ting Yao, *Senior Member, IEEE*, and Tao Mei, *Fellow, IEEE*

**Abstract**—Point cloud analysis faces computational system overhead, limiting its application on mobile or edge devices. Directly employing small models may result in a significant drop in performance since it is difficult for a small model to adequately capture local structure and global shape information simultaneously, which are essential clues for point cloud analysis. This paper explores feature distillation for lightweight point cloud models. To mitigate the semantic gap between the lightweight student and the cumbersome teacher, we propose bidirectional knowledge reconfiguration (BKR) to distill informative contextual knowledge from the teacher to the student. Specifically, a top-down knowledge reconfiguration and a bottom-up knowledge reconfiguration are developed to inherit diverse local structure information and consistent global shape knowledge from the teacher, respectively. However, due to the farthest point sampling in most point cloud models, the intermediate features between teacher and student are misaligned, deteriorating the feature distillation performance. To eliminate it, we propose a feature mover’s distance (FMD) loss based on optimal transportation, which can measure the distance between unordered point cloud features effectively. Extensive experiments conducted on shape classification, part segmentation, and semantic segmentation benchmarks demonstrate the universality and superiority of our method.

**Index Terms**—3D Point Cloud Analysis, Feature Distillation, Earth Mover’s Distance.

## I. INTRODUCTION

WITH the popularity of 3D sensing devices, 3D data are widely used in many applications, such as autonomous driving, robotics, and virtual reality. Among all kinds of 3D data forms, point clouds are considered a simple but efficient representation. To process irregular, unordered, and unstructured point clouds, early works transform point clouds into regular voxels [1] or multiview images [2]. However, these methods lose rich geometric structure. Since the success of PointNet [3], processing point clouds directly has been the dominant solution for 3D point cloud analysis [4], [5]. The subsequent methods, *e.g.*, PointNet++ [6], KCNet [7] and DensePoint [8] have achieved significant improvements in point cloud classification and segmentation tasks. These methods can be divided into three categories. 1) MLP-based

methods [6], [9] treat each point independently and map points into high-dimensional features. 2) CNN-based methods [10], [11] design convolution kernels to capture geometric topologies. 3) Transformer-based methods [12], [13] take advantage of a transformer to extract long-range information.

Despite these advancements, there are still some practical challenges. One is the computational overhead of the system. With the need for applications on mobile or edge devices, point cloud analysis with a small model size, light computation cost, and high performance has attracted much attention. However, current point cloud analysis methods often depend on cumbersome models with expensive computations. For example, PointTransformer [12] requires more than 18.6 GFLOPs on the ModelNet40 dataset when 1024 points are sampled as input. Another key challenge is the irregularity of point clouds, making it difficult to represent discriminative semantic features for elusive shapes. Some methods [3], [7] learn directly from irregular point clouds and sacrifice complexity for effectiveness. Other methods [6], [8] attempt to make full use of the contextual information, including both the global shape and the local structure representations.

To address the above challenges, in this paper, we investigate lightweight point cloud analysis from the perspective of feature distillation, where the performance of a lightweight student network is improved by transferring informative knowledge from the intermediate features of a cumbersome teacher network. As illustrated in Table II, conventional knowledge distillation algorithms show limited performance in point clouds since the diverse local structure and global shape information of the point cloud are not fully explored during distillation. To solve this problem, we propose a novel bidirectional knowledge reconfiguration (BKR) mechanism for point cloud feature distillation. Specifically, a top-down knowledge reconfiguration and a bottom-up knowledge reconfiguration are designed, where the former is developed for inheriting diverse local structure information from the teacher, and the latter is employed to absorb high-level global shape knowledge from the teacher. In addition, we also design a residual connection to encourage distilling knowledge from the same level. Therefore, BKR mitigates the semantic gap between lightweight students and cumbersome teachers. Additionally, BKR inherits contextual knowledge from the teacher to all the scales of the student. In this way, each semantic level of the student network can simultaneously learn contextual information from the teacher network with both the local structure and the global shape knowledge.

Peipei Li, Xing Cui and Man Zhang are with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China. (e-mail: lipeipei@bupt.edu.cn; cuixing@bupt.edu.cn; zhangman@bupt.edu.cn).

Ting Yao and Tao Mei are with HiDream.ai, Beijing, China. (e-mail: tingyao.ustc@gmail.com; tmei@live.com).

Corresponding author: Yibo Hu. (e-mail: huyibo871079699@gmail.com).

This work was done during Xing Cui’s internship and Yibo Hu’s working in JD AI Research.

Furthermore, 3D point clouds are discrete and unordered. Generally, farthest point sampling (FPS) is employed in most point cloud analysis models [6], [10], [12] to reduce the resolution of the point cloud. However, the randomness of FPS results in a misalignment between the intermediate features of the teacher and student, which may further lead to inferior or even destroyed distillation performance. Inspired by optimal transportation theory [14], we propose feature mover’s distance (FMD) to measure the discrepancy between misaligned teacher and student features. Specifically, to exploit the local structure information, we divide the transportation task into several subproblems where each subproblem focuses on a local area. We further propose a distance-based transportation strategy that approximates the least-expensive transportation flow to simplify the solving procedure of the transportation problem. Extensive experiments are conducted on several benchmarks, demonstrating the effectiveness and the universality of the proposed method. To summarize, our contributions are fourfold:

- We design a new feature distillation method for lightweight point cloud analysis: a universal knowledge transfer framework for various point cloud models.
- Bidirectional knowledge reconfiguration (BKR) is proposed to transfer both the low-level structure knowledge and the high-level shape information from the teacher to all the semantic levels of the student.
- Since there exists a potential position inconsistency in point cloud features caused by the point sampling operation, the feature mover’s distance (FMD) is designed to align the features between the teacher and student.
- Our method significantly outperforms the previous distillation strategies on point cloud analysis, demonstrating the effectiveness and universality of our framework.

## II. RELATED WORK

In this section, we briefly review existing works related to our method, including point-based classification and segmentation, model compression via knowledge distillation and earth mover’s distance.

### A. Point-based Classification and Segmentation

Methods on point-based classification and segmentation can be divided into three categories: MLP-based [3], [6], [9], CNN-based [10], [11], [15] and transformer-based [12], [15], [16]. PointNet [3] pioneers MLP-based point cloud classification and segmentation. It utilizes MLP to map points to high-dimensional features and aggregates global features through max pooling, thereby extracting permutation invariant features. However, it fails to capture local structures and ignores fine-grained patterns. To solve this problem, PointNet++ [6] designs a hierarchical structure to combine features from multiple scales. Although PointNet++ achieves better performance, it still has limitations in information extraction due to the asymmetric structure. To counter this, PointMixer [9] proposes a universal set operator to build a symmetric architecture.

Another network, RandLA-Net [17], improves the efficiency of point cloud processing by using random point sampling instead of point selection.

However, those MLP-based methods only process points individually, ignoring the geometry structure information. To counter this, some researchers intend to design convolution operators on point clouds. DGCNN [18] recovers the topological information of the point cloud via a graph and uses EdgeConv to capture features over a long range. PointConv [10] focuses on nonuniform sampling point clouds, a discrete approximation of a continuous convolution. To learn relationships in point clouds, DensePoint [8] employs relation-shape convolution and builds a dense connection structure to extract dense contextual representations. In contrast, AdaptConv [19] explores an adaptive kernel generated from a pair of points.

More recently, transformer-based methods have been proposed for effective point cloud feature learning. Point transformer [12] explores how to extract long-distance relationships in large scenes by developing a self-attention layer for point cloud processing. Another network, DTNet [13], aggregates pointwise and channelwise self-attention models simultaneously for better feature representation. Although the above methods show good performance, they all ignore the memory and computational costs.

### B. Model Compression via Knowledge Distillation

Knowledge distillation [20] is a model compression technique that has been widely applied in image processing, such as image classification [21]–[23], face analysis [24], [25], semantic segmentation [26], [27] and object detection [28], [29]. Existing KD methods can be categorized into different categories [30]. Based on the number of levels where the distillation occurs, we divide knowledge distillation into single-level methods and multi-level methods.

For single-level methods, the model distills knowledge only between certain layers of the network. Among them, KD [20] minimizes the KL divergence between the last logit outputs of the teacher and the student networks. Furthermore, DKD [31] improves the flexibility of logit distillation by formulating distillation loss into a target class term and a non-target term. Recently, many works have focused on optimizing the distillation process via intermediate representations. For example, FitNet [32] utilizes intermediate features as hints to train a deeper and thinner student. NST [33] reviews the distributions of neuron selectivity and matches the distribution between the teacher and student. SimKD [34] designs a simple soft target distillation technique and reuses the classifier layer to narrow the performance gap. PEFD [35] observes the positive effect of the projector in feature distillation. Therefore, an ensemble of projectors is introduced to improve the performance.

For multi-level methods, knowledge is distilled for multiple layers of the network. AT [36] designs several methods for transferring attention maps between the teacher and student. SP [37] distills knowledge by preserving the pairwise similarities, which utilizes the pairwise activation similarities within each minibatch to supervise the distillation process. Recent works, ReviewKD [38] and SemCKD [39], further utilize the

intermediate features of the teacher model by exploring multi-layer knowledge. Contrary to the aforementioned approaches that are designed for image processing, we introduce knowledge distillation to point cloud analysis aiming at transferring the diverse local structure information and the global shape knowledge from the intermediate point cloud features of a cumbersome teacher to a lightweight student.

### C. Earth Mover's Distance

Earth mover's distance (EMD) [40] is proposed to measure the distance between two sets of weighted objects or probability distributions. It has the form of an optimal transportation problem and is defined as the transportation cost under the least-expensive transportation flow. Specifically, let  $F_r = \{(F_r^1, s_1), \dots, (F_r^N, s_N)\}$  be a set of sources consisting of  $N$  pairs, where  $F_r^i$  and  $s_i$  denote the  $i$ -th source feature and its corresponding weight, respectively. Let  $F_t = \{(F_t^1, t_1), \dots, (F_t^N, t_N)\}$  be a set of destinations, where  $F_t^j$  and  $t_j$  denote the  $j$ -th target feature and its corresponding weight, respectively. The ground distance between  $F_r^i$  and  $F_t^j$  is denoted by  $d_{i,j}$ . The goal of the transportation problem is to find the least-expensive flow  $\Pi = (\pi_{ij}) \in \mathbb{R}^{N \times N}$  from  $F_r$  to  $F_t$ . The transportation problem can be formulated as a linear programming problem:

$$\begin{aligned} EMD(F_r, F_t) &= \min_{\Pi \geq 0} \sum_{i,j} d_{i,j} \pi_{i,j}, \\ \text{subject to } &\sum_j \pi_{i,j} = s_i, \quad i \in [1, N], \\ &\sum_i \pi_{i,j} = t_j, \quad j \in [1, N]. \end{aligned} \quad (1)$$

Then, the least-expensive transportation flow can be achieved with the help of linear programming algorithms, such as the Sinkhorn algorithm [41].

Recently, EMD has been widely used in image processing [42]–[44]. For example, DeepEMD [45] computes the EMD between dense image features to represent the image distance. DeepFace-EMD [42] reranks face identification results with EMD to improve out-of-distribution generalization. DensePCR [46] predicts the low-resolution point cloud via the EMD loss to measure the consistency of two point sets.

Despite its effectiveness, EMD is a computationally intensive formulation that requires considerable time and memory. To alleviate this problem, EXSinkhorn [47] adds an entropic regularization [41] and adaptively doubles the regularization parameter. SW [48] and its variants [49] characterize high-dimensional probability distributions into one-dimensional space to accelerate the calculation. In addition, some works [49]–[51] explore minibatch solutions to reduce the memory and computational cost. BoMb-OT [49] proposes optimal coupling to consider the relationship between minibatches, which approximates the original transportation strategy and constructs a good global mapping. Recently, m-POT [51] utilized partial optimal transportation to solve the misspecified mapping problem.

However, these approaches still need to solve complex linear programming problems to find the optimal transportation

flow. The computational cost is  $O(\max(m, n)^3)$ , which is untenable for the gradient descent-based method. REMD [14], [52] solves this problem by relaxing the optimal transportation problem and removing one of the two constraints:

$$R_{F_r}(F_r, F_t) = \min_{\Pi \geq 0} \sum_{i,j} d_{i,j} \pi_{i,j} \quad \text{s.t.} \quad \sum_j \pi_{i,j} = s_i, \quad (2)$$

$$R_{F_t}(F_r, F_t) = \min_{\Pi \geq 0} \sum_{i,j} d_{i,j} \pi_{i,j} \quad \text{s.t.} \quad \sum_i \pi_{i,j} = t_j. \quad (3)$$

Thus, REMD can be formulated as:

$$\begin{aligned} L_{REMD} &= REMD(F_r, F_t) \\ &= \max(R_{F_r}(F_r, F_t), R_{F_t}(F_r, F_t)) \\ &= \max\left(\sum_i t_i \min_j d_{i,j}, \sum_j s_j \min_i d_{i,j}\right). \end{aligned} \quad (4)$$

Although REMD has shown satisfactory performance in natural language processing [14] and image processing [52], it only considers optimal transportation of feature weights globally, failing to transfer local structure information in sparse point clouds effectively. Therefore, we propose a feature mover's distance (FMD) to explore the global shape information as well as the rich local structure information.

## III. APPROACH

### A. Preliminary

We denote an input point cloud with  $N$  points as  $X \in \mathbb{R}^{N \times d_{in}}$ , where  $d_{in}$  is the input dimension. The corresponding positions are defined as  $P \in \mathbb{R}^{N \times 3}$ . Usually,  $X$  only contains normalized 3D coordinates, *i.e.*,  $X = P$ , but it can also be combined with additional attributes, such as surface normal and color. Given an input  $X$  and a lightweight student network  $\mathcal{S}$ . The output  $Y_s$  can be formulated as:

$$Y_s = \mathcal{S}(X) = \mathcal{S}_c \circ \mathcal{S}_L \circ \dots \circ \mathcal{S}_2 \circ \mathcal{S}_1(X), \quad (5)$$

where  $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_L$  are the sequential blocks of the student.  $\mathcal{S}_c$  represents the classifier in the classification task or the decoder in the segmentation task.  $\circ$  is a nesting function, where  $g \circ f(\cdot) = g(f(\cdot))$ . We denote the intermediate features of the student as  $\{F_{s,1}, F_{s,2}, \dots, F_{s,L}\}$ .  $F_{s,l}$  is calculated by:

$$F_{s,l} = \mathcal{S}_l \circ \dots \circ \mathcal{S}_2 \circ \mathcal{S}_1(X). \quad (6)$$

The teacher network  $T$  shares a similar process. We denote the intermediate features of the teacher as  $\{F_{t,1}, F_{t,2}, \dots, F_{t,L}\}$ .

### B. Overall Framework

In this paper, we propose bidirectional knowledge reconfiguration (BKR), a novel feature distillation mechanism, for lightweight point cloud analysis. The overall framework is illustrated in Fig. 1. BKR consists of top-down knowledge reconfiguration (TDKR), bottom-up knowledge reconfiguration (BUKR) and residual connection (RES), aiming at alleviating

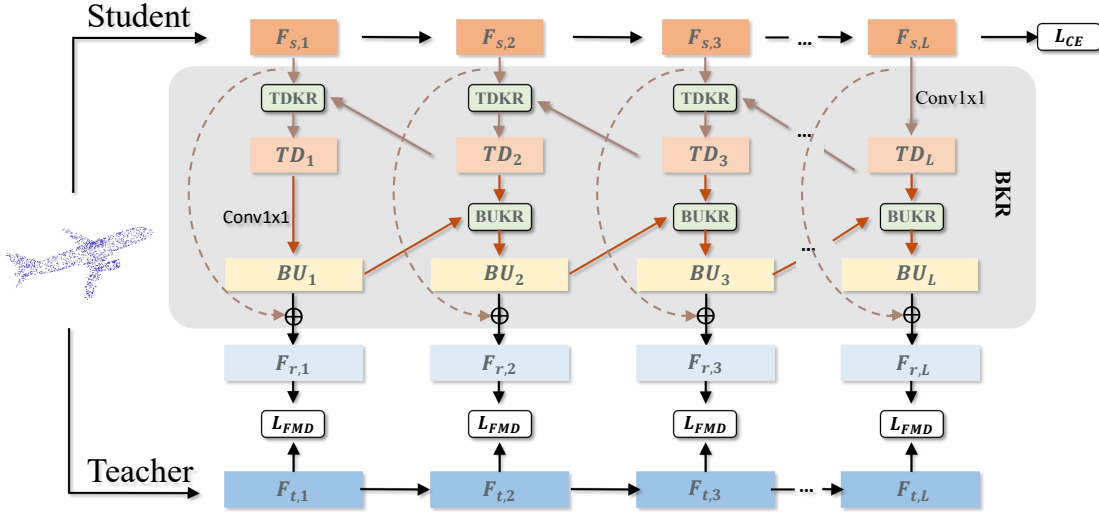


Fig. 1: The overall framework of our method. Bidirectional knowledge reconfiguration (BKR) contains top-down knowledge reconfiguration (TDKR), bottom-up knowledge reconfiguration (BUKR), and residual connection.

the semantic gap between teacher and student, as well as distilling the contextual knowledge from the teacher to all the semantic levels of the student. However, we observe that position inconsistency between the corresponding teacher and student features, caused by the random sampling operation, is one of the main factors affecting the performance of point cloud feature distillation. To solve this problem, we further design a feature mover’s distance (FMD), which can measure the discrepancy between misaligned student features and teacher features effectively.

### C. Bidirectional Knowledge Reconfiguration

Multi-level distillation is widely employed in feature distillation and shows satisfactory performance [36], [37], [53], which usually transfers the same-level knowledge between teacher and student. However, in point cloud analysis, neglecting cross-level knowledge may lead to a loss of rich 3D geometric information and is not conducive to grasping the diverse shape information formed by point clouds [8]. Inspired by multiscale feature learning [54], [55], we propose bidirectional knowledge reconfiguration for point cloud feature distillation, imposing multi-level and multiscale contextual knowledge from the teacher to all the semantic levels of the student hierarchically. We divide layers with the same resolution into a group and view them as a level. In each level, the feature of the last layer is employed to distill the knowledge. Specifically, a top-down knowledge reconfiguration is first employed to merge the information from top to bottom of the student so that the low-level structure knowledge of the teacher can be spread to deep student layers. In addition to low-level structure knowledge, features at high levels represent global knowledge, which is essential for perceiving the overall shape of the point cloud. To further inherit the high-level shape knowledge from the teacher, we perform a bottom-up knowledge reconfiguration on the features produced by the top-down knowledge reconfiguration. Finally, the reconfigured

feature and the original student feature are fused via a residual connection to better inherent information from the same level.

1) *Top-down Knowledge Reconfiguration (TDKR)*: As shown in Fig. 1, we denote  $\{TD_1, TD_2, \dots, TD_L\}$  as the reconfigured features of TDKR, where  $TD_l$  is formulated as:

$$TD_l = \begin{cases} TDKR(TD_{l+1}, F_{s,l}), & l = 1, \dots, L-1 \\ Conv1 \times 1(F_{s,l}), & l = L \end{cases} \quad (7)$$

Fig. 2(a) presents the building block of TDKR. Taking the  $l$ -th level of the student as an example, with the reconfigured feature  $TD_{l+1} \in \mathbb{R}^{n' \times d}$  from the  $(l+1)$ -th level, we first upsample the feature resolution to the same size as the corresponding teacher feature. Similar to [6], we obtain the upsampled feature by interpolating feature values of  $(l+1)$ -th level points at coordinates of the  $l$ -th level points. The output is denoted as  $TD_{l+1}^\uparrow \in \mathbb{R}^{n \times d}$ :

$$TD_{l+1}^\uparrow = Upsample(TD_{l+1}). \quad (8)$$

Specifically, if the feature is global, we simply use repetition as the upsampling operation. Additionally, the original student feature  $F_{s,l} \in \mathbb{R}^{n \times d}$  undergoes a  $1 \times 1$  convolution to match the dimension of  $TD_{l+1}^\uparrow$ , which is termed  $F'_{s,l} \in \mathbb{R}^{n \times d}$ :

$$F'_{s,l} = Conv1 \times 1(F_{s,l}). \quad (9)$$

Inspired by [56], [57], we employ a gate mechanism to control the information flows from different features. Specifically, we concatenate  $TD_{l+1}^\uparrow$  and  $F'_{s,l}$  as  $F_{td,l} \in \mathbb{R}^{n \times 2d}$  and employ a  $1 \times 1$  convolution with a sigmoid function to generate the weight  $w_{td,l} \in \mathbb{R}^{n \times 2}$ . Then, the weight is split into two gates  $g_{td,l}^1 \in \mathbb{R}^{n \times 1}$  and  $g_{td,l}^2 \in \mathbb{R}^{n \times 1}$ . TDKR is calculated as:

$$TDKR(TD_{l+1}, F_{s,l}) = [g_{td,l}^1] \cdot F'_{s,l} + [g_{td,l}^2] \cdot TD_{l+1}^\uparrow, \quad (10)$$

where  $[g_{td,l}^1] \in \mathbb{R}^{n \times d}$  and  $[g_{td,l}^2] \in \mathbb{R}^{n \times d}$  are the repetitions of  $g_{td,l}^1$  and  $g_{td,l}^2$   $d$  times, respectively. In this way, the weights

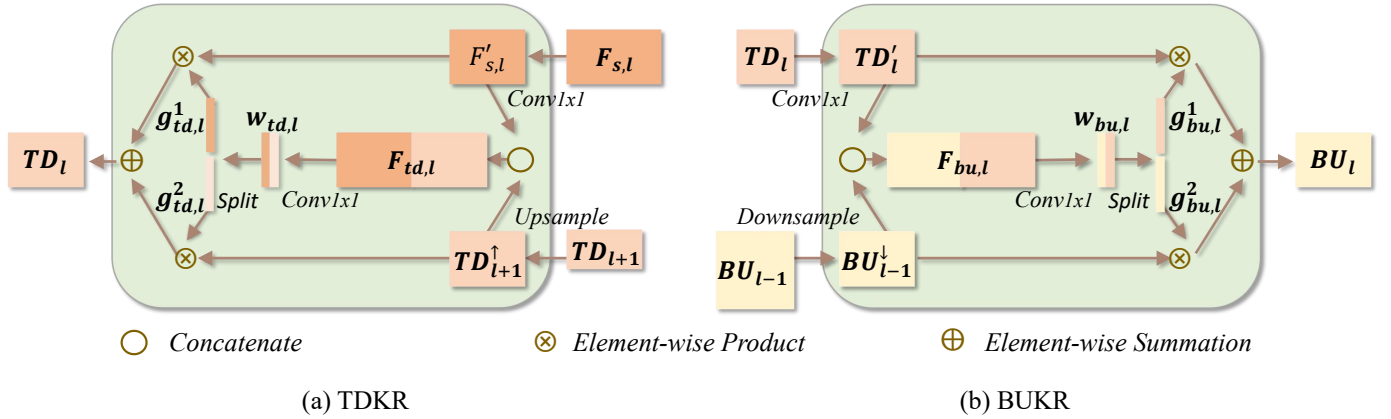


Fig. 2: The architectures of (a) top-down knowledge reconfiguration (TDKR) and (b) bottom-up knowledge reconfiguration (BUKR). The size of the feature blocks represents the relative shape of the features.

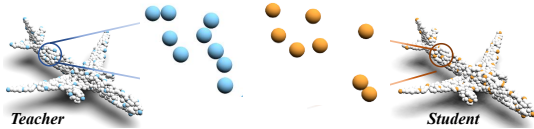


Fig. 3: Illustration of position inconsistency. The blue dots represent teacher features. The orange dots are student features.

are generated dynamically based on the input features. Thus, the information flows from different levels that carry diverse knowledge can be reconfigured adaptively.

2) *Bottom-up Knowledge Reconfiguration (BUKR)*: As illustrated in Fig. 2 (b), the structure of BUKR is similar to that of TDKR but different in detail: BUKR performs downsampling on low-level features for feature fusion, while TDKR performs upsampling on high-level features. We define  $\{BU_1, BU_2, \dots, BU_L\}$  as the outputs of BUKR, where  $BU_l$  is formulated as:

$$BU_l = \begin{cases} BUKR(BU_{l-1}, TD_l), & l = 2, \dots, L \\ Conv1 \times 1(TD_l), & l = 1 \end{cases}. \quad (11)$$

Specifically,  $BU_{l-1}$  is first downsampled to match the resolution, and a  $1 \times 1$  convolution is performed on  $TD_l$  to match the dimension:

$$BU_{l-1}^\downarrow = \text{Downsample}(BU_{l-1}), \quad (12)$$

$$TD_l' = Conv1 \times 1(TD_l). \quad (13)$$

Then, we calculate the weight  $w_{bu,l} \in \mathbb{R}^{n \times 2}$  from the concatenation of  $BU_{l-1}^\downarrow$  and  $TD_l'$  in the same way as TDKR.  $w_{bu,l}$  is further split into  $g_{bu,l}^1 \in \mathbb{R}^{n \times 1}$  and  $g_{bu,l}^2 \in \mathbb{R}^{n \times 1}$  as two gates. Finally, the output of BUKR can be written as:

$$BUKR(BU_{l-1}, TD_l) = [g_{bu,l}^1] \cdot TD_l' + [g_{bu,l}^2] \cdot BU_{l-1}^\downarrow, \quad (14)$$

where the notations are similar to TDKR.

3) *Residual Connection (RES)*: TDKR and BUKR can transfer cross-level information, while rich knowledge at the same level might be ignored. To effectively inherit the same level of knowledge from the teacher, a residual connection is employed to obtain the reconfigured feature  $F_{r,l}$ :

$$F_{r,l} = BU_l + F_{s,l}. \quad (15)$$

#### D. Feature Mover's Distance

As shown in Fig. 3, The randomness of farthest point sampling (FPS) makes the position and order of points different between the teacher and the student. Taking the  $l$ -th level as an example, after FPS, the point positions of the student  $P_{s,l} \in \mathbb{R}^{N \times 3}$  are not equal to the point positions of the teacher  $P_{t,l} \in \mathbb{R}^{N \times 3}$ , leading to feature misalignment between the teacher and student. We present more analysis in Section IV-D1 to show the misalignment. To this end, it is essential to align the point positions of the intermediate features before distilling the knowledge from teacher to student.

Inspired by the optimal transportation theory, we propose the feature mover's distance (FMD) to align the point positions of the features between the teacher and student. Specifically, we first divide the original optimal transportation problem into  $N$  subproblems to leverage local structure information. Denote  $F_{r,l} = \{F_{r,l}^1, \dots, F_{r,l}^N\}$  as the reconfigured feature of the student and  $P_{s,l} = \{P_{s,l}^1, \dots, P_{s,l}^N\}$  as its corresponding positions. We divide the student feature into  $N$  subsets. Therefore, each subset contains one element, i.e.,  $\hat{F}_{r,l}^i = \{F_{r,l}^j \mid j = i\}$ . Similarly, let  $F_{t,l}$  be the feature of the teacher and  $P_{t,l} = \{P_{t,l}^1, \dots, P_{t,l}^N\}$  be its corresponding positions. As transporting products to neighboring destinations is an approximation of the least-expensive transportation strategy [58], [59], we define the teacher feature subset  $\hat{F}_{t,l}^i = \{F_{t,l}^j \mid j \in N_{P_{s,l}}^i(P_{t,l})\}$ .  $N_{P_{s,l}}^i(P_{t,l})$  is the index of the  $k$  nearest neighbors of student position  $P_{s,l}^i$  in teacher position set  $P_{t,l}$ .

Finally, we define FMD as the feature discrepancy under a distance-based transportation strategy. Specifically, we propose

TABLE I: Resource usage for different models. T and S represent the original teacher and compressed student model.

		Shape Classification		Object Part Segmentation		Semantic Segmentation	
		MAdds(M)	Params(M)	MAdds(M)	Params(M)	MAdds(M)	Params(M)
PN++	T	868	1.48	1154	1.41	1042	0.97
	S	19	0.03	35	0.03	62	0.02
DGC	T	2449	1.81	4538	1.46	6181	0.98
	S	45	0.03	1149	0.82	127	0.02
PConv	T	1171	19.57	10012	10.12	9990	10.11
	S	41	0.31	396	0.17	247	0.17
PT	T	18600	9.58	37840	19.40	/	/
	S	320	0.15	740	1.92	/	/

a distance-based transportation strategy  $\Pi^l = (\pi^l_{i,j})$  to approximate the least-expensive transportation strategy. Similar to [60], we determine the transportation strategy based on the ground distance  $d_{i,j}$  and use the normalized Gaussian radial basis function to calculate the  $\pi^l_{i,j}$  of the  $l$ -th level features:

$$\pi^l_{i,j} = \frac{e^{-d_{i,j}^2/2\tau^2}}{\sum_{h \in N_{P_{s,l}^i}(P_{t,l})} e^{-d_{i,h}^2/2\tau^2}}, \quad (16)$$

where  $d_{i,j} = \left\| P_{s,l}^i - P_{t,l}^j \right\|_2$  and  $\tau$  is a temperature parameter. FMD is calculated as follows:

$$L_{FMD}^l = FMD(F_{r,l}, F_{t,l}) = \sum_{i=1}^N s_{i,l} \left\| F_{r,l}^i - \sum_{j \in N_{P_{s,l}^i}(P_{t,l})} \pi^l_{i,j} F_{t,l}^j \right\|_2, \quad (17)$$

where  $F_{r,l}$  and  $F_{t,l}$  are the reconfigured student feature and the teacher feature of the  $l$ -th level, respectively. Compared to REMD in Eq. (4) that only considers the global nearest destination, FMD takes  $k$  nearest local neighbors into account, which transfers local structure information of different levels effectively and makes the measurement more robust.

Inspired by average pooling correlation (APC) [42], we formulate  $s_{i,l}$  as follows:

$$s_{i,l} = \max(0, \langle F_{r,l}^i, \frac{\sum_j F_{t,l}^j}{N} \rangle). \quad (18)$$

During the training process, we utilize both the original cross-entropy loss  $L_{CE}$  and the FMD loss  $L_{FMD}^l$ . The total loss function is:

$$L = L_{CE} + \lambda \sum_{l=1}^L L_{FMD}^l, \quad (19)$$

where  $\lambda$  is a trade-off hyperparameter.

#### IV. EXPERIMENTS

We evaluate the effectiveness of our method on ModelNet40 [61] for point cloud classification, ShapeNetPart [62] for object part segmentation and S3DIS [63] for point cloud semantic segmentation. Since there are few studies on point cloud distillation, we select the Feature-L2 (F-L2) as our baseline, which is a classical feature distillation method in image

TABLE II: Results on shape classification.

	PN++	(1/8)PN++	F-L2	SP	FitNet	NST	AT	KD	OFD	DKD	PEFD	Ours
OA	92.54	88.48	88.03	88.31	88.57	88.61	89.08	89.02	89.23	89.18	89.28	<b>90.28</b>
mAcc	90.17	80.61	79.69	80.03	82.17	82.59	82.79	82.94	83.25	83.08	83.27	<b>84.55</b>
	DGC	(1/8)DGC	F-L2	SP	FitNet	NST	AT	KD	OFD	DKD	PEFD	Ours
OA	92.26	82.65	84.03	84.62	83.72	84.22	84.24	83.60	85.04	83.63	85.27	<b>86.46</b>
mAcc	89.42	68.09	71.19	72.25	71.13	71.55	71.95	69.76	72.40	69.89	72.71	<b>74.77</b>
	PConv	(1/8)PConv	F-L2	SP	FitNet	NST	AT	KD	OFD	DKD	PEFD	Ours
OA	92.34	74.53	73.23	74.44	74.07	74.38	76.38	76.74	77.32	76.96	78.13	<b>83.73</b>
mAcc	89.15	62.38	60.43	60.57	60.75	61.01	62.74	63.81	65.15	64.03	67.51	<b>72.16</b>
	PT	(1/8)PT	F-L2	SP	FitNet	NST	AT	KD	OFD	DKD	PEFD	Ours
OA	92.31	87.18	86.57	86.69	87.42	87.66	87.82	87.90	86.01	87.88	87.98	<b>88.50</b>
mAcc	89.92	82.58	80.83	81.49	82.63	82.99	83.47	83.00	80.73	83.11	83.89	<b>84.25</b>

processing. In F-L2, all the intermediate features of the student are first transformed to match the size of the corresponding teacher features. Then,  $L_2$  loss is employed as the distillation objective. We choose widely used distillation methods as competitors, including KD [20], FitNet [32], NST [33], AT [36], SP [37], OFD [64], DKD [31] and PEFD [35]. Four classical models are chosen as the backbones, including the MLP-based model PointNet++ (PN++) [6], graph-based model DGCNN (DGC) [18], CNN-based model PointConv (PConv) [10] and transformer-based model PointTransformer (PT) [12]. We treat the original model as the teacher and reduce the width to 1/8 as the student, which is marked by a prefix of (1/8). Table I lists the number of parameters (Params) and the multiadds (MAdds) of these models.

For PointNet++ [6] and PointConv [10], in addition to the point positions, we also employ the surface normals as the additional input. For all the experiments, we set the data augmentations and the training hyperparameters the same as the open source codes<sup>1,2,3,4</sup>. For the tradeoff parameter  $\lambda$ , we conduct cross-validation on the ModelNet40 dataset and find that  $\lambda = 0.1$  achieves the best results for classification and  $\lambda = 0.01$  achieves the best results for segmentation. For the competitors, we also conduct the same cross-validation experiment to choose the tradeoff parameter. In addition, we choose  $k = 5$  as the number of neighbors in FMD. Code is available at <https://github.com/cuixing100876/BKR>.

#### A. Shape Classification

1) *Data and Metrics*: The ModelNet40 dataset [61] consists of 12,311 meshed CAD models from 40 categories, which is split into 9,843 models for training and 2,468 models for testing. We follow the data preparation of [61] and employ the mean accuracy within each category (mAcc) and the overall accuracy (OA) as the evaluation metrics.

2) *Results*: As shown in Table II, when dealing with models that involve sampling operations, such as PointNet++, PointConv, and PointTransformer, the utilization of F-L2 may have a negative impact on the performance of student model. This is because directly transferring intermediate feature knowledge without reconfiguration and alignment may

<sup>1</sup>[https://github.com/yanx27/Pointnet\\_Pointnet2\\_pytorch](https://github.com/yanx27/Pointnet_Pointnet2_pytorch)

<sup>2</sup><https://github.com/AnTao97/dgcnn.pytorch>

<sup>3</sup>[https://github.com/DylanWusee/pointconv\\_pytorch](https://github.com/DylanWusee/pointconv_pytorch)

<sup>4</sup><https://github.com/qq456cvb/Point-Transformers>



TABLE III: Results on object part segmentation.

	PN++	(1/8)PN++	F-L2	SP	FitNet	NST	AT	KD	OFD	DKD	PEFD	Ours
ins.mIoU	85.21	76.29	75.92	75.99	76.34	76.50	76.71	76.82	76.94	76.84	77.25	<b>79.22</b>
cat.mIoU	81.74	58.05	57.93	57.97	58.03	58.37	58.25	58.32	58.60	58.37	58.43	<b>59.84</b>
	DGC	(1/8)DGC	F-L2	SP	FitNet	NST	AT	KD	OFD	DKD	PEFD	Ours
ins.mIoU	84.86	72.39	73.02	73.57	72.95	73.07	73.44	72.54	74.24	72.95	74.64	<b>76.36</b>
cat.mIoU	82.23	48.75	51.55	55.56	51.24	51.63	55.32	50.64	55.85	50.71	55.98	<b>57.53</b>
	PCConv	(1/8)PCConv	F-L2	SP	FitNet	NST	AT	KD	OFD	DKD	PEFD	Ours
ins.mIoU	85.18	78.95	76.36	76.50	77.01	78.37	79.65	79.91	80.02	79.90	79.93	<b>80.22</b>
cat.mIoU	81.95	59.98	55.83	55.85	56.73	59.87	60.90	61.47	61.67	61.55	62.01	<b>63.38</b>
	PT	(1/8)PT	F-L2	SP	FitNet	NST	AT	KD	OFD	DKD	PEFD	Ours
ins.mIoU	83.75	74.76	73.57	75.06	75.14	75.30	75.50	75.90	74.32	75.76	75.84	<b>77.83</b>
cat.mIoU	79.95	60.76	58.47	64.17	64.56	64.61	65.00	65.56	58.51	65.44	65.65	<b>66.15</b>

potentially result in performance degradation. Our method (BKR+FMD) outperforms other distillation methods on all four backbones. Specifically, for the PointNet++ model, our method outperforms DKD by 1.10% and 1.47% in OA and mAcc, respectively. Moreover, our method also outperforms PEFD by large margins, *i.e.*, 1.00% and 1.28% in OA and mAcc, respectively, which demonstrates the effectiveness of BKR and FMD. Besides, our method improves the mACC of the student by 3.94%, 6.68%, 9.78% and 1.67% and the OA by 1.8%, 3.81%, 9.2% and 1.32% with PointNet++, DGCNN, PointConv and PointTransformer, respectively. The improvement demonstrates that the proposed BKR and FMD can benefit the knowledge transfer procedure in various kinds of point cloud models, demonstrating the universality of our method.

### B. Object Part Segmentation

1) *Data and Metrics*: The ShapeNetPart dataset [62] contains 16,880 models from 16 shape classes. There are 14,006 models for training and 2,874 models for testing. Each point is annotated with one label from 50 parts, and the number of parts for each class is 2-6. For a fair comparison, we follow the same testing protocol with [61]. The category mIoU and the instance mIoU are employed for evaluation.

2) *Results*: Similar to the experiments on ModelNet40, we compare our method (BKR+FMD) with the competitors on all four backbones for the object part segmentation task. The results are presented in Table III. Our method surpasses PEFD by 1.97% and 1.41% on instance mIoU and category mIoU for PointNet++ [6], respectively. In the case of DGCNN, although all the distillation methods improve the performance of the original student, our method achieves the largest improvement. For PointConv, our method achieves an improvement of 1.27% and 3.4% on instance mIoU and category mIoU, respectively. The success on the object part segmentation task further reveals the applicability of our method.

### C. Semantic Segmentation

1) *Data and Metrics*: The S3DIS dataset [63] contains 271 rooms in 6 indoor areas. There are 273 million 3D RGB points scanned from three different buildings, each of which is assigned a semantic label from 13 classes. We train the models on Areas 1-4 and 6 and test on Area 5, which is unseen during training. The mean classwise intersection over union (mIoU), mAcc and OA are employed as the evaluation metrics.

TABLE IV: Results on semantic segmentation.

	PN++	(1/8)PN++	F-L2	SP	FitNet	NST	AT	KD	OFD	DKD	PEFD	Ours
OA	82.75	79.58	79.26	79.38	79.68	79.69	80.26	80.33	80.58	80.40	80.61	<b>81.02</b>
mAcc	61.16	57.25	56.89	57.20	57.75	57.63	58.01	58.17	58.60	58.47	58.73	<b>60.30</b>
mIoU	52.23	46.09	45.34	45.83	46.39	46.24	47.27	47.42	48.32	48.15	48.44	<b>50.03</b>
	DGC	(1/8)DGC	F-L2	SP	FitNet	NST	AT	KD	OFD	DKD	PEFD	Ours
OA	83.70	77.19	78.59	79.39	78.24	78.72	79.16	77.83	79.44	78.36	79.56	<b>80.05</b>
mAcc	54.07	43.16	46.16	47.49	44.17	47.02	47.14	44.03	47.77	44.52	47.83	<b>48.50</b>
mIoU	47.21	35.51	36.78	39.62	36.50	37.28	38.42	35.92	39.26	36.69	39.19	<b>39.92</b>
	PCConv	(1/8)PCConv	F-L2	SP	FitNet	NST	AT	KD	OFD	DKD	PEFD	Ours
OA	84.76	81.99	81.57	81.56	81.76	81.84	82.29	82.33	82.58	82.47	82.68	<b>83.62</b>
mAcc	65.68	60.04	59.57	59.64	59.78	59.95	61.34	61.56	61.83	61.55	61.78	<b>62.52</b>
mIoU	55.31	51.47	50.84	51.04	51.10	51.23	51.45	51.77	52.23	51.84	52.14	<b>52.75</b>

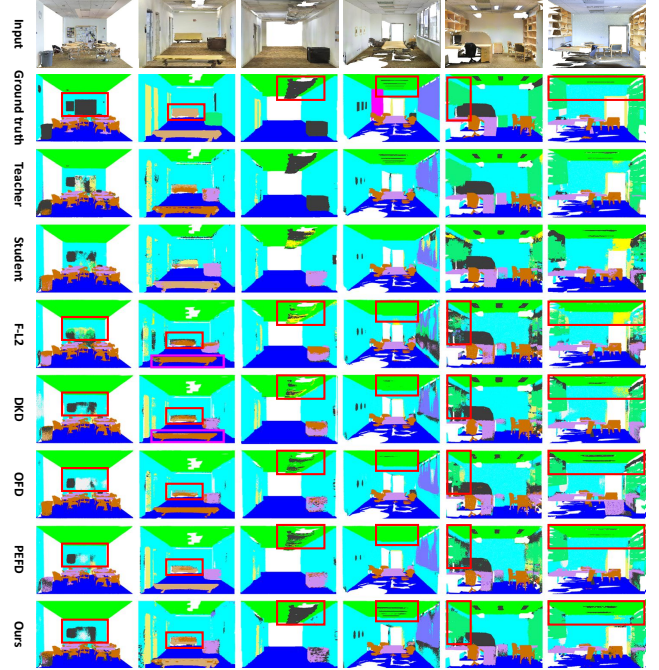


Fig. 4: Visualization of semantic segmentation results.

2) *Results*: Similarly, we conduct comparisons between our method and other distillation methods on several backbones. However, due to the more complex scenarios and serious self-obscuring, semantic segmentation is more challenging than object part segmentation, leading to less effectiveness of the distillation. As shown in Table IV, there are performance drops or slight performance improvements with previous distillation methods. With the help of reconfiguration and alignment, our method consistently and significantly improves the semantic segmentation performance, especially on mAcc and mIoU.

3) *Visualization*: Fig. 4 presents the visualization results of PointNet++ [6] on S3DIS. The predictions of our method are closer to the ground truth and capture more connected and consistent local details since the bidirectional knowledge reconfiguration has the ability to well inherit the contextual knowledge from the teacher model. Besides, Fig. 4 also shows qualitative comparisons between our method with other competing methods, including F-L2, DKD, OFD and PEFD. As shown in Fig. 4, F-L2 obtains unsatisfactory performance, which even degenerates the performance of the student model.

TABLE V: Ablation study on (a) shape classification, (b) object part segmentation, and (c) semantic segmentation. T and S represent the original teacher and the compressed student.

(a) Shape Classification			(b) Object Part Segmentation			(c) Semantic Segmentation			
Methods	OA	mAcc	Methods	cat. mIoU	ins. mIoU	Methods	OA	mAcc	mIoU
(T)PointNet++	92.54	90.17	(T)PointNet++	85.21	81.74	(T)PointNet++	82.75	61.16	52.23
(S)(1/8)PointNet++	88.48	80.61	(S)(1/8)PointNet++	76.29	58.05	(S)(1/8)PointNet++	79.58	57.25	46.09
F-L2	88.03	79.69	F-L2	75.92	57.93	F-L2	79.26	56.89	45.34
REMD	88.86	82.54	REMD	76.60	58.05	REMD	79.78	57.75	47.01
FMD	89.06	82.77	FMD	76.88	58.14	FMD	80.23	58.77	47.66
TDKR+FMD	89.75	83.94	TDKR+FMD	77.46	58.87	TDKR+FMD	80.47	59.22	48.64
BUKR+FMD	89.18	83.09	BUKR+FMD	77.20	58.23	BUKR+FMD	80.53	58.89	48.31
TDKR+BUKR+FMD	89.92	84.43	TDKR+BUKR+FMD	77.91	59.67	TDKR+BUKR+FMD	80.80	59.72	49.41
BKR+FMD (Ours)	<b>90.28</b>	<b>84.55</b>	BKR+FMD (Ours)	<b>79.22</b>	<b>59.84</b>	BKR+FMD (Ours)	<b>81.02</b>	<b>60.30</b>	<b>50.03</b>

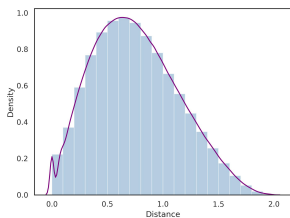


Fig. 5: Histogram of the distance between point pairs.

This may be primarily due to its inherent problem of position inconsistency, which results in misaligned knowledge that distracts the distillation procedure. DKD outperforms F-L2 since it transfers the knowledge in logits, effectively mitigating the issue of position inconsistency in the intermediate features. However, the neglect of information within the intermediate features by DKD leads to insufficient transferred knowledge. For example, in the third example, DKD treats “clutter” (black) as “ceiling” (green). Meanwhile, the feature distillation methods, *i.e.*, OFD and PEFD, achieve better results by effectively utilizing the rich information in the intermediate features. Compared to these competitors, our method achieves the best performance. As shown in Fig. 4, our method excels in accurately segmenting both global and local semantic areas. For example, it successfully captures the global shape in the fifth example and accurately identifies the local object, such as sofa, in the second example. These results demonstrate the necessity of FMD in solving the position inconsistency problem, as well as the effectiveness of BKR in leveraging diverse knowledge within the intermediate features.

#### D. Analysis

1) *Position Inconsistency in Feature Distillation*: To clarify the position inconsistency problem, we simulate the sampling process and visualize the normalized frequency histogram of the distance between point positions sampled by the teacher and the student. In particular, we employ the preprocessed ModelNet40 dataset in which each input contains 1024 points and samples 512 points for the teacher and the student. The sampling process is the same as the first stage of many point cloud analysis models, such as PointNet++ [6] and PointConv [10]. All the training data are used, and the distance between point pairs is calculated by Euclidean distance. We then count and plot the normalized frequency histogram.

TABLE VI: Results of different feature levels.

	(a) $L_2$	(b) FMD
Levels	mIoU	mIoU
Level-1	46.18	47.14
Level-2	46.39	46.94
Level-3	47.74	47.98
Level-4	47.73	48.35
Level-3,4	46.52	48.55
Level-2,3,4	46.38	48.83
Level-1,2,3,4	45.34	49.19

As shown in Fig. 5, the distance is between  $0 \sim 2$  because the input point cloud positions are normalized to  $-1 \sim 1$  during data preprocessing. Obviously, many point pairs have nonnegligible Euclidean distances. Specifically, approximately 26.98% of the point pairs have a Euclidean distance greater than 1, indicating a large inconsistency between the teacher and student. Such inconsistency leads to misaligned intermediate features, which limits feature distillation effectiveness. There are also some point pairs with distances less than 0.25 or even equal to 0. These aligned or near-aligned points account for why other distillation methods can be effective without feature alignment.

We further analyze the influence of position inconsistency at different levels in feature distillation. We choose PointNet++ as the backbone and conduct experiments on the S3DIS dataset. Specifically, distillation is performed on the features of each level separately. Two distillation methods are employed. One is our baseline which directly forces the student feature to mimic the teacher feature by  $L_2$  loss. The other one replaces the distillation loss in the baseline with the proposed FMD. The results are summarized in Table VI. For  $L_2$ , when distilling with single-level, it is observed that shallow features are more sensitive to position inconsistency than deep features. This is because the higher the feature level, the larger the perceptual area captured, which is able to represent more global shape information, making less misalignment between features of different positions. However, the performance is getting worse when more levels are added for distillation. This is because the utilization of more distillation levels may potentially lead to the accumulation of misaligned knowledge distillation. Besides, the inconsistency in knowledge transfer across different levels may further disrupt the distillation procedure, ultimately leading to a decline in performance. For FMD, since features are well aligned in FMD, both low-level and high-level knowledge can be well transferred from the



TABLE VII: The universality and effectiveness of FMD. Experiments are conducted on (a) shape classification and (b) object part segmentation with PointNet++ as the backbone.

(a) Shape Classification								
	F-L2	SP	FitNet	NST	AT	OFD	PEFD	BKR
OA	88.03	88.31	88.57	88.61	89.08	89.23	89.28	89.68
mAcc	79.69	80.03	82.17	82.59	82.79	83.25	83.27	83.76
F-L2+FMD SP+FMD FitNet+FMD NST+FMD AT+FMD OFD+FMD PEFD+FMD BKR+FMD								
OA	88.81	89.65	88.85	89.38	89.47	89.75	89.68	<b>90.28</b>
mAcc	83.08	83.66	82.33	83.87	83.84	83.93	83.98	<b>84.55</b>

(b) Object Part Segmentation								
	F-L2	SP	FitNet	NST	AT	OFD	PEFD	BKR
ins.mIoU	75.92	75.99	76.34	76.50	76.71	76.94	77.25	77.76
cat.mIoU	57.93	57.97	58.03	58.37	58.25	58.60	58.43	58.44
F-L2+FMD SP+FMD FitNet+FMD NST+FMD AT+FMD OFD+FMD PEFD+FMD BKR+FMD								
ins.mIoU	77.37	77.78	76.59	77.61	77.58	77.76	77.65	<b>79.22</b>
cat.mIoU	58.73	58.59	58.13	58.41	58.53	58.79	58.83	<b>59.84</b>

teacher to the student, obtaining satisfactory results. Besides, as the shallow feature in Level-1 captures more local structure information which is significant in the semantic segmentation task, the better performance of Level-1 can be attributed to the abundant local structure information. Besides, it is worth noting that employing multi-level features in feature distillation outperforms the single-level method, which is consistent with the observation of previous methods [39], [65].

2) *Ablation Study*: To further demonstrate the effectiveness of the proposed BKR and FMD, we design an ablation study on the ModelNet40, ShapeNetPart, and S3DIS datasets with PointNet++ as the backbone. F-L2 is our baseline. As shown in Table V, utilizing the proposed FMD can boost the performance. In classification, FMD helps the student model outperform F-L2 by 3.08%. The effectiveness of FMD is more remarkable on semantic segmentation. Specifically, FMD outperforms REMD by 0.45%, 1.02%, and 0.65% on OA, mAcc, and mIoU, respectively. We also combine FMD with other feature distillation methods. As shown in table VII, FMD can consistently improve performance, demonstrating the universality and effectiveness of FMD.

In addition, we quantitatively analyze the effectiveness of TDKR, BUKR and BKR. As shown in Table V, although BUKR+FMD improves the performance of FMD marginally, the collaboration effect between TDKR and BUKR is remarkable which is consistent with our conclusion that “both local structure and global shape information are essential clues for point cloud”. Taking object part segmentation as an example, on the one hand, TDKR+BUKR+FMD outperforms TDKR+FMD by 0.80% on ins. mIoU, indicating the assisting role of BUKR to TDKR. On the other hand, TDKR+BUKR+FMD outperforms FMD by 1.03% and 1.53% on cat. mIoU and ins. mIoU, further proving the necessity of collaboration between TDKR and BUKR. In addition, our framework (BKR+FMD), which combines TDKR, BUKR and residual connection, achieves the best results, showing that residual connections can bring rich information and improve knowledge transfer. Moreover, as shown in Table VII, although the results of other feature distillation methods can be improved by FMD, they are still inferior to our method, *i.e.*, BKR+FMD, further demonstrating the superiority of BKR.

3) *Analysis of Hyperparameters*: We analyze the nearest number  $k$  in FMD and the tradeoff parameter  $\lambda$  by cross-

TABLE VIII: Analysis of the parameter  $k$ .

(a) Shape Classification					
k	1	3	5	7	9
OA	89.29	89.49	<b>90.28</b>	89.63	89.60
mAcc	83.84	84.91	<b>84.55</b>	83.63	83.29

(b) Object Part Segmentation					
k	1	3	5	7	9
ins.mIoU	76.72	78.46	<b>79.20</b>	78.92	77.75
cat.mIoU	58.14	59.61	<b>59.84</b>	59.40	58.44

TABLE IX: Analysis of the tradeoff parameter  $\lambda$ .

(a) Shape Classification					
$\lambda$	0.1	0.05	0.01	0.005	0.001
OA	<b>90.28</b>	89.91	89.60	89.82	89.71
mAcc	<b>84.55</b>	83.31	83.76	84.13	83.28

(b) Object Part Segmentation					
$\lambda$	0.1	0.05	0.01	0.005	0.001
ins.mIoU	77.99	78.41	<b>79.20</b>	78.56	78.23
cat.mIoU	59.70	59.29	<b>59.84</b>	59.39	59.22

validation. Specifically, 20% of the training set is used as the validation set, and the rest is employed to train the model. We vary  $k$  in 1, 3, 5, 7, 9 and  $\lambda$  in 0.1, 0.05, 0.01, 0.005, 0.001. Experiments are conducted on ModelNet40 [61] for shape classification and ShapeNetPart [62] for object part segmentation with PointNet++ [6] as the backbone.

Table VIII and Table IX present the results of varying  $k$  and  $\lambda$ , respectively. Our method is more sensitive to the nearest number  $k$  than the tradeoff parameter  $\lambda$ . This is because the number of nearest neighbors in FMD controls the receptive scale of the student and further determines the scope of contextual knowledge transferred from the teacher. Within a certain range, a larger  $k$  will form a better representation of the local structure. After adding more neighbors, the performance will not increase because only a moderate  $k$  can balance the local structure information and global shape knowledge. As shown in Table VIII, we chose  $k = 5$  in our experiments.

## V. CONCLUSIONS

In this paper, we design a universal feature distillation strategy for lightweight point cloud analysis. Since both the local structure knowledge and global shape knowledge of the teacher are essential for the student, a bidirectional knowledge reconfiguration (BKR) is presented to inherit the contextual knowledge from the teacher to all the scales of the student using bidirectional reconfiguration. Specifically, a top-down reconfiguration is developed for inheriting diverse local structure information, and a bottom-up reconfiguration is employed to inherit high-level shape knowledge. Since there exists a potential position inconsistency caused by the random point sampling operation in point cloud analysis, a feature mover’s distance (FMD) is proposed to conduct the feature alignment. Experiments on shape classification, part segmentation and semantic segmentation benchmarks with various point cloud analysis networks show the effectiveness and universality of our framework.

## VI. ACKNOWLEDGMENTS

This research is sponsored by Beijing Nova Program (Z211100002121106), National Natural Science Foundation of China (Grant No. 62306041, 62276031).

## REFERENCES

- [1] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *CVPR*, 2018.
- [2] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *ICCV*, 2015.
- [3] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *CVPR*, 2017.
- [4] L. Li, Z. Li, S. Liu, and H. Li, "Frame-level rate control for geometry-based lidar point cloud compression," *TMM*, 2022.
- [5] C.-H. Wu, C.-F. Hsu, T.-K. Hung, C. Griwodz, W. T. Ooi, and C.-H. Hsu, "Quantitative comparison of point cloud compression algorithms with pcc arena," *TMM*, 2022.
- [6] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *NeurIPS*, 2017.
- [7] Y. Shen, C. Feng, Y. Yang, and D. Tian, "Mining point cloud local structures by kernel correlation and graph pooling," in *CVPR*, 2018.
- [8] Y. Liu, B. Fan, G. Meng, J. Lu, S. Xiang, and C. Pan, "Densepoint: Learning densely contextual representation for efficient point cloud processing," in *ICCV*, 2019.
- [9] J. Choe, C. Park, F. Rameau, J. Park, and I. S. Kweon, "Pointmixer: Mlp-mixer for point cloud understanding," in *ECCV*, 2022.
- [10] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *CVPR*, 2019.
- [11] S. Qiu, S. Anwar, and N. Barnes, "Geometric back-projection network for point cloud classification," *TMM*, 2022.
- [12] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *ICCV*, 2021.
- [13] X.-F. Han, Y.-F. Jin, H.-X. Cheng, and G.-Q. Xiao, "Dual transformer for point cloud analysis," *TMM*, 2022.
- [14] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *ICLR*, 2015.
- [15] C. Chen, S. Qian, Q. Fang, and C. Xu, "Hapgn: Hierarchical attentive pooling graph network for point cloud segmentation," *TMM*, 2021.
- [16] W. Wu, Q. Shan, and L. Fuxin, "Pointconvformer: Revene of the point-based convolution," *arXiv preprint arXiv:2208.02879*, 2022.
- [17] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *CVPR*, 2020.
- [18] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *TOG*, 2019.
- [19] H. Zhou, Y. Feng, M. Fang, M. Wei, J. Qin, and T. Lu, "Adaptive graph convolution for point cloud analysis," in *ICCV*, 2021.
- [20] G. Hinton, O. Vinyals, J. Dean *et al.*, "Distilling the knowledge in a neural network," in *NeurIPS*, 2014.
- [21] P. K. Sharma, A. Abraham, and V. N. Rajendiran, "A generalized zero-shot quantization of deep convolutional neural networks via learned weights statistics," *TMM*, 2021.
- [22] Z. Hao, Y. Luo, Z. Wang, H. Hu, and J. An, "Cdfkd-mfs: Collaborative data-free knowledge distillation via multi-level feature sharing," *TMM*, 2022.
- [23] L. Zhang, D. Du, C. Li, Y. Wu, and T. Luo, "Iterative knowledge distillation for automatic check-out," *TMM*, 2021.
- [24] X. Wu, R. He, Y. Hu, and Z. Sun, "Learning an evolutionary embedding via massive knowledge distillation," *IJCV*, 2020.
- [25] J. She, Y. Hu, H. Shi, J. Wang, Q. Shen, and T. Mei, "Dive into ambiguity: Latent distribution mining and pairwise uncertainty estimation for facial expression recognition," in *CVPR*, 2021.
- [26] C. Yang, H. Zhou, Z. An, X. Jiang, Y. Xu, and Q. Zhang, "Cross-image relational knowledge distillation for semantic segmentation," in *CVPR*, 2022.
- [27] D. Ji, H. Wang, M. Tao, J. Huang, X.-S. Hua, and H. Lu, "Structural and statistical texture knowledge distillation for semantic segmentation," in *CVPR*, 2022.
- [28] X. Qu, C. Ding, X. Li, X. Zhong, and D. Tao, "Distillation using oracle queries for transformer-based human-object interaction detection," in *CVPR*, 2022.
- [29] G. Li, X. Li, Y. Wang, S. Zhang, Y. Wu, and D. Liang, "Knowledge distillation for object detection via rank mimicking and prediction-guided feature imitation," in *AAAI*, 2022.
- [30] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *IJCV*, 2021.
- [31] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang, "Decoupled knowledge distillation," in *CVPR*, 2022.
- [32] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," in *ICLR*, 2015.
- [33] Z. Huang and N. Wang, "Like what you like: Knowledge distill via neuron selectivity transfer," *arXiv preprint arXiv:1707.01219*, 2017.
- [34] D. Chen, J.-P. Mei, H. Zhang, C. Wang, Y. Feng, and C. Chen, "Knowledge distillation with the reused teacher classifier," in *CVPR*, 2022.
- [35] Y. Chen, S. Wang, J. Liu, X. Xu, F. de Hoog, and Z. Huang, "Improved feature distillation via projector ensemble," in *NeurIPS*, 2022.
- [36] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *ICLR*, 2016.
- [37] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *ICCV*, 2019.
- [38] P. Chen, S. Liu, H. Zhao, and J. Jia, "Distilling knowledge via knowledge review," in *CVPR*, 2021.
- [39] D. Chen, J.-P. Mei, Y. Zhang, C. Wang, Z. Wang, Y. Feng, and C. Chen, "Cross-layer distillation with semantic calibration," in *AAAI*, 2021.
- [40] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *IJCV*, 2000.
- [41] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *NeurIPS*, 2013.
- [42] H. Phan and A. Nguyen, "Deepface-emd: Re-ranking using patch-wise earth mover's distance improves out-of-distribution face identification," in *CVPR*, 2022.
- [43] K. D. Doan, P. Yang, and P. Li, "One loss for quantization: Deep hashing with discrete wasserstein distributional matching," in *CVPR*, 2022.
- [44] Z. Zhang, Y. Liu, C. Han, H. Shi, T. Guo, and B. Zhou, "Petsgan: Rethinking priors for single image generation," in *AAAI*, 2022.
- [45] C. Zhang, Y. Cai, G. Lin, and C. Shen, "Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers," in *CVPR*, 2020.
- [46] P. Mandikal and V. B. Radhakrishnan, "Dense 3d point cloud reconstruction using a deep pyramid network," in *WACV*, 2019.
- [47] J. Chen, Y. P. Liu, R. Peng, and A. Ramaswami, "Exponential convergence of sinkhorn under regularization scheduling," *arXiv preprint arXiv:2207.00736*, 2022.
- [48] N. Bonneel, J. Rabin, G. Peyré, and H. Pfister, "Sliced and radon wasserstein barycenters of measures," *JMIV*, 2015.
- [49] K. Nguyen, N. Ho, T. Pham, and H. Bui, "Distributional sliced-wasserstein and applications to generative modeling," in *ICLR*, 2020.
- [50] K. Fatras, Y. Zine, R. Flamary, R. Gribonval, and N. Courty, "Learning with minibatch wasserstein: asymptotic and gradient properties," *arXiv preprint arXiv:1910.04091*, 2019.
- [51] K. Nguyen, D. Nguyen, Q. Nguyen, T. Pham, H. Bui, D. Phung, T. Le, and N. Ho, "On transportation of mini-batches: A hierarchical approach," in *ICML*, 2022.
- [52] N. Kolkin, J. Salavon, and G. Shakhnarovich, "Style transfer by relaxed optimal transport and self-similarity," in *ICCV*, 2019.
- [53] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *CVPR*, 2017.
- [54] C. Guo, B. Fan, Q. Zhang, S. Xiang, and C. Pan, "Augfpn: Improving multi-scale feature learning for object detection," in *CVPR*, 2020.
- [55] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *CVPR*, 2018.
- [56] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *CVPR*, 2019.
- [57] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018.
- [58] E. Ricci, G. Zen, N. Sebe, and S. Messelodi, "A prototype learning framework using emd: Application to complex scenes analysis," *TPAMI*, 2012.
- [59] J. Wagner and B. Ommer, "Efficient clustering earth mover's distance," in *ACCV*, 2010.
- [60] I. Lang, A. Manor, and S. Avidan, "Samplenet: Differentiable point cloud sampling," in *CVPR*, 2020.
- [61] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *CVPR*, 2015.
- [62] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas, "A scalable active framework for region annotation in 3d shape collections," *TOG*, 2016.
- [63] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *CVPR*, 2016.
- [64] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, and J. Y. Choi, "A comprehensive overhaul of feature distillation," in *ICCV*, 2019.

- [65] M. Ji, B. Heo, and S. Park, “Show, attend and distill: Knowledge distillation via attention-based feature matching,” in *AAAI*, 2021.